Collaboration and Transition: Distilling Item Transitions into Multi-Query Self-Attention for Sequential Recommendation

Tianyu Zhu
Beihang University
Beijing, China
ztybuaa@126.com

Yansong Shi
Tsinghua University
Beijing, China
shiys.18@sem.tsinghua.edu.cn

Yuan Zhang
Kuaishou Technology
Beijing, China
yuanz.pku@gmail.com

Yihong Wu
University of Montreal
Montreal, Canada
yihong.wu@umontreal.ca

Fengran Mo
University of Montreal
Montreal, Canada
fengran.mo@umontreal.ca

Jian-Yun Nie
University of Montreal
Montreal, Canada
nie@iro.umontreal.ca

ABSTRACT

Modern recommender systems employ various sequential modules such as self-attention to learn dynamic user interests. However, these methods are less effective in capturing collaborative and transitional signals within user interaction sequences. First, the self-attention architecture uses the embedding of a single item as the attention query, which is inherently challenging to capture collaborative signals. Second, these methods typically follow an auto-regressive framework, which is unable to learn global item transition patterns. To overcome these limitations, we propose a new method called Multi-Query Self-Attention with Transition-Aware Embedding Distillation (MQSA-TED). First, we propose an L-query self-attention module that employs flexible window sizes for attention queries to capture collaborative signals. In addition, we introduce a multi-query self-attention method that balances the bias-variance trade-off in modeling user preferences by combining long and short-query self-attentions. Second, we develop a transition-aware embedding distillation module that distills global item-to-item transition patterns into item embeddings, which enables the model to memorize and leverage transitional signals and serves as a calibrator for collaborative signals. Experimental results on four real-world datasets show the superiority of our proposed method over state-of-the-art sequential recommendation methods.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

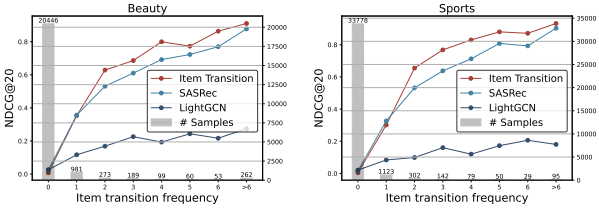
KEYWORDS

sequential recommendation, self-attention, knowledge distillation

ACM Reference Format:
Tianyu Zhu, Yansong Shi, Yuan Zhang, Yihong Wu, Fengran Mo, and Jian-Yun Nie. 2018. Collaboration and Transition: Distilling Item Transitions into Multi-Query Self-Attention for Sequential Recommendation. In *WSDM '24: The 17th ACM International Conference on Web Search and Data Mining, March 4–8, 2024, Merida, Mexico.* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

WSDM '24, March 4–8, 2024, Merida, Mexico

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX



Dataset	Trans. Freq.	Item Trans.	SASRec	LightGCN	# Samples
Beauty	0	0.0081	0.0243	0.0261	20,446
	>0	0.5520	0.5180	0.1672	1,917
	All	0.0547	0.0666	0.0382	22,363
Sports	0	0.0045	0.0161	0.0209	33,778
	>0	0.4772	0.4523	0.1034	1,820
	All	0.0287	0.0384	0.0251	35,598

Figure 1: Performance of three methods w.r.t. item transition frequency on two datasets. Item Transition performs better on test samples with frequent transitions, while LightGCN performs better on test samples lacking transition instances. SASRec achieves the best performance on average.

1 INTRODUCTION

In recent years, there has been an increasing focus on modeling dynamic user preferences in modern recommender systems [2, 30], which is achieved by incorporating various sequential modules such as RNN [6], CNN [21], and Transformer [8, 20]. These sequential recommenders aim to integrate contextual factors derived from recent user interactions into personalized user interests. Contextual factors exhibit typical item-to-item transition patterns. The main challenge in sequential recommendation lies in effectively learning both personalized user interests and general item transition patterns while maintaining an appropriate balance between the two factors. For instance, a user interested in sportswear may also seek a shirt after purchasing a suit. If we only rely on collaborative signals to generate recommendations, we may overlook the user's temporary need for items to complement their suit. On the other hand, if we only consider transitional signals to make recommendations, we

may neglect the user's primary interest in sportswear. Therefore, it is crucial to leverage both signals and find a balance between them. Here we define the collaborative and transitional signals in the context of sequential recommendation tasks:

