Knowledge Graph-based Session Recommendation with Session-Adaptive Propagation

Yu Wang
yu.wang.1@vanderbilt.edu
Vanderbilt University
Nashville, TN, USA

Amin Javari
amin_javari@homedepot.com
The Home Depot
Atlanta, GA, USA

Janani Balaji
janani_balaji@homedepot.com
The Home Depot
Atlanta, GA, USA

Walid Shalaby
w.fouad.cs@gmail.com
The Home Depot
Atlanta, GA, USA

Tyler Derr
tyler.derr@vanderbilt.edu
Vanderbilt University
Nashville, TN, USA

Xiquan Cui
xiquan_cui@homedepot.com
The Home Depot
Atlanta, GA, USA

ABSTRACT

Session-based recommender systems (SBRSs) predict users' next interacted items based on their historical activities. While most SBRSs capture purchasing intentions locally within each session, capturing items' global information across different sessions is crucial in characterizing their general properties. Previous works capture this cross-session information by constructing graphs and incorporating neighbor information. However, this incorporation cannot vary adaptively according to the unique intention of each session, and the constructed graphs consist of only one type of user-item interaction. To address these limitations, we propose knowledge graph-based session recommendation with session-adaptive propagation. Specifically, we build a knowledge graph by connecting items with multi-typed edges to characterize various user-item interactions. Then, we adaptively aggregate items' neighbor information considering user intention within the learned session. Experimental results demonstrate that equipping our constructed knowledge graph and session-adaptive propagation enhances session recommendation backbones by 10%-20%. Moreover, we provide an industrial case study showing our proposed framework achieves 2% performance boost over an existing well-deployed model at The Home Depot e-platform.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**.

KEYWORDS

Session Recommendation, Knowledge Graph, Adaptive Propagation

ACM Reference Format:
Yu Wang, Amin Javari, Janani Balaji, Walid Shalaby, Tyler Derr, and Xiquan Cui. 2024. Knowledge Graph-based Session Recommendation with Session-Adaptive Propagation. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3589335.3648324

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0172-6/24/05…$15.00
https://doi.org/10.1145/3589335.3648324

1 INTRODUCTION

Transformer-based models have shown state-of-the-art performance for session-based recommender systems (SBRSs) by leveraging their attention mechanisms and deep learning capabilities [9, 41]. While transformers are adept at capturing local session data and individual item preferences [4, 15], they are limited in their ability to capture global transitional patterns among items [12, 23, 44]. This limitation has sparked research into hybrid graph-based SBRSs that combine the strengths of transformers with Graph Neural Networks (GNNs) to capture both local and global dependencies [12, 23, 39, 44]. For example, in Figure 1(b), borrowing the transitional information between flower and lopper in Figure 1(a) helps characterize the intent of session A as decorating the garden and hence increases the probability of predicting the next item to be garden-related, e.g., watering can.

However, the existing hybrid models face two significant challenges. Firstly, the lack of full connectivity between GNN and transformer models limits their ability to capture the session dynamics [22, 37, 40]. While GNNs excel at capturing item-item relationships, they often struggle to incorporate the broader session context, and the learned item representations lack full awareness of the session context. For example, in Figure 1, although both sessions A and B have involved the flower, their intentions are quite different: one for decorating the garden while the other for decorating the kitchen. In contrast to blindly aggregating all neighbors' information to the flower without considering the session context [12, 23, 44], our approach learns session-aware item embeddings by selectively propagating information from relevant neighbors based on the current session. For instance, when determining which neighbors' information is aggregated to the flower, we recommend incorporating the lopper and watering can when decorating the garden in (b), while the sink and the table when decorating the kitchen in (c).

Secondly, the global transitional patterns among items in these hybrid models are typically constructed based on one type of interaction, such as co-purchase patterns [12, 22, 23, 37]. However, on e-commerce platforms [40], items could form multiple relationships: substitution items are typically co-viewed and complementary items are typically co-add-to-carted (co-ATC) [21, 36] by the same user. Uniformly using neighbor information may result in the dilution of diverse relationships among items and consequentially lead to unsatisfactory recommendations for users.

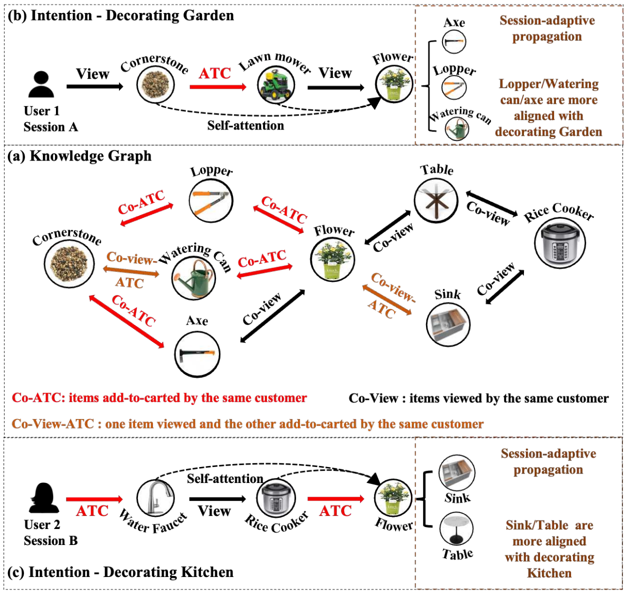

(b) Intention - Decorating Garden
User 1
Session A
View
Cornerstone
ATC
Lawn mower
View
Flower
Self-attention
Axe
Lopper
Watering can
Session-adaptive propagation
Lopper/Watering can/axe are more aligned with decorating Garden
(a) Knowledge Graph
Lopper
Co-ATC
Co-ATC
Table
Co-view
Rice Cooker
Flower
Co-view
Cornerstone
Co-view-ATC
Watering Can
Co-ATC
Co-view-ATC
Sink
Co-view
Co-ATC
Axe
Co-view
Co-ATC: items add-to-carted by the same customer
Co-View : items viewed by the same customer
Co-View-ATC : one item viewed and the other add-to-carted by the same customer
User 2
Session B
ATC
Water Faucet
Self-attention
View
Rice Cooker
ATC
Flower
Sink
Table
Session-adaptive propagation
Sink/Table are more aligned with decorating Kitchen
(c) Intention - Decorating Kitchen

Figure 1: Since the user in Session A (b) intends to decorate the garden, while the user in Session B (c) intends to decorate the kitchen, the corresponding neighbors from the knowledge graph (a) are different for the same flower. By our proposed session-adaptive propagation, the flower aggregates more information from the lopper/watering can in (b) while more information from the sink/table in (c).

To overcome the above two challenges, we propose a knowledge graph-based SBRS with session-adaptive propagation. Our contributions are summarized as follows:

- For the first challenge, we propose a session-adaptive graph propagation to adaptively aggregate items' neighbor information based on the session contexts obtained by the transformer model.
- For the second challenge, we construct an item knowledge graph by extracting three different item co-relations with users. Additionally, we employ the heterogeneous graph transformer [11] in the message-passing design to effectively aggregate neighborhood information based on the relationship type.
- We experimentally verify the proposed framework in enhancing existing SBRs, compare the performance improvement caused by different types of edges, and verify the session-adaptive propagation by visualizing the changing attentions of the same item in aggregating neighbors in different sessions.

