Enhancing Hypergraph Neural Networks with Intent Disentanglement for Session-based Recommendation

Yinfeng Li
Department of Electronic Engineering, Tsinghua University, Beijing, China

Chen Gao†
Department of Electronic Engineering, Tsinghua University, Beijing, China

Hengliang Luo
Meituan Inc., Beijing, China

Depeng Jin
Department of Electronic Engineering, Tsinghua University, Beijing, China

Yong Li
Department of Electronic Engineering, Tsinghua University, Beijing, China

ABSTRACT

Session-based recommendation (SBR) aims at the next-item prediction with a short behavior session. Existing solutions fail to address two main challenges: 1) user interests are shown as dynamically coupled intents, and 2) sessions always contain noisy signals. To address them, in this paper, we propose a hypergraph-based solution, **HIDE**. Specifically, HIDE first constructs a hypergraph for each session to model the possible interest transitions from distinct perspectives. HIDE then disentangles the intents under each item click in micro and macro manners. In the micro-disentanglement, we perform intent-aware embedding propagation on session hypergraph to adaptively activate disentangled intents from noisy data. In the macro-disentanglement, we introduce an auxiliary intent-classification task to encourage the independence of different intents. Finally, we generate the intent-specific representations for the given session to make the final recommendation. Benchmark evaluations demonstrate the significant performance gain of our HIDE over the state-of-the-art methods.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**;

KEYWORDS

Session-based Recommendation; Graph Neural Networks; Disentangled Representation Learning

ACM Reference Format:
Yinfeng Li, Chen Gao†, Hengliang Luo, Depeng Jin, and Yong Li. 2022. Enhancing Hypergraph Neural Networks with Intent Disentanglement for Session-based Recommendation. In *Proceedings of the 45th Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531794

†Chen Gao is the corresponding author (chgao96@gmail.com).

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

ACM ISBN 978-1-4503-8732-3/22/07.
https://doi.org/10.1145/3477495.3531794

1 INTRODUCTION

Recommender systems play critical roles in helping users discover items on e-commerce platforms. However, in real-world scenarios, user profiles and long-term historical interactions are usually not available [31], resulting inferior performance of traditional recommendation approaches [4, 10, 24]. Hence, session-based recommendation (SBR) [11], focusing on predicting users' next interacted items based on short anonymous behavior sequences (sessions), has attracted broad attention in the community.

The existing works of SBR can be principally divided into three types: conventional models, sequential-based models, and GNN-based models [9]. **Conventional models** [23, 25] rely on item co-occurrences but neglect the sequential patterns. **Sequential-based methods** [11, 16, 18] model the items in chronological order with GRU [6] or attention layers [27]. Recently, Graph Neural Networks (GNN) [26] have been getting increasing attention in SBR. In particular, **GNN-based methods** represent each session as a graph to consider the relations of items, which further fall into two categories: normal graph (NG) and hypergraph (HG) [1]. Specifically, NG-based methods [31, 32] model the session as directed item-item graph to capture the item correlations; HG-based methods [33] regard the item transitions as high-order relations (item transition is often triggered by the joint effect of previous item clicks) and introduce hypergraph to model the high-order relations. However, these existing works commonly concentrate on how to model the item transitions, with two unresolved challenges as follows.

- **User interests are dynamic and made up of coupled intents.** The item transitions in a session reveal the user's dynamic interests, and each item click can be regarded as the result of coupled intents. For example, user clicks an *iPhone* probably because he/she is an *Apple fan* or he/she just wants to buy a smartphone.
- **Session contains both user interests and noisy signals.** In SBR, a user may accidentally click on items that are not interested. Suppose a session for a certain user, *iPhone* → *iPad* → *milk* → *AirPods*. Obviously, *milk* may be a noisy sample that clicked by mistake which may affect the modeling of the user's real interests.

To tackle these challenges, we propose ***H**ypergraph neural networks with **I**ntent **D**is**E**ntanglement* (**HIDE**). In particular, we first convert each session sequence into a hypergraph [8], encoding the possible high-order interest transitions with multiple types of hyperedges from distinct perspectives. Then, to address the coupled intents in the user's dynamic interests, we propose to disentangle the intents under each item click in both micro and macro manners. For the micro-disentanglement, we slice each item embedding into disentangled chunks (each chunk corresponds to a specific intent) and then separately perform intent-aware embedding propagation to learn the disentangled intent in each item click. For the macro-disentanglement, we introduce an auxiliary learning task of intent classification to ensure the independence of the disentangled chunks. Finally, we unify the recommendation task and the intent classification task under a *primary&auxiliary* learning framework and jointly optimize them for better recommendation and intent disentanglement.