Definition 1.1 (Collaborative Signals). *In the context of sequential recommendation, collaborative signals refer to the similarities between sequences of users' interacted items.*

Definition 1.2 (Transitional Signals). *In the context of sequential recommendation, transitional signals refer to the transition frequency between pairs of users' interacted items.*

Specifically, collaborative signals can be used by following a sequence-to-item methodology, leveraging the collaborative behavior of users to identify patterns in their interactions and recommend relevant items. On the other hand, transitional signals exploit item-to-item relationships in user interaction sequences, enabling the identification of trigger items that will lead to related purchases.

Although recent sequential recommendation methods such as SASRec [8] have demonstrated remarkable performance, they still encounter inherent limitations in effectively capturing both signals within user interaction sequences. To highlight these limitations, we conducted experiments comparing the performance of SASRec with two baseline methods: Item Transition and LightGCN [5]. Item Transition is a memory-based, non-personalized method that makes recommendations based on the global transition frequency from the current item to candidate items, serving as a benchmark based on transitional signals (see Section 3.2 for details). LightGCN is a state-of-the-art non-sequential recommendation method that learns user and item embeddings through linear propagation on the user-item interaction graph, serving as a benchmark based on collaborative signals. We conducted experiments on two Amazon datasets, Beauty and Sports [32], and grouped the test samples based on the transition frequency observed in the training data. Results shown in Figure 1 reveal two limitations of SASRec in leveraging both signals:

First, SASRec's ability to leverage collaborative signals is outperformed by LightGCN. For test samples where the item transition frequency is zero, LightGCN consistently outperforms SASRec on both datasets. This observation shows the limited ability of SASRec to generalize to test samples lacking observed item transitions. Notably, SASRec uses the embedding of the most recent item as the query in its self-attention module, which can be regarded as an attention-enhanced first-order Markov chain model that is inherently difficult to leverage collaborative signals.

Second, SASRec's ability to leverage transitional signals is outperformed by Item Transition. For test samples where the item transition frequency exceeds one, *i.e.*, the transition occurs multiple times in the training data, Item Transition significantly outperforms SASRec on both datasets. This observation highlights the limited effectiveness of SASRec in leveraging transitional signals.

Inspired by these observations, we propose a new method called *Multi-Query Self-Attention with Transition-Aware Embedding Distillation* (MQSA-TED) for sequential recommendation tasks, which consists of two main components to capture collaborative and transitional signals, respectively. First, we propose an L-query self-attention module that uses flexible window sizes for attention queries to capture collaborative signals. By enlarging the window size L, the model can leverage similarities between longer-range sequences of users' interacted items to generate recommendations. However, using a large L will result in bias as user interests may shift over time. To strike a balance between bias and variance in modeling users' dynamic interests, we introduce a multi-query self-attention method by combining long and short-query self-attentions. Second, we develop a transition-aware embedding distillation module that distills global item-to-item transition patterns into item embeddings, which serves as a calibration module that enables the model to effectively memorize and leverage transitional signals when making recommendations. Notably, our proposed method achieves inherent disentanglement of user collaboration modeling and item transition modeling by employing dual supervision: the original item embedding captures item-to-item transitional signals, while the item embedding after self-attention modules captures sequence-to-item collaborative signals. Our contributions in this paper are summarized as follows:

- We propose an L-query self-attention module that uses flexible window sizes for attention queries to capture collaborative signals. We also design a multi-query self-attention method that combines long and short-query self-attentions to balance the bias-variance trade-off in modeling users' dynamic interests.
- We develop a transition-aware embedding distillation module that distills the global item-to-item transition patterns into item embeddings to capture transitional signals, which serves as a calibration module for collaborative signals.
- We conduct extensive experiments on four real-world datasets to show the effectiveness of our proposed method. The results also highlight the different effects of the proposed two modules in improving recommendation performances.