2 RELATED WORK

Earlier works in SBRSs leverage Markov Chains to infer the conditional probability of an item based on the previously interacted items [24, 26]. More recent works have resorted to deep learning for session recommendation. Recurrent neural networks (RNNs) [8, 41] such as GRU4Rec [9] and Transformers (TFs) [4] such as SASRec [15] have been developed to model the interactions among adjacent items in a session. To capture even more complex transitional patterns, graph-based methods such as SR-GNN [39] extract the session graph for each session and use a gated GNN to learn session embeddings. Different from previous works that only capture transitional patterns within the session, we construct a global graph with different types of edges to capture even more broad information, the related works of which are reviewed next.

Prior work explore global transitional patterns across different sessions by querying the global item graph [12, 23, 44]. [12] designs a global context-enhanced inter-session relation encoder to capture the inter-session item-wise dependencies. [44] constructs the dual session graph to model the pair-wise transition relationship between items based on the global connections. [23] constructs the global graph by merging all individual session graphs. Recently, KSTT [43] uses an item-category knowledge graph for session recommendation. However, these models learn item embeddings from the global graph without session-tailored modification. Only GCE-GNN [37] and GCARM [22] consider session adaptation in aggregating neighbors' information. However, GCE-GNN quantifies the importance of neighbors based on their similarity to the whole session without differentiating central items. GCARM treats all transitions similarly without distinguishing different interaction types. To handle these issues, we design a session-adaptive propagation to query neighbors based on session contexts and interaction types. The constructed KG in this work has only one type of node, the item, and we will consider other node types in the future. We put a more comprehensive review of related works in Appendix A.1.

3 THE PROPOSED FRAMEWORK

Our framework, as illustrated in Figure 2, comprises of a GNN that obtains item embeddings by adaptively aggregating information from neighboring items based on the target session, and a transformer model that acquires session embeddings for predicting the next item. In the subsequent sections, we first explain the construction of the item knowledge graph, and then the GNN-based message passing model and the transformer-based prediction model.

3.1 Item Knowledge Graph Construction

As items typically exhibit two types of correlations, substitution and complementary [44], we extract three distinct types of edges, as depicted in Figure 2(a)-(c), by examining whether two items co-occur within the same session: co-view, co-ATC, and co-view-ATC edges. For instance, in the first session shown in Figure 2(a), the user first views the cornerstone, then adds the flower to the cart, and subsequently views the lawn mower. This sequence forms three edges: the co-view edge between the cornerstone and the lawn mower, the co-view-ATC edge between the cornerstone and the flower, and the co-view-ATC edge between the flower and the lawn mower. More formally, we define the edge weight from item v_j to v_i of type co-t_1-t_2 as follows:

$$w_{i\leftarrow j}^{(t_1,t_2)} = \frac{\sum_{m=1}^{M} \mathbb{1}(v_i, v_j \in \mathcal{S}_m,\ \tau(v_i,\mathcal{S}_m)=t_1,\ \tau(v_j,\mathcal{S}_m)=t_2)}{\sum_{m=1}^{M} \mathbb{1}(v_i \in \mathcal{S}_m,\ \tau(v_i,\mathcal{S}_m)=t_1)}, \quad (1)$$

where $\mathcal{S}_m$ is the m^{th}-session, M is the total number of sessions in the historical data, and $t_1, t_2 \in \mathcal{T} = \{\text{view}, \text{ATC}\}$ and $\mathbb{1}$ is an indicator function. Specifically, $\mathbb{1}(v_i \in \mathcal{S}_m, v_j \in \mathcal{S}_m, \tau(v_i,\mathcal{S}_m) = t_1, \tau(v_j,\mathcal{S}_m) = t_2) = 1$ if **(a)** both v_i, v_j belong to the m^{th} session $\mathcal{S}_m$, **(b)** user interacts with v_i following type t_1 and **(c)** interacts with v_j following type t_2 in $\mathcal{S}_m$ and otherwise 0. Note that, we normalize

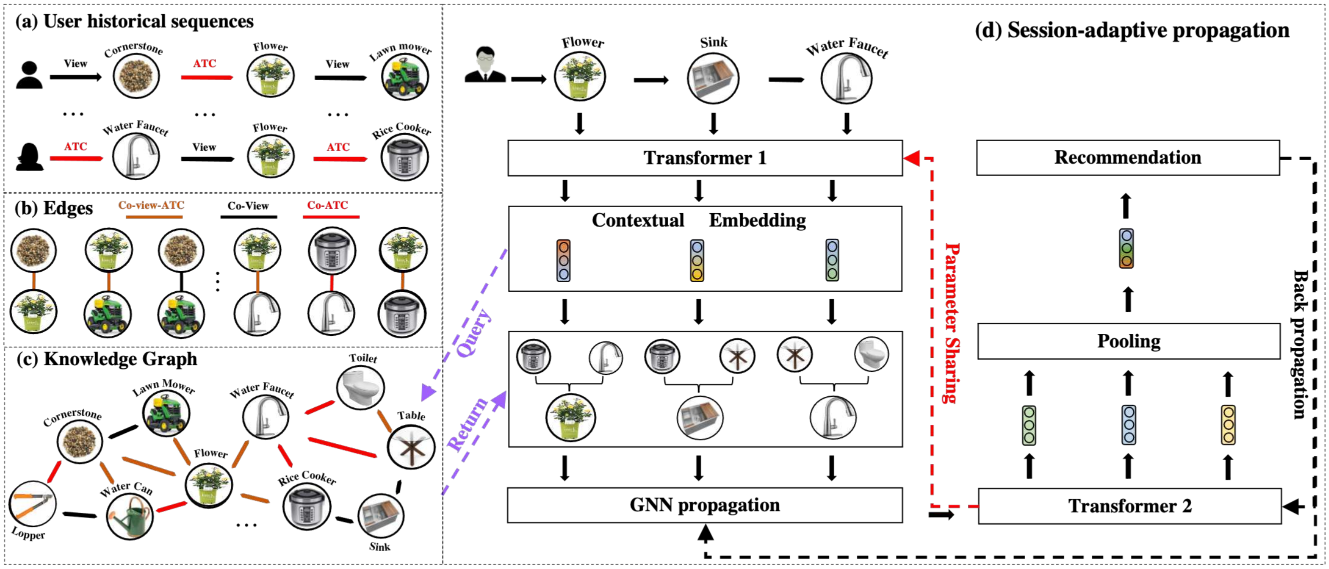


Figure 2: In (a)-(c), we first extract three types of edges from historical sessions to construct item knowledge graph. Then in (d), we forward the given session through the 1^{st} transformer layer to obtain items' contextual embeddings, which are used for query-relevant neighbors for GNNs to perform graph propagation. The propagated item embeddings are fed into the 2^{nd} transformer with a pooling layer afterward to obtain session embedding for the recommendation.

edge weights based on the degree of the head node to avoid popular items from dominating the message-passing of GNNs [6, 29].

Based on our empirical experiments, it was observed that the obtained graph may contain items with over a hundred neighbors. Therefore, in order to mitigate the heavy computational issue [42] and prevent over-smoothing [3] during the message passing, we sparsify the graph by retaining only the top-K neighbors for each neighborhood type, based on the acquired edge weights. The statistics of the constructed networks are provided in Table 1.