To sum up, the main contributions of our paper are as follows.

- We develop HIDE to 1) capture the high-order interest transitions with hypergraph and 2) activate user's core preferences under different intents with intent disentanglement.
- We perform intent-aware attentive convolution on each session hypergraph to generate disentangled intent chunks (**micro intent disentanglement**) , and further unify two tasks (recommendation and intent classification) to encourage the independence of different intents (**macro intent disentanglement**).
- Extensive experiments on benchmark datasets show HIDE has overwhelming superiority over the state-of-the-art methods.

2 PRELIMINARIES

2.1 Problem Statement

Let $\mathcal{V} = \{v_1, v_2, ..., v_m\}$ denote the set of items where m is the number of items. A session s can be denoted as an item-click sequence $s = [v_{s,1}, v_{s,2}, ..., v_{s,n}]$, which ordered by the timestamps, where $v_{s,i} \in \mathcal{V}$ is the item clicked in s and n is the session length. Given a session s, SBR aims to predict the next interaction $v_{s,n+1}$. The recommendation model generates preference scores for all the items in $\mathcal{V}$ and selects top-K preferred items to make the final recommendation.

2.2 Hypergraph

Hypergraph [1] introduce a special kind of edge (can connect more than two nodes), *hyperedge*, to capture high-order relations. Formally, a hypergraph can be formulated as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of nodes and hyperedges, respectively. The hyperedge $e \in \mathcal{E}$ can be denoted as $e = \{v_{i_1}, v_{i_2}, ..., v_{i_k}\} \subseteq \mathcal{V}$, and thus hyperedge can be regarded as a subset of node set $\mathcal{V}$.

3 METHODOLOGY

In this section, we first introduce the initialization of intent-aware embedding for each item and then show how to construct a session hypergraph for each session from distinct perspectives. After that, we represent how to generate disentangled item embeddings with micro-and macro-disentanglement. Finally, we describe how to predict the next item and optimize the HIDE model.

3.1 Intent-Aware Embedding

As discussed above, existing SBR methods [22, 31–33] embed the items into the same embedding space, which fails to capture disentangled user intents. Distinct from those methods, we slice the item embeddings into K chunks, coupling each chunk with a certain user intent. For each item $v_i \in \mathcal{V}$, its intent-aware embedding is initialized as $\mathbf{h}_{v_i} = (\mathbf{h}^1_{v_i}, \mathbf{h}^2_{v_i}, ..., \mathbf{h}^K_{v_i})$, where $\mathbf{h}_{v_i} \in \mathbb{R}^d$ is the embedding of v_i and d denotes the embedding size. $\mathbf{h}^k_{v_i} \in \mathbb{R}^{\frac{d}{K}}$ is v_i's chunked representation under k-th intent. Obviously, the k-th chunk of all items are in the same k-th subspace, which indicates a certain user intent. For k-th subspace, we build an *intent prototype* (serves as the center of the corresponding subspace), by clustering the k-th chunk of all items, denoted as $\mathbf{h}^k_p = \mathbf{Mean}(\mathbf{h}^k_{v_i} | v_i \in \mathcal{V})$.

3.2 Session Hypergraph Construction

In SBR, the item transitions are many-to-many and thus high-order since the current item click is often triggered by the joint effect of previous item clicks. Moreover, the contextual relations of items and the similarity of items under certain intent are also high-order. To accurately capture the above high-order relations, we propose to construct a hypergraph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ for each session s, where node set $\mathcal{V}_s \subseteq \mathcal{V}$ consists of the clicked items in session s and $\mathcal{E}_s$ denotes the hyperedge set. As Figure 1 (a) shows, we construct three types of hyperedges to capture the high-order item relations from three distinct perspectives as follows.

a) Transition Hyperedges $\mathcal{E}^t_s$. The relative chronological order of item transitions is demonstrated the key factor to SBR [11, 32]. To maintain the item-transitions order in each session, as Figure 1 (a) shows, for item v_2, we connect its incoming items $\{v_1, v_2, v_3\}$ with a hyperedge, which reveals the high-order correlation of items that facilitate the click on v_2.