2 PRELIMINARIES

2.1 Problem Formulation

The sequential recommendation task aims to predict the next item that a user will interact with based on their historical interactions. Let $\mathcal{U} = \{u_1, u_2, \cdots, u_{|\mathcal{U}|}\}$ be the set of users, $\mathcal{I} = \{i_1, i_2, \cdots, i_{|\mathcal{I}|}\}$ be the set of items, and $S^{(u)} = [i_1^{(u)}, i_2^{(u)}, \cdots, i_{n_u}^{(u)}]$ be the interaction sequence of user u, where n_u denotes the length of the sequence. The problem is formulated as calculating the probability of the next item being interacted with, given the user's historical interactions:

$$p\left(i_{n_u+1}^{(u)} | S^{(u)}\right). \tag{1}$$

Then the top-N items will be recommended to user u based on these probabilities in descending order.

2.2 SASRec

Here we briefly introduce the *SASRec* [8] model, which is a state-of-the-art sequential recommender based on the self-attention module in *Transformer* [24] and will be used as the base model in our approach. Given a user interaction sequence of the most recent n items $[i_1, i_2, \cdots, i_n]$ (here we omit the superscript (u) for simplicity), an embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{I}| \times d}$ is used to convert the sequence into an embedding sequence $[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n]$, where d is the embedding size. Then a learnable positional embedding $\mathbf{P} \in \mathbb{R}^{n \times d}$ is added

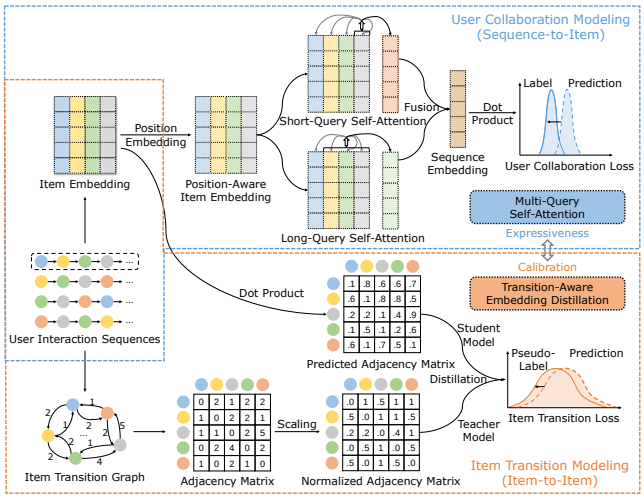


Figure 2: Illustration of the proposed MQSA-TED method. It consists of two main components: 1) Multi-Query Self-Attention for user collaboration modeling, and 2) Transition-Aware Embedding Distillation for item transition modeling.

to encode the position information, resulting in $[\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \cdots, \hat{\mathbf{e}}_n]$, where $\hat{\mathbf{e}}_t = \mathbf{e}_t + \mathbf{p}_t$. Next, the transformer [24] module is used as the encoder:

$$[\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \cdots, \tilde{\mathbf{e}}_n] = \text{Transformer}([\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \cdots, \hat{\mathbf{e}}_n]), \tag{2}$$

which adopts multiple blocks of self-attention and feed-forward networks. The self-attention layer is used to capture the long-term sequential dependency as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}, \tag{3}$$

$$\mathbf{Q} = \hat{\mathbf{E}}\mathbf{W}^Q, \ \mathbf{K} = \hat{\mathbf{E}}\mathbf{W}^K, \ \mathbf{V} = \hat{\mathbf{E}}\mathbf{W}^V, \tag{4}$$

where $\mathbf{Q}$ represents the queries, $\mathbf{K}$ the keys, $\mathbf{V}$ the values, and $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V \in \mathbb{R}^{d\times d}$ are the projection matrices for queries, keys, and values, respectively. Finally, the model predicts ranking scores by taking the dot product between the sequence embedding and the candidate item embeddings as $\hat{\mathbf{r}}_t = \tilde{\mathbf{e}}_t\mathbf{E}^T$. The cumulative cross-entropy loss is used for model training as follows:[1]

$$\mathcal{L}_{rec} = -\sum_{t=1}^{n} \mathbf{r}_t \log \text{softmax}(\hat{\mathbf{r}}_t), \tag{5}$$

where $\mathbf{r}_t \in \mathbb{R}^{1\times|I|}$ is a one-hot vector converted from the index of the ground truth item at timestamp t.