3.2 Session-adaptive Propagation

As mentioned earlier, a straightforward approach to utilize item graphs in SBRSs is to employ a GNN model to obtain item embeddings from the graph. These acquired embeddings can then be utilized in a transformer-based model to predict the next item based on the target session [12, 23, 44].

However, seen in Figure 1, a key limitation of this architecture is that the contribution of each neighboring item remains unchanged in obtaining the center item's embedding, regardless of the session from which the item originates. As such, the item embeddings obtained from the GNN is unaware of the contextual information specific to the target session. This approach is sub-optimal since the purchasing intention associated with the same item naturally differs across different sessions, thereby necessitating changes in the selection of compatible neighbors based on the session-specific intention. Motivated by this observation, we propose a session-adaptive propagation that dynamically propagates neighbor information according to the item's unique context within each session.

To implement the above idea, given the target session, we input the initial item embedding within the session into a transformer, resulting in the generation of its session-aware representation. Subsequently, this obtained embedding is utilized to determine the edge weights for the GNN model. Finally, following the message-passing process, the updated item embeddings are once again fed into another transformer model to make the final prediction.

More formally, we first obtain item initial embeddings $\mathbf{E}^1$ by integrating item meta-attributes such as item title and category. Then, given an item v_i in the session $\mathcal{S}_m$, we use the item initial embeddings $\mathbf{E}^1$ to obtain its contextual embedding $\mathbf{c}_i^m$:

$$\mathbf{c}_i^m = transformer_1(v_i, \mathbf{E}^1, \mathcal{S}_m), \quad \forall v_i \in \mathcal{S}_m \tag{2}$$

where $transformer_1$ is the 1^{st} transformer. With the contextual embedding $\mathbf{c}_i^m$ as the query, inspired by [11], we perform heterogeneous graph transformer-based propagation to adaptively aggregate v_i's neighbors' information relevant to the current session intention:

$$\mathbf{h}_i^{m,l} = \frac{1}{|\mathcal{T}|^2} \sum_{t \in \mathcal{T} \times \mathcal{T}} ||_{h=1}^{H} \sum_{v_j \in \mathcal{N}_i^t} \alpha_{i \leftarrow j}^{h,l,t,m} \mathbf{V}^{h,l,t} \mathbf{h}_j^{m,l-1} \tag{3}$$

$$\alpha_{i \leftarrow j}^{h,l,t,m} = \frac{(\mathbf{Q}^{h,l,t} \mathbf{h}_i^{m,l-1})^\top (\mathbf{K}^{h,l,t} \mathbf{c}_j^m)}{\sqrt{d/H}}, \forall v_i \in \mathcal{S}_m, \mathcal{S}_m \in \mathcal{S}, \tag{4}$$

where $\alpha_{i \leftarrow j}^{h,l,t,m}$ denotes the graph attention from item v_j to v_i under the head h, edge type t at layer l. $\mathbf{Q}^{h,l,t}, \mathbf{K}^{h,l,t}, \mathbf{V}^{h,l,t}$ represent the query, key, and value matrix at the head h, edge type t, layer l of graph attention. H is the total number of heads. After L layers graph transformer-based propagation, we obtain the final item embeddings $\mathbf{h}_i^{m,L}$. *Since different session contexts provide different contextual embeddings* $\mathbf{c}_i^m \neq \mathbf{c}_i^{m'}$, *the calculated attention coefficients would also be different, i.e.,* $\alpha_{i \leftarrow j}^{h,l,t,m} \neq \alpha_{i \leftarrow j}^{h,l,t,m'}$. *Then* $\mathbf{h}_i^{m,L}$ *and* $\mathbf{h}_i^{m',L}$) *would be different, and they would only include the neighborhood information that is relevant to the item's unique intention provided by the corresponding session context* $\mathcal{S}_m$ ($\mathcal{S}_{m'}$).

To demonstrate the effectiveness of our proposed session-adaptive propagation in learning neighborhood attention based on the session context, we conduct a case study that visualizes the learned neighborhood attention for the same item across different sessions is indeed different. Further details are presented in Section 4.7.

3.3 Prediction and Optimization

After L layers graph propagation, the item embeddings, e.g., $\mathbf{h}_i^{m,L}, \forall v_i \in \mathcal{S}_m$, are fed into $transformer_2$ with mean pooling to obtain the session embedding $\mathbf{S}_m$ and then we compute the cross-entropy loss to optimize the whole framework:

$$\mathcal{L} = \sum_{m=1}^{M} \sum_{i=1}^{|\mathcal{V}|} \mathrm{Y}_{m,i} \log \hat{\mathrm{Y}}_{m,i}, \quad \mathrm{S}_m = transformer_2(\{\mathrm{h}_i^{m,L} | v_i \in \mathcal{S}_m\}) \quad (5)$$

where $\mathcal{V}$ is the total item space, $\mathrm{Y}_m \in \{0, 1\}^{|\mathcal{V}|}$ is the one-hot encoded next ground-truth item of the session $\mathcal{S}_m$, $\hat{\mathrm{Y}}_m = \sigma(p(\mathrm{S}_m)) \in \mathbb{R}^{|\mathcal{V}|}$ is the predicted probability distribution of the next item for the session $\mathcal{S}_m$ from the linear prediction head p followed by the softmax normalization σ. Note that S_m could be used in other tasks if paired with corresponding prediction heads, such as predicting the category/price of the next item as discussed in Section 4.4. The parameters of $transformer_2$ are optimized via back-propagation, and we periodically synchronize the parameters from $transformer_1$ to $transformer_2$ [7]. During the inference stage, we make the next item prediction by finding the item $i = \arg\max_{j \in \mathcal{V}} \mathrm{Y}_{m,j}$.

4 EXPERIMENTS

In this section, we verify the effectiveness of the proposed knowledge graph construction method and the session-adaptive propagation mechanism through extensive experiments. We first introduce the datasets and the experimental settings.

4.1 Experimental Setup

4.1.1 Datasets. We conduct experiments on the following datasets:

- **Diginetica**[1] comes from CIKM Cup 2016. We follow the same preprocessing as [39]: sessions in the last and second-to-last week are used as testing and validation data. We filter out sessions of length less than 1 and items appearing less than 5 times. We extract item co-purchase edges from all sessions in train-purchase.csv and item co-view edges from only training/validation sessions in train-item-view.csv to avoid data leakage. The meta-attribute includes item title, category, and price.
- **Yoochoose 1/64**[2] comes from the RecSys Challenge 2015. Sessions on the last and second last day are used as testing and validation data. We filter out sessions of length less than 1 and items appearing less than 5 times. We extract item co-purchase edges from all sessions in yoochoose-buys.dat and item co-view edges from only training/validation sessions in yoochoose-clicks.dat to avoid data leakage. Since items in this dataset have only one category attribute and it represents different meanings, e.g., 1-12 for item real categories, 'S' for special offer, and 8-10 digit numbers for item brand, we only use item ID to demonstrate the effectiveness of session-adaptive heterogeneous propagation.

Table 1: Statistics of datasets used for experiments and their corresponding knowledge graphs.