b) Context Hyperedges $\mathcal{E}^c_s$. Since sequential context depicts user's latent interests [28], we utilize w-size sliding window on item sequence to capture local interest. Then the items in window are connected by hyperedges, such as $e = \{v_1, v_2\} \in \mathcal{E}^c_{s_2}$ when $w = 2$. Obviously, with various window sizes, we can obtain user's local interests from different granularities. Finally, we gather hyperedges from different sliding windows as $\mathcal{E}^c_s = \cup^W_{w=1} \mathcal{E}^c_{s_w}$.

c) Intent Hyperedges $\mathcal{E}^i_s$. The items under different intents tend to have different similarities. For example, *iPhone* and *iPad* are similar under the intent of *Apple brand* while irrelevant under the intent of *buying a smartphone*. Hence, we construct intent hyperedges to capture intent-specific item correlations based on intent prototypes. For the k-th intent, we first calculate the cosine similarity between the k-th intent prototype and item $v_i \in \mathcal{V}_s$, denoted as $S_{ki} = \cos(\mathbf{h}^k_p, \mathbf{h}^k_{v_i})$. We then regard each intent as a hyperedge that connects the top-εn items according to S_{ki}, where n is the item number in session, and ε controls the sparsity of hyperedge.

We gather the above three types of hyperedges to generate the hyperedge set of session s as $\mathcal{E}_s = \mathcal{E}^t_s \cup \mathcal{E}^c_s \cup \mathcal{E}^i_s$. Note that we remove the redundant overlapping hyperedges during the gathering.

3.3 Micro-disentanglement

Session data always contains noisy signals such as items clicked by mistake, which may confuse the recommendation model. Luckily, the intents provide an opportunity to filter out noises, since intents under item clicks in each session are more stable. For example, the intent of session *iPhone* → *iPad* → *Macbook* → *AirPods* may be *Apple brand*. Hence, it will be easier to detect the items clicked by mistake as noise from the intent perspective. To sufficiently capture

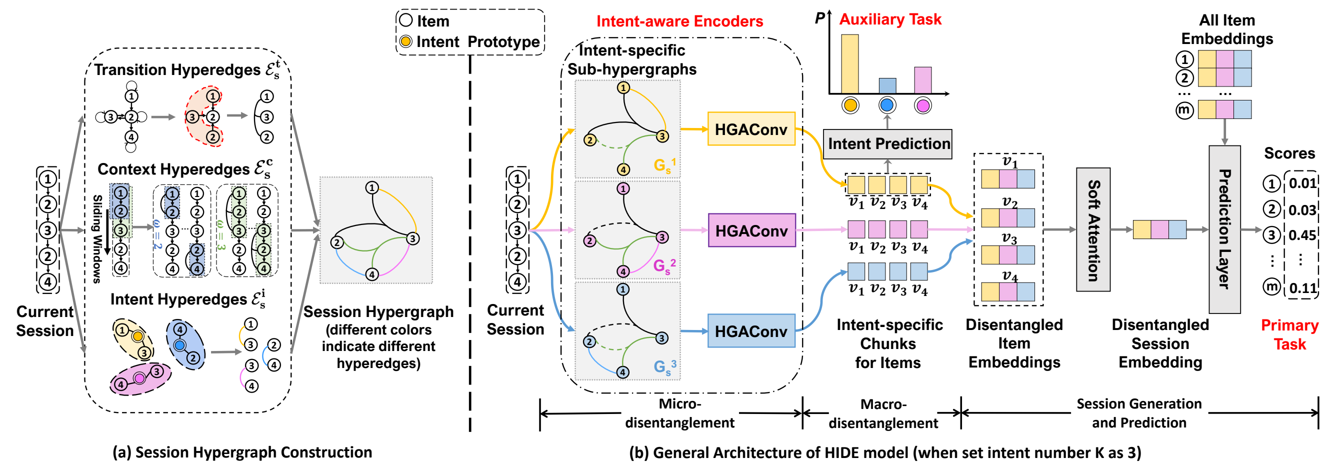

Figure 1: Illustration of the session hypergraph construction (a) and the general architecture of HIDE model (b).

user intents, we propose to disentangle K distinct user intents with K parallel intent-aware encoders in a micro manner, where each encoder separately captures a certain intent.

Intent-aware Encoder. To ensure the encoder only captures one aspect of intent, we propose to perform propagation on intent-specific sub-hypergraph with corresponding intent chunks of items. For each session s, its sub-hypergraph under k-th intent $\mathcal{G}_s^k$ contains all the transition and context hyperedges but only involves the intent hyperedge under k-th intent. As for the node features, we only take the k-th chunk for each item as the input of the encoder. Inspired by [7, 28], we propose hypergraph attentive convolution (**HGAConv**) to learn intent-specific chunks for items on intent-specific sub-hypergraph $\mathcal{G}_s^k$, which contains *node to hyperedge (n2e)* and *hyperedge to node (e2n)* propagation.