3 METHODOLOGY

In this section, we present the proposed method, which consists of two main components as illustrated in Figure 2: 1) Multi-Query Self-Attention for user collaboration modeling, and 2) Transition-Aware Embedding Distillation for item transition modeling.

[1]This loss function has been shown more effective than the negative sampling-based binary cross-entropy loss [13] and we use it for all models in our experiments.

3.1 Multi-Query Self-Attention for User Collaboration Modeling

We adopt SASRec as our base model owing to its strong ability to capture long-term sequential dependency and its state-of-the-art performance in sequential recommendation tasks [8]. SASRec uses the self-attention module in Transformer [24], whose main components are the *queries*, *keys*, and *values*, as shown in Equation (4). Specifically, the attention query at timestamp t in SASRec can be expressed as follows:

$$\mathbf{q}_t = \hat{\mathbf{e}}_t\mathbf{W}^Q, \tag{6}$$

where $\hat{\mathbf{e}}_t$ is the embedding vector of the item at timestamp i after adding the positional embedding, and $\mathbf{W}^Q$ is a learnable projection matrix. Then, the attention weights assigned to historical items $[i_1, i_2, \cdots, i_t]$ at timestamp t are determined by the scaled dot-product between the query embedding and the key embeddings as shown in Equation (3). Therefore, the attention weights are dominated by the single item at timestamp t, leading to a type of short-query self-attention.

However, this type of self-attention is limited in leveraging collaborative signals, especially when the item at timestamp t is inconsistent with the user's primary preference. Specifically, SASRec can be viewed as a self-attention-enhanced first-order Markov chain model and its recommendation results can be significantly affected by a minor change in the order of users' interacted item sequences, such as swapping the position of the last two items. In other words, SASRec may generalize poorly on test samples lacking observed item transitions. However, real-world recommendation scenarios such as restaurant recommendations on Yelp have shown that user interests are relatively stable and less sensitive to the order of several recent choices [34] but SASRec may have difficulty coping with this situation. To address this limitation, we propose an L-query self-attention approach. First, we define the L-query self-attention as follows:

Definition 3.1 (L-query Self-Attention). *An L-query self-attention is a type of self-attention module that uses the embeddings or their transformed representations of the most recent L timestamps' items (tokens) as the attention query.*

Here we use the simple mean-pooling of the embeddings of the last L items at timestamp t as the query embedding:

$$\tilde{\mathbf{q}}_t = \text{mean-pooling}(\hat{\mathbf{e}}_{t-L+1}, \hat{\mathbf{e}}_{t-L+2}, \cdots, \hat{\mathbf{e}}_t)\tilde{\mathbf{W}}^Q, \tag{7}$$

where L is a hyperparameter that controls the range of the attention query. Alternatively, other functions can be used to generate the query embedding, such as a weighted summation with time decay.

It is important to note that the hyperparameter L controls the range of the historical context in self-attention. Using a large value of L means that the model relies on long-range historical items to represent user interests, which contributes to capturing collaborative signals but may accumulate bias as user interests may shift over time. Conversely, using a small value of L means that the model adopts the latest interacted items to represent user interests but can introduce variance due to the small number of used items. To balance the bias-variance trade-off, we propose a *Multi-Query Self-Attention* (MQSA) method that combines the short-query

self-attention (with $L = 1$, similar to SASRec) with the long-query self-attention (with a larger L) using a hyperparameter α:

$$\tilde{\mathbf{e}}_t = \alpha \cdot \tilde{\mathbf{e}}_t^{short} + (1 - \alpha) \cdot \tilde{\mathbf{e}}_t^{long}. \tag{8}$$

Then, the sequence embedding $\tilde{\mathbf{e}}_t$ is used along with the embedding of candidate items to predict their ranking scores through dot product. Notably, we can also allow the model to learn the optimal α. However, simultaneously learning the weights and the embeddings is challenging due to its inherent complexity. We could also incorporate more Ls. We leave these for exploration in future work.