Dataset	Train/Val/Test Seqs	# Edges	# Nodes	Sparse Seqs	Meta data
Diginetica	675,673/ 43,541/ 68,571	1,576,571	123,273	13,867	Title/Category/Price
Yoochoose 1/64	369,142/ 45,864/ 55,898	1,025,176	52,739	15,385	–
The Home Depot (THD)	3,169,140/ 672,873/ 672,873	4,162,712	1,317,149	227,941	Title/Category/Brand/ Color/Manufacturer/Class/ Department

- **THD is a real industrial-level dataset** from The Home Depot, the largest home improvement retailer in the USA. We sample 3,169,140/672,873/672,873 Add-to-Cart (ATC) sessions chronologically for constructing the train/valid/testing data. The multiplex graph is constructed by extracting co-view, co-ATC, and co-view-ATC edges. Items in this dataset have 7 meta-attributes: product title, hierarchical categories (i.e., L1, L2, L3, Leaf), brand, manufacturer, color, department, and class name.

Statistics of the three datasets are summarized in Table 1. Note that sparse sessions refer to the ones containing items appearing less than 5 times in the whole dataset. These sessions are used to evaluate our framework on sessions including cold-start items.

4.1.2 Item embedding initialization. We initialize item embeddings based on their meta-attributes. We use different embedding techniques to extract different types of item metadata [25], which can be numerical, categorical, and textual. For each numerical attribute, we directly embed it as a real-valued number. For each categorical attribute, we initialize a unique learnable embedding matrix. For textual attributes such as title and description, we first construct the token embedding matrix and then the title/description embedding is computed by mean pooling over the embeddings of corresponding tokens in that sentence. Note that pre-trained NLP models are not preferred here to avoid capturing noisy semantic signals. For example, even though silver sinks and creamy white stones share semantic-similar colors, they are essentially purposed for decorating different rooms. We empirically observe that utilizing this token-based embedding avoids capturing noisy semantic signals since they correspond to different tokens. We concatenate different types of embeddings to form the final item meta-embeddings and feed them into the transformer model.

4.1.3 Backbones. Note that our constructed knowledge graph and the proposed session-adaptive propagation can be applied to enhance any embedding-based SBRS. To demonstrate this, we select three fundamentally different but representative backbones and equip them with our framework:

- **GRU4Rec** [9]: The very first model leveraging RNNs to characterize item sessions for session recommendation.
- **SASRec** [15]: The very first model leveraging the self-attention from the transformer to draw context from all user-item interactions in the same session.
- **KGHT** [40]: A graph-based model constructing the item relational graph and leveraging graph attention to capture item relations for recommendations. Since this work is not initially designed for session recommendation, we modify it to align with our problem setting.

[1] https://competitions.codalab.org/competitions/11161
[2] https://www.kaggle.com/datasets/chadgostopp/recsys-challenge-2015

Table 2: Performance comparison (%) of utilizing meta-attribute embedding layer and session-adaptive heterogeneous propagation layer. The best and runner-up are in bold and underlined. Note that N(M)@10 represents NDCG(MRR)@10.

Backbone		Diginetica				Yoochoose 1/64				THD				Increase (↑)
		N@10		M@10		N@10		M@10		N@10		M@10		
		Full	Sparse	Full	Sparse	Full	Sparse	Full	Sparse	Full	Sparse	Full	Sparse	
RNN	GRU4Rec	12.66	11.04	9.55	8.19	32.77	28.32	26.52	22.81	14.40	7.66	12.34	6.53	–
	GRU4Rec$_m$	12.38	10.62	9.27	7.77	–	–	–	–	17.09	10.42	14.19	8.41	10.55%
	GRU4Rec$_m$*	13.01	11.47	9.73	8.37	–	–	–	–	18.13	11.33	15.23	9.34	18.88%
Trans-former	SASRec	14.70	13.00	11.11	9.57	33.68	28.59	27.16	22.87	15.27	8.11	12.97	6.83	–
	SASRec$_m$	14.68	12.79	11.08	9.28	–	–	–	–	17.87	11.11	14.87	9.01	11.94%
	SASRec$_m$*	15.31	13.68	11.60	10.12			–	–	18.53	11.72	15.53	9.64	18.28%
Graph	KGHT	17.73	15.99	13.33	11.74	34.81	30.14	28.04	24.10	16.59	9.29	14.02	7.74	–
	KGHT$_m$	17.70	15.78	13.27	11.52	–	–	–	–	18.47	11.68	15.49	9.59	8.45%
	KGHT$_m$*	17.81	16.88	13.38	12.49	–	–	–	–	18.78	11.89	**15.74**	9.79	11.59%
	KGHT$_q$	18.64	16.74	13.97	12.28	**35.80**	**31.71**	**28.80**	**25.21**	17.30	10.20	14.45	8.35	4.25%
	KGHT$_{qm*}$	**18.93**	**17.16**	**14.22**	**12.65**	–	–	–	–	**18.83**	**12.17**	15.67	**9.91**	14.10%

X: Backbone X using item ID but no meta-attributes; **X**$_m$: Backbone X using item meta-attributes but no ID; **X**$_{m*}$: Backbone X using both item meta-attributes and ID; **X**$_q$: Backbone X using Graph and Session-adaptive propagation; **X**$_{qm*}$: Backbone X using Graph, Session-adaptive propagation, and meta-attribute.

Equipping each of the above three backbones with the item meta-attribute embedding layer and the session-adaptive propagation layer, we end up with 11 model configurations. We name each new configuration by combining the name of its backbone and the equipped techniques, e.g., GRU4Rec$_m$ denotes the backbone of GRU4Rec with item meta-attribute embedding layer, SASRec$_{m*}$ denotes the backbone of SASRec with both item meta-attribute and ID, and KGHT$_q$ denotes the backbone of KGHT equipped with session-adaptive propagation layer. The architecture of each model configuration is summarized in Table 3.

4.1.4 Evaluation Metric and Implementation Details. Following [13, 38], we report the average of NDCG@10 (N@10) and MRR@10 (M@10) over all sequences in the test set. We assign a dedicated embedding layer for each attribute, and the embedding dimension is determined based on the total number of distinct tokens of the corresponding attribute vocabulary. To ensure a fair comparison, we tune the following hyperparameters for each model individually: the dimension of ID embedding layer $\{64, 128, 256\}$, the number of layers for self-attention and graph propagation $\{1, 2, 3\}$, the number of attention heads $\{1, 4, 8\}$, the dimension of hidden embeddings $\{100, 512\}$, learning rate $\{1e^{-4}, 1e^{-3}, 1e^{-2}\}$, the L2 penalty $\{0, 5e^{-4}\}$, training epochs $\{100, 200\}$, the batch size $\{100, 512\}$, dropout ratio $\{0.1, 0.25\}$ across all models. We save the model performing best on validation sessions and evaluate it on testing sessions. Training epochs for Diginetica/Yoochoose/THD are set to be 200/200/30. We synchronize the parameters of $transformer_2$ from $transformer_1$ every epoch.