1) Node to hyperedge (n2e). Some nodes connected by a hyperedge reveal intents, but others may be noise. Therefore we aggregate the nodes v_o with attention mechanism [27] to obtain the corresponding hyperedge feature $\mathbf{f}_j^k$ under k-th intent, denoted as $\mathbf{f}_j^k = \mathbf{AGG}_{n2e}\left(\alpha_{jo}^k \mathbf{h}_{v_o}^k | v_o \in e_j\right)$, where $\mathbf{AGG}_{n2e}$ is the aggregation function (we use SUM here) and α_{jo}^k denotes the attention coefficient of node v_o in hyperedge e_j. To calculate it, we assume that the nodes connected by hyperedge e_j can form a cluster and then calculate the cluster's average as $\mathbf{h}_{c_j}^k = \mathrm{Mean}(\mathbf{h}_{v_o}^k | v_o \in e_j)$. Since nodes close to the cluster center are more likely to be core intents, we calculate attention scores as follows,

$$\alpha_{jo}^k = \frac{\exp(\mathrm{LeakyReLU}(\mathbf{q}_1^{k\top}(\mathbf{h}_{c_j}^k \odot \mathbf{h}_{v_o}^k)))}{\sum_{v_{o'} \in e_j} \exp(\mathrm{LeakyReLU}(\mathbf{q}_1^{k\top}(\mathbf{h}_{c_j}^k \odot \mathbf{h}_{v_{o'}}^k)))}, \quad (1)$$

where $\mathbf{q}_1^k \in \mathbb{R}^{\frac{d}{K}}$ is the attention vector under the k-th intent and $\odot$ denotes the Hadamard product.

2) Hyperedge to node (e2n). Given the hyperedge features, we can further update node chunk-embedding under k-th intent as $\mathbf{h}_{v_i}^{k\,\prime} = \mathbf{AGG}_{e2n}\left(\beta_{ij}^k \mathbf{f}_j^k | e_j \in \mathcal{E}_{s_{v_i}}\right)$, where $\mathbf{h}_{v_i}^{k\,\prime}$ is the output feature of node v_i and β_{ij}^k is the attention coefficient of hyperedge e_j on node v_i in k-th intent. Here $\mathcal{E}_{s_{v_i}}$ is the set of hyperedges that connected to v_i. For each node v_i, we generate query representation as $\mathbf{h}_{q_{v_i}}^k = \mathbf{h}_{v_i}^k + \mathbf{s}^k$, where $\mathbf{s}^k$ is the average k-th intent feature of items in session s. Since hyperedges that match the intent of the current session and current item click will be more favorable (assigned with larger weights), we calculate the query-aware attention score as follows,

$$\beta_{ij}^k = \frac{\exp(\mathrm{LeakyReLU}(\mathbf{q}_2^{k\top}(\mathbf{h}_{q_{v_i}}^k \odot \mathbf{f}_j^k)))}{\sum_{e_{j'} \in \mathcal{E}_{s_{v_i}}} \exp(\mathrm{LeakyReLU}(\mathbf{q}_2^{k\top}(\mathbf{h}_{q_{v_i}}^k \odot \mathbf{f}_{j'}^k)))}, \quad (2)$$

where $\mathbf{q}_2^k \in \mathbb{R}^{\frac{d}{K}}$ is the attention vector for the k-th intent.

With the K intent-aware encoders, we can obtain the intent-specific chunks $(\mathbf{h}_{v_i}^{1\,\prime}, \mathbf{h}_{v_i}^{2\,\prime}, ..., \mathbf{h}_{v_i}^{K\,\prime})$ for each item v_i in any given session with the micro-disentanglement.