It is worth mentioning that the formulation of MQSA shares similar ideas with some approaches in the literature, such as FPMC [18] and Fossil [3], which explicitly model long-term user interests by employing user or item embeddings, respectively, and combine them with factorized Markov chains for sequential recommendation tasks. Compared to Fossil which uses the whole interacted items, MQSA introduces flexible window sizes of the last L items to control the bias-variance trade-off. Furthermore, MQSA employs self-attention modules to enhance expressiveness, resulting in improved performance compared to the use of pure item embeddings in Fossil.

3.2 Transition-Aware Embedding Distillation for Item Transition Modeling

Sequential recommendation models have demonstrated their effectiveness in enhancing recommendation accuracy by capturing long-term user interests [6, 8, 32]. However, these models may have limitations in leveraging the global item-to-item transitional signals. Specifically, most existing methods follow an auto-regressive framework [8, 32]. For each user, their preference at timestamp t is learned based on their interacted items up to and including t and then used to predict the item at timestamp $t + 1$. Nevertheless, this framework fails to enable the model to learn the global item-to-item transition patterns. In other words, the items not interacted with by a user are treated equally, without the consideration of the potential items that the current item i_t is more likely to trigger.

To address this limitation, we propose a heuristic recommender based on item transitions and then develop a knowledge distillation method to integrate these global item transition patterns into sequential models. Specifically, we construct a global item transition graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ represents item nodes and $\mathcal{E}$ represents transition edges between items. $\mathcal{G}$ is a weighted and directed graph, where the weight of each edge represents the transition frequency between two items within a time span k, based on all user interaction sequences. Note that the time span hyperparameter k is used to allow for long-term item transition patterns and is set to 1 by default. We use the adjacent matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$ of $\mathcal{G}$ as the heuristic recommender, where $a_{i,j}$ is the transition frequency from item i to item j, as shown in Figure 2. It is a memory-based non-personalized method that recommends items based on the transition frequency from the current item to candidate items, as introduced in our preliminary experiments in Section 1.

To distill the item transitions into the sequential model, we propose a *Transition-Aware Embedding Distillation* (TED) method. First, we normalize the transition frequencies using a row normalization approach as $\bar{a}_{i,j} = \frac{a_{i,j}}{\max_j a_{i,j}}$. Then, we use a softmax function with temperature τ to generate pseudo-labels for knowledge distillation:

$$\tilde{\mathbf{a}}_i = \text{softmax}(\bar{\mathbf{a}}_i/\tau), \tag{9}$$

where a higher value of τ generates a softer probability distribution over items [7].

We adopt a simple factorization model as the student model, which predicts the item transition distribution of item i by using the dot product between its embedding vector $\mathbf{e}_i$ and the embedding matrix $\mathbf{E}$ before the self-attention layers, where the dropout [19] strategy is also used for robust learning. We apply the softmax function with temperature τ to obtain the predicted transition probabilities:

$$\hat{\mathbf{a}}_i = \text{softmax}(\mathbf{e}_i\mathbf{E}^T/\tau). \tag{10}$$

We use the cross-entropy loss to distill the item transitions into the sequential model by comparing the predicted and pseudo-label transition probabilities:

$$\mathcal{L}_{kd} = -\sum_{i\in\mathcal{I}} \tilde{\mathbf{a}}_i \log \hat{\mathbf{a}}_i. \tag{11}$$

Therefore, the factorization model can learn from the Item Transition model, enabling the item embeddings to memorize the item transition patterns. The overall loss function for the full model is:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{kd}\mathcal{L}_{kd} + \lambda_\Theta||\Theta||_2^2, \tag{12}$$

where Θ is the parameters, λ_{kd} and λ_Θ are the hyperparameters that control the weights of distillation and l_2 regularization, respectively.

3.3 Discussion

3.3.1 Relationship Between Two Modules. Here we discuss the relationship between the user collaboration and item transition modules, and how they complement each other in capturing user preferences for generating recommendations.

Expressiveness vs. Calibration. The item transition module learns from a memory-based method that generates potential candidate items based on the global transition trends of the current item. However, it may generalize poorly to the items lacking observed transition patterns. On the other hand, the user collaboration module is a neural model that employs self-attentions to capture long-term user preferences and select the most likely next item based on historical items, resulting in a stronger ability to generalize but a limited ability to memorize and leverage item-to-item transition patterns. Therefore, the user collaboration model requires the item transition model to act as a calibrator for its predictions.