4.2 Model Configuration Analysis

To demonstrate the effectiveness of the proposed knowledge graph and session-adaptive propagation, we compare three backbones GRU4Rec, SASRec, and KGHT with their corresponding enhanced versions in Table 2. For brevity, we represent any of the three backbones as X in the following text:

- Compared with X, X_{m*} incorporates item meta-attributes and improves the performance by around 12% – 19% on average. This is because ID-based embedding only captures the topological proximity of each item to all other items and cannot provide generalizability, especially when items' topological information is noisy/sparse. Leveraging meta-attributes alleviates this issue by transferring the learned information among items sharing the same meta-attribute. Since THD has more abundant types of well-curated meta-attributes than the ones of Diginetica, as evidenced in Table 1, X_{m*} achieves an even larger performance gain over X on THD than on Diginetica.
- Compared with X, X_m achieves comparable and sometimes slightly worse performance on Diginetica, e.g., 9.55% for GRU4Rec while 9.27% for GRU4Rec$_m$ in MRR@10. This is because solely relying on meta-attributes to represent items may lose topological information in the sessions. Two items sharing the same meta-attributes will be encoded the same, even though they may be involved in significantly different sessions. This is also evidenced by the better performance of X_m* than X_m after combining both item ID and item meta-attribute. Different from Diginetica, X_m always achieves higher performance than X on THD because more abundant meta-attributes there enable the concatenated embeddings to be more unique and hence can somewhat mimic the function of Item ID in capturing topological information embedded in the sessions.

Table 3: An architecture comparison of different backbones.

Backbone	ID	Meta-attribute	Graph	Session-adaptive propagation	Multi-task learning
GRU4Rec	✓	✗	✗	✗	✗
GRU4Rec$_m$	✗	✓	✗	✗	✗
GRU4Rec$_m$*	✓	✓	✗	✗	✗
M-GRU4Rec$_m$*	✓	✓	✗	✗	✓
SASRec	✓	✗	✗	✗	✗
SASRec$_m$	✗	✓	✗	✗	✗
SASRec$_m$*	✓	✓	✗	✗	✗
M-SASRec$_m$*	✓	✓	✗	✗	✓
KGHT	✓	✗	✓	✗	✗
KGHT$_m$	✗	✓	✓	✗	✗
KGHT$_m$*	✓	✓	✓	✗	✗
KGHT$_q$	✓	✗	✓	✓	✗
KGHT$_{qm*}$	✓	✓	✓	✓	✗
M-KGHT$_{qm*}$	✓	✓	✓	✓	✓

Table 4: Ablation study on different types of edges.

Backbone	Diginetica		Yoochoose 1/64		THD		Decrease (↓)
	N@10	M@10	N@10	M@10	N@10	M@10	
KGHT$_{qm*}$	18.94	14.22	35.80	28.80	18.83	15.67	–
– w/o diff	18.34	13.80	35.15	28.22	18.68	15.50	2.02%
– w/o v	15.21	11.46	34.36	27.68	18.66	15.53	9.78%
– w/o a	18.70	13.97	35.04	28.09	18.76	15.64	1.39%
– w/o va	/	/	/	/	18.79	15.63	0.23%

* **w/o diff**: uniformly aggregate different types of neighbors with no differentiation.
* **w/o v/a/va**: with no co-view/co-ATC/co-view-ATC edges.

- Comparing among different backbones, graph-based models achieve higher performance than non-graph-based ones, which aligns with the notion that incorporating global information across different sessions is conducive to the recommendation [12, 23, 44]. More specifically, KGHT$_q$ gains 4.25% improvement over KGHT because the designed session-adaptive propagation only aggregates the most relevant neighbor information to the session context and avoids introducing unrelated neighbors.

4.3 Influence of Different Types of Edges

We further analyze the influence of co-view, co-ATC and co-view-ATC edges in the constructed knowledge graph. Specifically, we use **KGHT**$_{qm*}$ as the baseline and respectively remove three types of edges, e.g., **KGHT**$_{qm*}$ w/o v removes the co-view edges and **KGHT**$_{qm*}$ w/o diff treat different types of edges uniformly by employing the same graph attention layer[3]. The performance of removing each specific type of edges is reported in Table 4 and we draw three observations:

- On Diginetica and Yoochoose 1/64, removing co-view edges decreases the performance most because of the following two reasons. First, sessions in these two datasets track user click activities rather than purchase activities and consist of more substitution items rather than complementary ones [21, 36]. Therefore, removing co-view edges that essentially capture the substitution relationship between any two items hurts the performance more than removing other types of edges. Secondly, co-view edges are constructed from both training and validation sessions, message-passing along which captures more recent transitional patterns encoded in the validation sequences. The performance decrease here indicates the distribution shift in transitional patterns from training sessions to validation sessions. One promising direction is to treat sessions at different time stamps differently, as recent transition patterns may be more indicative of the future sessions than the past ones [10].
- Conversely, on THD dataset, since the session is composed of the user's sequentially add-to-cart activity, removing co-view edges causes minor performance degradation compared with the one on the other two datasets. Moreover, removing co-view-ATC edges causes little-to-no performance change, which indicates that capturing the view-to-ATC transition patterns may not help predict users' next clicked items.

Table 5: Performance improvement by multi-task learning.

Backbone	Diginetica		THD	
	N@10	M@10	N@10	M@10
	(All, Sparse)	(All, Sparse)	(All, Sparse)	(All, Sparse)
GRU4Rec$_{m*}$	(13.01, 11.47)	(9.73, 8.37)	(18.13, 11.33)	(15.23, 9.34)
M-GRU4Rec$_{m*}$	(13.05, 11.57)	(9.74, 8.48)	(18.61, 11.76)	(15.66, 9.71)
SASRec$_{m*}$	(15.31, 13.68)	(11.60, 10.12)	(18.54, 11.72)	(15.53, 9.64)
M-SASRec$_{m*}$	(15.66, 13.84)	(11.83, 10.18)	(18.80, 12.08)	(15.78, 9.97)
KGHT$_{qm*}$	(18.93, 17.16)	(14.22, 12.65)	(18.83, 12.17)	(15.67, 9.91)
M-KGHT$_{qm*}$	(18.95, 17.11)	(14.23, 12.53)	(18.98, 12.28)	(15.81, 10.02)

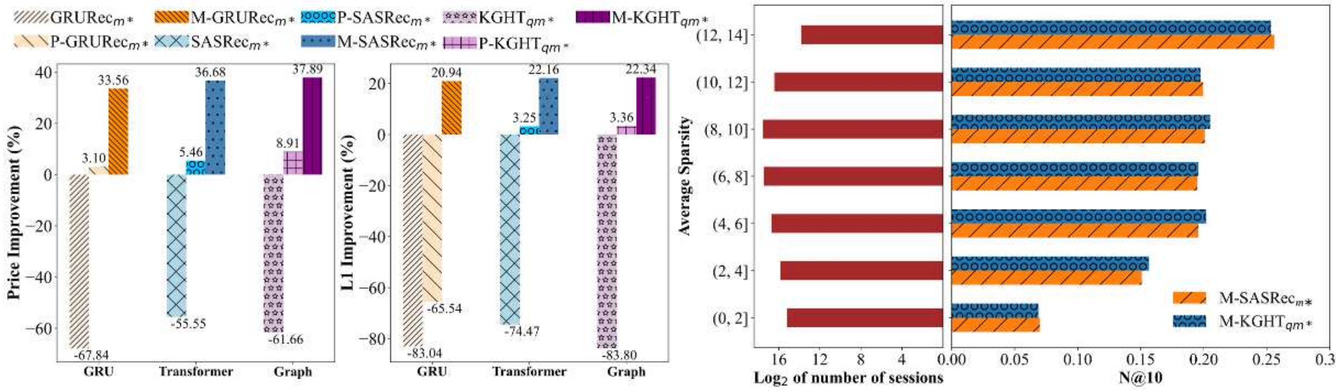
4.4 Multi-task Learning

In this section, we explore the impact of squeezing item meta-label information such as taxonomy and price into the item embeddings on the performance of next item prediction. Concretely, we feed the session embedding $\mathbf{S}_m$ obtained from Eq. (5) into task-specific prediction heads to predict the next item meta-labels. The loss of predicting next-item meta-labels is combined with the loss of next-item prediction to jointly train the whole model and hence can be essentially deemed as self-supervised learning [14, 30]. We first present the performance of next-item prediction before/after incorporating multi-task learning in Table 5. Specifically. **M-X** denotes the basic model **X** augmented by the multi-task learning. The item meta-labels used in Diginetica and THD are the price and L1 categories, respectively.