3.4 Macro-disentanglement

To avoid the redundancy among intent-aware chunked representations, we should encourage the intent-aware chunks to be independent of the macro perspective. Existing works [20, 29] simply regularize the independence in an unsupervised way, which has been demonstrated insufficient to capture semantics [19]. Different from it, we formulate the macro-disentanglement as an intent classification task. Specifically, we predict the intent class with the intent chunks of items $\{\mathbf{h}_{v_i}^{k\,\prime} | v_i \in \mathcal{V}_s\}$ in the given session s, denoted as,

$$\hat{y}_s^p = \mathrm{Softmax}(\mathrm{MLP}(\{\mathbf{h}_{v_i}^{k\,\prime} | v_i \in \mathcal{V}_s\})), \quad (3)$$

where $\hat{y}_s^p$ denotes the predicted probability for all intents and MLP is the one-layer multilayer perceptron. The loss function of intent classification can be formulated as follows,

$$\mathcal{L}_d^s = -\sum_{k=1}^{K} 1_{p=k} \log(\hat{y}_{sk}^p), \quad (4)$$

where $1_{p=k}$ is an indicator function, taken to be 1 when the predicted intent label is correct.

3.5 Session Generation and Prediction

Given a session $s = [v_{s,1}, v_{s,2}, ..., v_{s,n}]$, we have the disentangled intent-specific chunks for each item v_{si} as $(\mathbf{h}_{v_{si}}^{1\,\prime}, \mathbf{h}_{v_{si}}^{2\,\prime}, ..., \mathbf{h}_{v_{si}}^{K\,\prime})$. Following GCE-GNN [31], the reversed position embeddings are introduced to learn the corresponding weights of items under each

Table 1: Statistics of datasets (★ denotes session number).

Dataset	#Clicks	#Train★	#Test★	#Items	Average length
Tmall	818,479	351,268	25,898	40,728	6.69
Last.fm	3,043,614	2,434,242	315,850	35,232	11.76

intent. Similar to [31, 33], we first calculate the item weights γ_i^k under each intent with a soft-attention mechanism as follows,

$$\begin{aligned} \mathbf{z}_i^k &= \tanh(\mathbf{W}_1^k[\mathbf{h}_{v_{si}}^{k}{}' \| \mathbf{p}_{n-i+1}]), \quad \mathbf{h}_{s*}^k = \mathbf{W}_2^k[\mathbf{h}_{v_{si}}^{k}{}' \| \mathbf{h}_{v_{sn}}^{k}{}'], \\ \gamma_i^k &= \mathbf{q}^{k\top}\sigma((\mathbf{W}_3^k[\mathbf{h}_{v_{si}}^{k}{}' \odot \mathbf{h}_{s*}^k]) \| (\mathbf{W}_4^k \mathbf{h}_{s*}^k) \| (\mathbf{W}_5^k \mathbf{z}_i^k) + \mathbf{b}^k), \end{aligned} \tag{5}$$

where $\mathbf{p}_{n-i+1}$ and $\mathbf{h}_{v_{si}}^{k}{}'$ are the reversed position embedding and the k-th intent chunk of item $v_{s,i}$, respectively. Here $\mathbf{W}_1^k, \mathbf{W}_2^k \in \mathbb{R}^{\frac{d}{K}\times\frac{2d}{K}}$, $\mathbf{W}_i^k|_{i=3}^5 \in \mathbb{R}^{\frac{d}{K}\times\frac{d}{K}}$ and $\mathbf{b}^k, \mathbf{q}^k \in \mathbb{R}^{\frac{3d}{K}}$ are the learnable parameters for the k-th intent.

Then, we aggregate the learned k-th intent chunks of items in session s to generate the corresponding intent-specific session representation $\mathbf{h}_s^k$, denoted as,

$$\mathbf{h}_s^k = \sum_{i=1}^n \gamma_i^k \cdot \mathbf{h}_{v_{si}}^{k}{}', \tag{6}$$

where $\mathbf{h}_s^k$ is the representation of session s under k-th intent.

Finally, we calculate the preference score under each intent of session s on candidate item v_i and combine the scores among all intents to get the final preference score, denoted as,

$$p_{si} = \sum_{k=1}^K \mathbf{h}_s^{k\top} \mathbf{h}_{v_i}^k. \tag{7}$$

For session s, $\hat{\mathbf{p}}_s = [\hat{p}_{s1}, \hat{p}_{s2}, ..., \hat{p}_{sm}]$ is the score vector that contains the predicted scores on all m candidate items, and the final probability of each candidate item can be formulated as $\hat{y}_s = \mathrm{Softmax}(\hat{\mathbf{p}}_s)$.

3.6 Model Optimization

To optimize our HIDE model, we leverage a standard cross entropy loss function for each session s, which is defined as follows,

$$\mathcal{L}_r^s = -\sum_{i=1}^m y_{si}\log(\hat{y}_{si}) + (1-y_{si})\log(1-\hat{y}_{si}), \tag{8}$$

where y_s denotes the ground truth label (one-hot vector).