Disentangled Learning. The user collaboration and item transition modules are inherently disentangled, as we employ dual supervision where the original item embedding captures item-to-item transitional signals while the item embedding after self-attentions captures sequence- to-item collaborative signals.

Retrieval vs. Re-Ranking. The item transition and user collaboration modules can be regarded as a retrieval model and a re-ranking model, respectively. The retrieval model provides insight into generating potential candidate items, while the re-ranking model provides insight into selecting the most relevant items for users based on their respective interaction histories.

3.3.2 Comparison with Existing Methods. The proposed Transition-Aware Embedding Distillation (TED) module serves as a calibrator based on the item transition graph. Here we compare it with recent graph-based regularization methods:

Graph Regularization (GraReg) [28] is a Euclidean distance-based regularization term on embedding layers using a k-nearest neighbor (k-NN) graph:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{reg} \sum_{(i,j)\in\mathcal{E}} ||\mathbf{e}_i - \mathbf{e}_j||^2, \tag{13}$$

where λ_{reg} is the coefficient hyperparameter for graph regularization, and $\mathcal{E}$ is the edges in the k-NN graph. We can use the transition frequency as the weights of the edges here. Therefore, GraReg uses the k most related items for regularization, leading to learning localized transition patterns. Additionally, GraReg introduces an alignment loss but lacks a uniformity loss, where related items should be close to each other while unrelated ones should be separated [25]. In contrast, TED uses the global item transitions as the teacher model, enabling the item embeddings to memorize and leverage transitional signals.

Graph-based Embedding Smoothing (GES) [34] employs graph convolutions on the global item transition graph for embedding smoothing in sequential recommenders:

$$\mathbf{E}^{(l+1)} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{E}^{(l)}, \tag{14}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix of the item transition graph with self-loops, $\tilde{\mathbf{D}}$ is the degree matrix of $\tilde{\mathbf{A}}$, and l is the number of graph convolutional layers. However, stacking multiple graph convolutional layers may result in over-smoothing problems [10], potentially leading to a decline in model performance. In comparison, TED incorporates a hyperparameter to control the power of item transition distillation, allowing for flexibility in different recommendation scenarios.

3.3.3 Model Complexity. Here we analyze the space and time complexity of the proposed model.

Space Complexity. The learnable parameters in SASRec come from item embeddings, positional embeddings, self-attention layers, feed-forward layers, and layer normalization. The total number of parameters in SASRec is $O(|\mathcal{I}|d + nd + d^2)$ [8]. Our proposed model introduces the long-query self-attention, which adds $O(d^2)$ for projection matrices, feed-forward networks, and layer normalization. The embedding distillation module does not add any extra parameters. Therefore, the space complexity of our proposed model is the same as that of SASRec.

Time Complexity. The computational complexity of the self-attention layer and the feed-forward layer in SASRec is $O(n^2d+nd^2)$. The cumulative cross-entropy loss has a complexity of $O(|\mathcal{I}|nd)$. Thus, the total computational complexity of SASRec is $O(|\mathcal{I}|nd + n^2d+nd^2)$. In our proposed model, the self-attention module has the same complexity as in SASRec. The embedding distillation module has a complexity of $O(|\mathcal{I}|nd)$. Hence, the time complexity of the proposed model is the same as that of SASRec with the cumulative cross-entropy loss.

Table 1: Summary of evaluation datasets. The datasets are from [32].

Dataset	# Users	# Items	# Actions	Density	Avg. Len.
Beauty	22,363	12,101	198,502	0.073%	8.88
Sports	25,598	18,357	296,337	0.063%	8.32
Toys	19,412	11,924	167,597	0.072%	8.63
Yelp	30,431	20,033	316,354	0.052%	10.40

4 EXPERIMENTS

We conduct experiments on four real-world datasets to evaluate the effectiveness of the proposed method.[2] The experiments are designed to answer the following research questions:

- **RQ1.** How does the proposed method compare with state-of-the-art sequential recommendation methods?
- **RQ2.** How do the hyperparameters and various components affect the model performance?
- **RQ3.** How does the proposed TED method compare with graph-based regularization methods?
- **RQ4.** Can the proposed TED method benefit various recommendation models?
- **RQ5.** How do the proposed two modules improve the model performance?