As shown in Table 5, in most cases, jointly training models by predicting the next-item meta-labels improves next-item prediction. This indicates that squeezing knowledge of the next-item meta-label could sometimes help next-item prediction. An exceptional case is applying **M-KGHT**$_{qm*}$ to Diginetica; the performance does not always increase because aggregating the non-informative meta-attributes of items by message-passing in Diginetica causes little-to-no benefit in next-item prediction. The performance improvement on THD achieved by multi-task learning is stronger than the one on Diginetica because meta-labels of items at THD are more carefully annotated and hence can uniquely characterize the functionality of corresponding items.

Predicting next-item meta-labels by itself has many real-world applications, such as previewing the users' purchasing budget if predicting the next-item price [16] and summarizing the users' purchasing intention if predicting the next-item category [1]. Therefore, we also compare the performance of next-item meta-label prediction with some other baselines, including **Top-N**, **X**$_{m*}$, **P-X**$_{m*}$, and **M-X**$_{m*}$. Their detailed definitions are included in the caption of Figure 3(a)-(b). We report the performance improvement of each model over the baseline **Top-N**. As shown in Figure 3(a)-(b), training prediction heads by multi-task learning significantly enhances the next-item meta-label prediction. The performance improvement of N@5 for predicting price on Diginetica of GRU4Rec increases from -67.84% to +33.56%, and N@5 for L1 category prediction on THD of SASRec increases from -74.47% to +22.16%. Compared among different backbones, graph-based models are better than transformer-based ones and further than GRU-based ones, showing that capturing global transitional patterns is also conducive to predicting the next item meta-label.

[3] Instead of averaging aggregated embeddings across all $\mathcal{T}^2$ types of edges in Eq (3), we use a uniformed query, value and key matrices.



(a) Next Item Price on Diginetica **(b) Next Item L1 category on THD** **(c) Performance of sessions of different average sparsity**

Figure 3: (a)-(b): Comparing the performance improvement(%) over the baseline Top-N on predicting next-item meta-labels. Top-N: the simple heuristic recommending the most frequent meta-labels among previous items within one session; (2) X$_{m*}$ uses an untrained prediction head to predict the task-specific label of the next item; (3) P-X$_{m*}$ first predicts top-N next items and use their task-specific labels as recommendations; (4) M-X$_{m*}$ uses a trained prediction head that is trained by multi-task learning to predict the task-specific label of the next item. (c) N@10 on sessions of different sparsity. M-KGHT$_{qm*}$ is better than M-SASRec$_{m*}$ on sessions of the sparsity at the middle level.

4.5 Analysis on Sessions of Different Sparsity

Prior research has identified degree-related biases in various contexts, including node classification [29], link prediction [18], knowledge graph completion [27], and conventional non-session-based recommender systems [33, 35]. Analogically, we assess the relationship between the next-item prediction performance of the session and its sparsity, defined as the average degree of items in that session. In essence, higher average sparsity in a session indicates lower average involvement of items from that session in other sessions and a reduced number of global transitional patterns that the global knowledge graph can capture. We group sessions based on their average sparsity and report the average N@10 of M-SASRec$_m$* and M-KGHT$_{qm*}$ across all sessions for each group in Figure 3(c).

Clearly, M-KGHT$_{qm*}$ performs better than M-SASRec$_{m*}$ when average sparsity ranges between 2-10 while worse when average sparsity is either below 2 or above 10. On one hand, items in sessions with extremely low sparsity participate in fewer sessions and thus, the captured global transitional patterns across different sessions are insufficient. Therefore, relying more on neighbors as M-KGHT$_{qm*}$ brings no advantages compared with relying more on meta-attributes as M-SASRec$_{m*}$. On the other hand, items in sessions with extremely higher sparsity generally have higher degrees and are usually co-view/co-ATC with many different types of items. Aggregating information from various types of neighboring items serving different functionalities may corrupt the intention of the current session. Because the number of sessions of sparsity between [2, 10] is higher than the number of other sessions, the overall performance of M-KGHT$_{qm*}$ is still higher than M-SASRec$_{m*}$. Further work can study this phenomenon from the fairness perspective and diversity perspective [5, 19, 45] and focus on balancing the performance of sessions of various sparsity by designing adaptive embedding sizes for items in different sessions [19, 45] based on their involvement in different sessions. In addition, we can define session diversity as the average involvement of items and adapt existing diversity methods into alleviate session unfairness.

Table 6: Performance comparison among different baselines.

Baseline	Diginetica		Yoochoose 1/64		THD	
	N@10	M@10	N@10	M@10	N@10	M@10
NARM	16.06	12.12	34.45	28.10	17.44	14.62
SR-GNN	17.10	12.82	35.44	28.66	17.08	14.45
M2TRec	15.66	11.83	33.68	27.16	18.80	15.78
M-KGHT$_{qm*}$	**18.95**	**14.23**	**35.80**	**28.80**	**18.98**	**15.81**

4.6 Performance Comparison with baselines

From the analysis in previous sections, we select the best model configuration: M-KGHT$_{qm*}$ and compare it with state-of-the-art baselines **NARM** [17], **SR-GNN** [39] and **M2TRec** [25]. For the implementation of M2TRec and M-KGHT$_{qm*}$ on Yoochoose 1/64 with no item meta-attributes, we directly use item ID as the input. We modify the implementation of NARM and SR-GNN to include the validation performance and hence align with our experimental setting. In Table 6, M2TRec performs worse than SR-GNN and NARM on Diginetica and Yoochoose 1/64 while better on THD. This is because M2TRec designs an item meta-embedding layer to integer item meta-features and the more informative meta-features on THD than the ones on Diginetica/Yoochoose maximize the benefit of using M2TRec. However, our proposed M-KGHT$_{qm*}$ achieves the best performance across all three datasets for both NDCG@10 and MRR@10. This exhibits the general ability of the proposed framework to realize superior performance over existing methods across datasets with varying real-world dynamics, i.e., having varying amounts of meta-attributes, user-item interaction types, and session lengths. Note that because THD is an industrial-scale dataset (THD has around 3 million training sessions that are 5/10 times larger than Diginetica/Yoochoose), the performance gain on THD is slightly weaker than the gains on the other two datasets.