Finally, we unify this recommendation task with the auxiliary intent prediction task mentioned above. The total loss for session s is then defined as,

$$\mathcal{L}^s = \mathcal{L}_r^s + \lambda\mathcal{L}_r^d, \tag{9}$$

where λ denotes the weight to balance the above two tasks.

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 Datasets. We evaluate HIDE on two benchmark datasets (**Tmall**[1] and **Last.fm**[2]) in SBR. **Tmall** contains anonymized shopping logs on Tmall APP. **Last.fm** collects the music listening behavior of users. Following [32, 33], we conduct the same preprocessing step for all methods to ensure a fair comparison. Specifically, we filter out sessions with only one item and items that appear less than 5 times. Moreover, for each session s, we generate sequences and corresponding labels by splitting the original sequence for data augmentation. The statistics of datasets are shown in Table 1.

[1]https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
[2]http://mtg.upf.edu/static/datasets/last.fm/lastfm-dataset-1K.tar.gz

Table 2: Benchmark evaluation ("M" refers to MRR).

Method	Tmall Dataset				Last.fm Dataset			
	P@10	**M@10**	**P@20**	**M@20**	**P@10**	**M@10**	**P@20**	**M@20**
Item-KNN	6.68	3.12	9.20	3.34	11.89	4.42	15.27	4.79
FPMC	13.05	7.11	16.08	7.34	8.53	3.56	12.91	3.85
GRU4Rec	9.50	5.75	10.98	5.92	12.85	5.18	14.94	5.57
NARM	19.21	10.39	23.35	10.68	14.64	6.39	18.07	6.73
STAMP	22.64	13.08	26.44	13.35	13.97	6.26	17.23	6.68
SR-GNN	23.49	13.47	27.65	13.76	15.53	6.74	19.74	7.18
FGNN	20.64	10.05	25.27	10.41	14.86	6.51	18.95	6.94
GCE-GNN	<u>28.03</u>	<u>15.07</u>	<u>33.41</u>	<u>15.43</u>	16.38	<u>7.17</u>	22.03	<u>7.63</u>
DHCN	26.24	14.63	31.51	15.08	<u>16.58</u>	7.02	<u>22.29</u>	7.49
SHARE	25.14	14.13	30.46	14.57	15.57	6.68	19.87	7.01
HIDE	**31.10**	**16.77**	**37.12**	**17.19**	**18.65**	**8.11**	**25.15**	**8.56**
Improv.	10.95%	11.28%	11.10%	11.41%	12.48%	13.11%	12.83%	12.19%

4.1.2 Metrics. Following [31, 32], we use two ranking metrics, *i.e.*, Precision@K (P@K) and MRR@K (M@K), for evaluation.

4.1.3 Baselines. We compare HIDE with the following methods, including: two *conventional methods* (Item-KNN [25] and FPMC [23]); three *sequential-based methods* (GRU4Rec [11], NARM [16] and STAMP [18]); three *GNN-based methods* (SR-GNN [32], FGNN [22] and GCE-GNN [31]) and two *hypergraph-based models* (DHCN [33] and SHARE [28]).

4.1.4 Hyper-parameters Settings. Following [32, 33], we set the item embedding size, the batch size and the L_2 penalty as 100, 100, and $1e^{-5}$, respectively. To optimize all methods, we use Adam optimizer with an initial learning rate of 0.001. For all baselines, we carefully tuned the hyper-parameters based on the original papers. For HIDE, we set the sparsity coefficient ε to 0.4. We conduct careful grid search for hyper-parameters (*i.e.* maximum window size W, intent number K and balanced weight λ), and use the best settings in our paper (we finally set $W = 6, K = 5, \lambda = 1e^{-4}$).

4.2 Overall Performance

Table 2 shows the overall performance, from which we have the following conclusions.

- **Our HIDE achieves the best performance.** HIDE significantly outperforms all baselines, outperforming the best baseline by 11.18% on Tmall and 12.65% on Last.fm on average. Moreover, HIDE outperforms the hypergraph-based methods (*i.e.*, DHCN and SHARE), which verifies the effectiveness of our designs for intent disentanglement to enhance the hypergraph model in SBR.
- **Sequential-based methods outperform the conventional models.** Sequential-based methods (*i.e.* NARM and STAMP) significantly outperform the conventional models (*i.e.* FPMC), which stresses the key role of the sequential modeling in SBR. That is, it verifies the necessity of our HIDE to appropriately model the sequential patterns with transition hyperedges.
- **GNN-based methods outperform sequential methods.** The performance gain of GNN-based models confirms GNN's remarkable capacity in SBR. Among them, GCE-GNN (capturing the chronological order) is the best baseline on Tmall while DHCN (modeling high-order relations) performs better in Precision on Last.fm. It demonstrates both chronological order and high-order relations are crucial in SBR , which validates our motivation

Table 3: Ablation study of the key designs of HIDE.