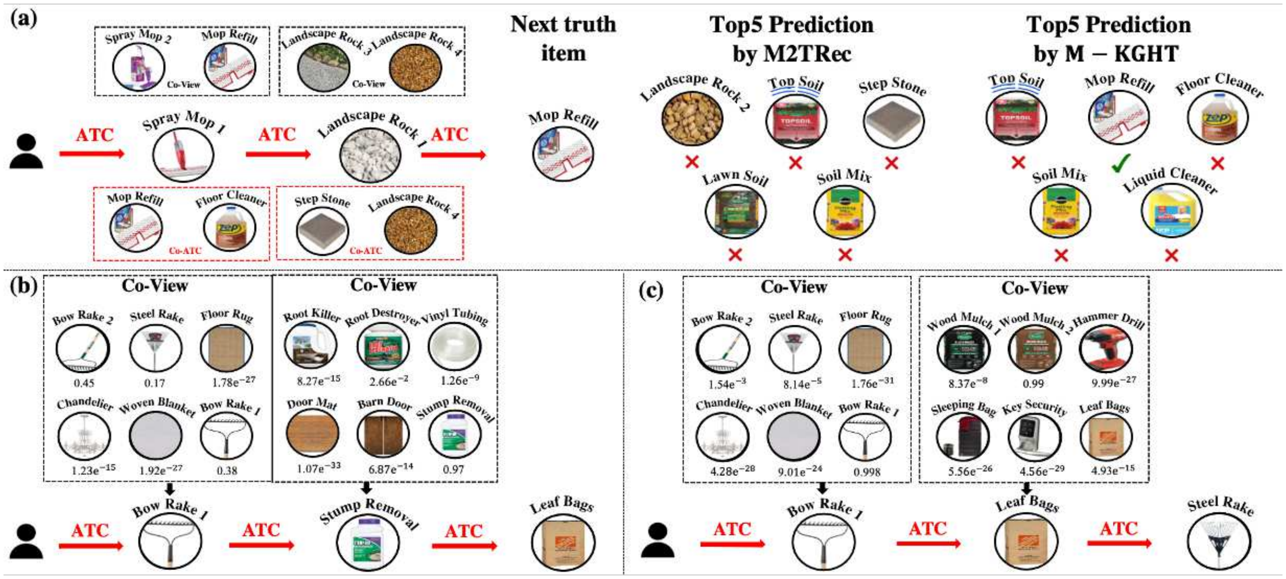


Figure 4: Case study of our proposed M-KGHT. (a) Comparing the Top5 recommendation by M2TRec and M-KGHT. By leveraging neighborhood information of Spray Mop1, the correct item Mop Refill appears in the recommendation list. (b)-(c) visualizes the learned attention of one attention head over co-view neighbors. Since users in both of these two sessions intend to clean the gardens, the session-adaptive graph propagation successfully learns the higher intention for garden-related items.

4.7 Industrial-level Case Study

We further deploy our designed system in the industrial-level setting by training it with 6 million sessions spanning the last two years and evaluating over 0.1 million sessions in the following month. We achieved 2% performance improvement over the previously deployed model at The Home Depot. Next, we conduct some case studies to visualize the effect of our proposed system.

4.7.1 Visualizing the recommendation of M2TRec and M-KGHT$_{qm*}$. To interpret the advantage of the proposed **M-KGHT**$_{qm*}$ over the second best model **M2TRec** on THD dataset, we select the sessions where the top-5 recommendation list given by **M-KGHT**$_{qm*}$ hit the next truth item while the one given by **M2TRec** does not. We visualize one example in Figure 4(a) where the customer sequentially add-to-cart the spray mop and landscape rock. Items on the top-5 recommendation list given by **M2TRec** are uniformly aligned with the landscape rock while recommended items by **M-KGHT**$_{qm*}$ align with both the spray mop and landscape. Furthermore, because **M-KGHT**$_{qm*}$ aggregates neighbor information of mop refill to spray mop, the recommendation hits the true item, which demonstrates the benefits of leveraging neighborhood information.

4.7.2 Visualizing the attention of session-adaptive propagation. We further visualize the graph attention learned by our session-adaptive propagation in Figure 4(b)-(c). Clearly, users generating these two sequences (b)-(c) intend to clean their own gardens and successfully, the model learns to aggregate less information from irrelevant neighbors, e.g., $1.78e^{-27}$ from floor rug to bow rake 1 in (b) and $9.99e^{-27}$ from hammer drill to leaf bags in (c). Interestingly, we find model sometimes pays attention to only one relevant neighbor. For example, even though both bow rake 1 and steel rake are aligned with the intention of our second customer in (c), the model pays its whole attention to bow rake 1. This aligns with diversity observation in [31] and motivates the design of multi-head attention to focus on different important neighbors. More importantly, we can find the neighborhood attention of the same item bow rake 1 varies from 0.38 in (b) to 0.998 in (c), demonstrating that even though the neighborhoods are the same, the attention assigned to them changes if the sequence content changes. This verify the effectiveness of our proposed session-adaptive propagation.

5 CONCLUSION

Characterizing user intention by modeling global transitional patterns of user-item interactions is essential in the session-based recommendation. Traditional transformer-based models fail to capture global transitional patterns among items. More recent GNN-augmented transformers ignore the session context and only consider one type of customer-item interaction. Given these problems, we propose a knowledge graph-based session recommendation framework with session-adaptive propagation. We construct the graph by extracting three different types of user-item interactions and design a session-adaptive propagation for aggregating neighbors' information based on their consistency with the session intention. A comprehensive ablation analysis shows the proposed strategies provide a 10%-20% improvement. Moreover, our case study on recommendation interpretation demonstrates that learned neighborhood attention is highly determined by the consistency of the neighbor with the session intention.

ACKNOWLEDGEMENTS

This research is supported by The Home Depot and the National Science Foundation (NSF) under grant number IIS2239881.

REFERENCES

[1] Renqin Cai, Jibang Wu, Aidan San, Chong Wang, and Hongning Wang. 2021. Category-aware collaborative sequential recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 388–397.

[2] Hanning Chen, Ali Zakeri, Fei Wen, Hamza Errahmouni Barkam, and Mohsen Imani. 2023. HyperGRAF: Hyperdimensional Graph-Based Reasoning Acceleration on FPGA. In *2023 33rd International Conference on Field-Programmable Logic and Applications (FPL)*. 34–41. https://doi.org/10.1109/FPL60245.2023.00013

[3] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. 2020. Simple and deep graph convolutional networks. In *International conference on machine learning*. PMLR, 1725–1735.

[4] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge. 2021. Transformers4Rec: Bridging the Gap between NLP and Sequential/Session-Based Recommendation. In *Fifteenth ACM Conference on Recommender Systems*. 143–153.

[5] Yushun Dong, Jing Ma, Song Wang, Chen Chen, and Jundong Li. 2023. Fairness in graph mining: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[6] Priyanka Gupta, Diksha Garg, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2019. NISER: Normalized item and session representations to handle popularity bias. *arXiv preprint arXiv:1909.04276* (2019).

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[8] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 843–852.

[9] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. *ICLR* (2016).

[10] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems* 33 (2020), 22118–22133.

[11] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*. 2704–2710.

[12] Chao Huang, Jiahui Chen, Lianghao Xia, Yong Xu, Peng Dai, Yanqing Chen, Liefeng Bo, Jiashu Zhao, and Jimmy Xiangji Huang. 2021. Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4123–4130.

[13] Di Jin, Luzhi Wang, Yizhen Zheng, Guojie Song, Fei Jiang, Xiang Li, Wei Lin, and Shirui Pan. 2023. Dual Intent Enhanced Graph Neural Network for Session-based New Item Recommendation. In *Proceedings of the ACM Web Conference 2023*. 684–693.

[14] Wei Jin, Tyler Derr, Haochen Liu, Yiqi Wang, Suhang Wang, Zitao Liu, and Jiliang Tang. 2020. Self-supervised learning on graphs: Deep insights and new direction. *arXiv preprint arXiv:2006.10141* (2020).

[15] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[16] Oleg Lashinin and Marina Ananyeva. 2022. Next-Basket Recommendation Constrained by Total Cost. (2022).

[17] 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.

[18] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented fairness in recommendation. In *Proceedings of the Web Conference 2021*. 624–632.

[19] Haochen Liu, Xiangyu Zhao, Chong Wang, Xiaobing Liu, and Jiliang Tang. 2020. Automated embedding size search in deep recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2307–2316.

[20] Yunchao Lance Liu, Yu Wang, Oanh Vu, Rocco Moretti, Bobby Bodenheimer, Jens Meiler, and Tyler Derr. 2023. Interpretable chirality-aware graph neural network for quantitative structure activity relationship modeling in drug discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 14356–14364.

[21] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*.

[22] Zhiqiang Pan, Fei Cai, Wanyu Chen, and Honghui Chen. 2021. Graph Co-attentive session-based recommendation. *ACM Transactions on Information Systems (TOIS)* 40, 4 (2021), 1–31.

[23] Ruihong Qiu, Zi Huang, Jingjing Li, and Hongzhi Yin. 2020. Exploiting cross-session information for session-based recommendation with graph neural networks. *ACM Transactions on Information Systems (TOIS)* 38, 3 (2020), 1–23.

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

[25] Walid Shalaby, Sejoon Oh, Amir Afsharinejad, Srijan Kumar, and Xiquan Cui. 2022. M2TRec: Metadata-aware Multi-task Transformer for Large-scale and Cold-start free Session-based Recommendations. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 573–578.

[26] Guy Shani, David Heckerman, Ronen I Brafman, and Craig Boutilier. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, 9 (2005).

[27] Harry Shomer, Wei Jin, Wentao Wang, and Jiliang Tang. 2023. Toward Degree Bias in Embedding-Based Knowledge Graph Completion. *arXiv preprint arXiv:2302.05044* (2023).

[28] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[29] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Yiqi Wang, Jiliang Tang, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. 2020. Investigating and mitigating degree-related biases in graph convoltuional networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1435–1444.

[30] Yijun Tian, Kaiwen Dong, Chunhui Zhang, Chuxu Zhang, and Nitesh V Chawla. 2023. Heterogeneous Graph Masked Autoencoders. In *AAAI*.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[32] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z Sheng, Mehmet A Orgun, and Defu Lian. 2021. A survey on session-based recommender systems. *ACM Computing Surveys (CSUR)* 54, 7 (2021), 1–38.

[33] Yu Wang and Tyler Derr. 2022. Degree-Related Bias in Link Prediction. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 757–758.

[34] Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2023. Knowledge graph prompting for multi-document question answering. *arXiv preprint arXiv:2308.11730* (2023).

[35] Yu Wang, Yuying Zhao, Yi Zhang, and Tyler Derr. 2022. Collaboration-Aware Graph Convolutional Networks for Recommendation Systems. *arXiv preprint arXiv:2207.06221* (2022).

[36] Zihan Wang, Ziheng Jiang, Zhaochun Ren, Jiliang Tang, and Dawei Yin. 2018. A path-constrained framework for discriminating substitutable and complementary products in e-commerce. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 619–627.

[37] 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.

[38] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential recommendation via personalized transformer. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 328–337.

[39] 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.

[40] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Xiyue Zhang, Hongsheng Yang, Jian Pei, and Liefeng Bo. 2021. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4486–4493.

[41] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Victor S Sheng S. Sheng, Zhiming Cui, Xiaofang Zhou, and Hui Xiong. 2019. Recurrent convolutional neural network for sequential recommendation. In *The world wide web conference*. 3398–3404.

[42] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD*. 974–983.

[43] Rongzhi Zhang, Yulong Gu, Xiaoyu Shen, and Hui Su. 2021. Knowledge-enhanced Session-based Recommendation with Temporal Transformer. *arXiv preprint arXiv:2112.08745* (2021).

[44] Wei Zhang, Zeyuan Chen, Hongyuan Zha, and Jianyong Wang. 2021. Learning from substitutable and complementary relations for graph-based sequential product recommendation. *ACM Transactions on Information Systems (TOIS)* 40, 2 (2021), 1–28.

[45] Xiangyu Zhao, Haochen Liu, Wenqi Fan, Hui Liu, Jiliang Tang, Chong Wang, Ming Chen, Xudong Zheng, Xiaobing Liu, and Xiwang Yang. 2021. Autoemb: Automated embedding dimensionality search in streaming recommendations. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 896–905.

[46] Yuying Zhao, Yu Wang, Yi Zhang, Pamela Wisniewski, Charu Aggarwal, and Tyler Derr. 2018. Fair Online Dating Recommendations for Sexually Fluid Users via Leveraging Opposite Gender Interaction Ratio. (2018).

A APPENDIX

A.1 Comprehensive Related Work

A.1.1 Session-based Recommendation System. Given that not all historical user-item interactions are beneficial to predict users' current preferences [32], SBR Systems have emerged with increasing attention in recent years [8, 9]. Earlier works leverage Markov Chains to infer the conditional probability of an item based on the previously interacted items [24, 26]. [26] infers the transition function based on user data and additionally employs skipping/clustering to enhance the recommendation. [24] factorizes the probability transition matrix of each user to model sequential behavior between every two adjacent clicks. More recent works have resorted to deep learning for session recommendation. Recurrent neural networks (RNNs) [8, 41] such as GRU4Rec [9] have been developed to model the interactions among adjacent items in a session. To further enhance the communications between non-adjacent items, transformers (TFs) [4, 28] such as SASRec [15] adopt a self-attention to allow information exchange between non-adjacent items in a session. To capture even more complex transitional patterns, graph-based methods such as SR-GNN [39] extract the session graph for each session and use a gated graph neural network to learn session embeddings. Different from previous works that only capture transitional patterns within the session, we construct a global graph with different types of edges to capture even more broad information, the related works of which are reviewed next.

A.1.2 Graph-based Recommendation System. Graph, as a general data structure representing relations of entities, has been widely adopted to assist many real-world applications [2, 20, 34, 46] and one of the most representative examples is session recommendation. Previous works explore global transitional patterns across different sessions by querying the global item graph [12, 23, 44]. [12] designs a global context-enhanced inter-session relation encoder to capture the inter-session item-wise dependencies. [44] constructs the dual session graph to model the pair-wise transition relationship between items based on the global connections. [23] constructs the global graph by merging all individual session graphs. The very recent work KSTT [43] resorts to an item-category knowledge graph for session recommendation. However, the proposed models in all the above works learn item embeddings from the global graph without any session-tailored modification. Only GCE-GNN [37] and GCARM [22] consider session adaptation in aggregating neighbors' information. However, GCE-GNN quantifies the importance of neighbors based on their similarity to the whole session without differentiating central items. GCARM treats all transitions similarly without distinguishing different types of interactions. To handle these two issues, we design a session-adaptive propagation to query neighbors based on session contexts and interaction types, the effectiveness of which is verified in Section 4.7. Note that in this work, although three types of item-item co-interaction edges are considered, the constructed knowledge graph has only one type of node, the item. We leave the inclusion of different node types as one future work, such as adding user nodes, which could provide a way to personalize the session recommendations.