Model		Tmall P@20	Tmall M@20	Last.fm P@20	Last.fm M@20
Multi-type Hyperedges	w/o T	36.88	16.87	22.43	7.51
	w/o C	35.02	14.96	23.64	7.92
	w/o I	36.96	16.95	24.87	8.32
	All	**37.12**	**17.19**	**25.15**	**8.56**
Micro-Disentanglement	HyperGAT	33.14	15.26	23.51	8.24
	HyperGCN	34.05	15.57	24.17	8.47
	HGAConv	**37.12**	**17.19**	25.15	**8.56**
Macro-Disentanglement	w/o Prediction	36.73	16.81	24.68	8.29
	w Prediction	**37.12**	**17.19**	**25.15**	**8.56**
Session Generation	HIDE-NP	32.78	15.14	22.34	7.38
	HIDE-NA	35.04	15.82	21.89	7.45
	HIDE	**37.12**	**17.19**	**25.15**	**8.56**

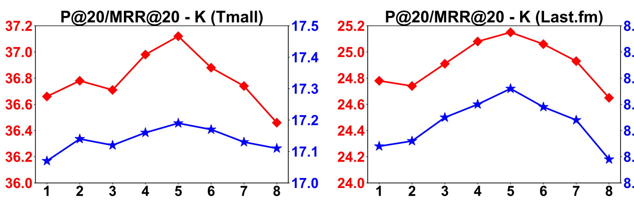

Figure 2: The impact of intent number K.

to model these high-order relations in session with multi-type hyperedges from three perspectives.

4.3 Ablation Study

To evaluate the effectiveness of several key designs in HIDE, we performed ablation studies as shown in Table 3.

1) Multi-type Hyperedges. We compare the performance without Transition Hyperedge (w/o T), without Context Hyperedge (w/o C), without intent Hyperedge (w/o I), and with all hyperedges, respectively. The results show that removing any type of hyperedges will lead to performance degradation.

2) Micro-disentanglement. We compare the performance of models with different propagation mechanisms, *i.e.*, HyperGAT [28], HyperGCN [8] and our designed HGConv. The results in Table 3 show that our designed HGConv achieves the best performance.

3) Macro-disentanglement. We remove the intent prediction task, and the performance drops significantly, which verifies the intent prediction task contributes to better performance.

4) Session Generation. We compare the models without position embedding (HIDE-NP), without soft attention (HIDE-NA), and HIDE, respectively. The results show that both position embedding and soft attention enhance the recommendation performance.

4.4 Hyper-parameter Study

As for the hyper-parameter study, we discuss the impact of one of the most critical hyper-parameters, intent-number K. Specifically, we vary K from 1 to 8 to study its impact. As Figure 2 shows, the performance of HIDE achieves the best when $K = 5$ and then drops when $K > 5$. Hence, we set K as 5 for all datasets to ensure promising performance.

4.5 Disentanglement Analysis

We further analyze the impact of intent disentanglement on session length. Specifically, we divide all sessions into five groups based

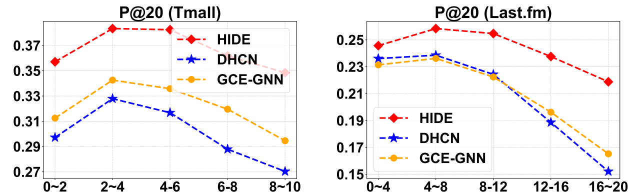

Figure 3: Performance with different session lengths.

on the length. For each group, we compare the performance of our HIDE with two SOTA baselines (GCE-GNN and DHCN) and present the P@20 performance. From the results in Figure 3, with longer sessions, GCE-GNN and DHCN suffer from significant performance drop, and the performance gap (between HIDE and baselines) becomes larger, which verifies that HIDE can effectively model user intents in a longer session (more complex intents).

5 RELATED WORKS

Session-based Recommendation (SBR). Earlier works [23, 25] on SBR foucs on the similarity of items. Then, Sequential-based methods [11, 16, 18] model the session with GRU [6] or attention layers [27]. Recently, the GNN-based methods [21, 22, 26, 28, 31–34] propose to build a session graph and capture the complex relations among items with graph neural network (GNNs) [26].

Hypergraph Learning. Hypergraph [1, 17] extends the capability of high-order relations with hyperedge, which can connect more than two nodes. HGNN [8] extends graph convolution to hypergraph. HyperGAT [7] proposed hypergraph attention networks and HGC-RNN [13] proposed dynamic hypergraph neural networks for the hypergraph learning.

Disentangled Representation Learning. Early studies [2, 3, 5, 12, 14] are based on variational auto-encoders (VAE) [15]. Recently, several methods [17, 20, 30, 35, 36] are proposed to disentangle the multiple latent factors in graph-based data mining, such as DGCF [29]. However, the above disentangled methods are not applicable to the session-based recommendation (SBR).

6 CONCLUSION AND FUTURE WORK

In this work, we propose a hypergraph-based solution HIDE for the session-based recommendation, which captures the high-order relations and latent intents in item transitions. Specifically, we first perform intent-aware embedding propagation to learn the disentangled intent chunks of each item click with micro-disentanglement; we then encourage the independence of each intent-specific chunk with intent classification from a macro perspective. Benchmark evaluations verify the effectiveness of our HIDE model and the necessity of the modeling of disentangled intents. As for future work, further A/B tests in real-world recommender systems can verify HIDE’s effectiveness in industrial applications.

ACKNOWLEDGMENT

This work is supported in part by National Natural Science Foundation of China under U1936217, 61972223 and 61971267. This work is also supported in part by National Key Research and Development Program of China under 2020AAA0106000.

REFERENCES

[1] Sameer Agarwal, Kristin Branson, Serge Belongie, and et al. 2006. Higher order learning with graphs. In *ICML*. 17–24.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.

[3] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding disentangling in *beta*-VAE. *arXiv preprint arXiv:1804.03599* (2018).

[4] Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Bundle recommendation with graph convolutional networks. In *Proceedings of the 43rd international ACM SIGIR conference on Research and development in Information Retrieval*. 1673–1676.

[5] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942* (2018).

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[7] Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be More with Less: Hypergraph Attention Networks for Inductive Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 4927–4936.

[8] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3558–3565.

[9] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2021. Graph neural networks for recommender systems: Challenges, methods, and directions. *arXiv preprint arXiv:2109.12843* (2021).

[10] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[11] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[12] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-vae: Learning basic visual concepts with a constrained variational framework. (2016).

[13] Jianwen Jiang, Yuxuan Wei, Yifan Feng, Jingxuan Cao, and Yue Gao. 2019. Dynamic Hypergraph Neural Networks.. In *IJCAI*. 2635–2641.

[14] Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. In *International Conference on Machine Learning*. PMLR, 2649–2658.

[15] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[16] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[17] Yinfeng Li, Chen Gao, Quanming Yao, Tong Li, Depeng Jin, and Yong Li. 2022. DisenHCN: Disentangled Hypergraph Convolutional Networks for Spatiotemporal Activity Prediction. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE.

[18] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1831–1839.

[19] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*. PMLR, 4114–4124.

[20] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. *arXiv preprint arXiv:1910.14238* (2019).

[21] Zhiqiang Pan, Fei Cai, Wanyu Chen, Honghui Chen, and Maarten de Rijke. 2020. Star graph neural networks for session-based recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1195–1204.

[22] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 579–588.

[23] Steffen Rendle, Christoph Freudenthaler, Lars Schmidt-Thieme, and et al. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[24] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[25] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[26] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks* 20, 1 (2008), 61–80.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[28] Jianling Wang, Kaize Ding, Ziwei Zhu, and James Caverlee. 2021. Session-based Recommendation with Hypergraph Attention Networks. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 82–90.

[29] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1001–1010.

[30] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming Zhang. 2020. DisenHAN: Disentangled Heterogeneous Graph Attention Network for Recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1605–1614.

[31] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 2020. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 169–178.

[32] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 346–353.

[33] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4503–4511.

[34] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation.. In *IJCAI*, Vol. 19. 3940–3946.

[35] Yu Zheng, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2022. Disentangling Long and Short-Term Interests for Recommendation. In *TheWebConf*.

[36] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. In *Proceedings of the Web Conference 2021*. 2980–2991.