4.1 Experimental Settings

4.1.1 Datasets. We adopt four datasets from [32] for experiments. The Beauty, Sports, and Toys datasets are from the Amazon Review Dataset in [4, 16].[3] The Yelp dataset is from the Yelp Open Dataset.[4] The training data, validation data, and test data are identical to those used in [32], which follows the leave-one-out evaluation protocol that treats the last item as the test data, the second last item as the validation data, and the remaining items as the training data for each user [8]. The dataset statistics are shown in Table 1.

4.1.2 Baselines. We compare the proposed method with various types of state-of-the-art baselines in sequential recommendation:

- **POP**: a non-personalized method that ranks items based on their popularity.
- **LightGCN** [5]: a *GCN*-based method that learns user and item embeddings through linear propagation on the user-item interaction graph.
- **FPMC** [18]: a *Markov chain*-based method that combines matrix factorization and factorized Markov chains.
- **Caser** [21]: a *CNN*-based method that uses horizontal and vertical convolutions to learn sequential patterns.
- **GRU4Rec** [6]: an *RNN*-based method that uses Gated Recurrent Units (GRU) to model dynamic user preferences.
- **SASRec** [8]: a *unidirectional Transformer*-based method that models user interests using the self-attention module in Transformer [24].
- **BERT4Rec** [20]: a *bidirectional Transformer*-based method that models user interests using the self-attention module in BERT [1].

[2]The codes and datasets are available at https://github.com/zhuty16/MQSA-TED
[3]https://cseweb.ucsd.edu/~jmcauley/datasets.html
[4]https://www.yelp.com/dataset

Table 2: Performance comparison of different methods on four datasets. The best results are in boldface and the second best are underlined. Asterisk (*) indicates statistically significant improvements over the best baseline determined by a two-sample t-test ($p < 0.01$) after repeating the experiments five times.

Dataset	Metric	POP	LightGCN	FPMC	Caser	GRU4Rec	SASRec	BERT4Rec	FMLP-Rec	MQSA-TED	Improv.
Beauty	HR@5	0.0077	0.0374	0.0596	0.0359	0.0489	0.0694	0.0419	0.0698	**0.0752***	7.23%
	NDCG@5	0.0042	0.0247	0.0419	0.0241	0.0342	0.0492	0.0275	0.0488	**0.0534***	8.58%
	HR@10	0.0135	0.0571	0.0838	0.0511	0.0695	0.0932	0.0647	0.0995	**0.1039***	4.44%
	NDCG@10	0.0061	0.0311	0.0497	0.0290	0.0408	0.0568	0.0349	0.0583	**0.0627***	7.48%
	HR@20	0.0217	0.0841	0.1151	0.0720	0.0998	0.1286	0.0992	0.1361	**0.1435***	5.40%
	NDCG@20	0.0081	0.0379	0.0576	0.0343	0.0484	0.0657	0.0435	0.0675	**0.0726***	7.62%
Sports	HR@5	0.0057	0.0252	0.0337	0.0195	0.0221	0.0380	0.0241	0.0415	**0.0455***	9.52%
	NDCG@5	0.0041	0.0170	0.0234	0.0128	0.0143	0.0267	0.0161	0.0287	**0.0320***	11.34%
	HR@10	0.0091	0.0384	0.0499	0.0290	0.0357	0.0541	0.0380	0.0598	**0.0643***	7.48%
	NDCG@10	0.0052	0.0212	0.0286	0.0159	0.0187	0.0318	0.0206	0.0346	**0.0380***	9.85%
	HR@20	0.0175	0.0576	0.0703	0.0431	0.0548	0.0752	0.0583	0.0847	**0.0906***	6.93%
	NDCG@20	0.0073	0.0260	0.0337	0.0195	0.0235	0.0371	0.0257	0.0409	**0.0446***	9.09%
Toys	HR@5	0.0065	0.0378	0.0664	0.0307	0.0420	0.0736	0.0379	0.0785	**0.0834***	6.24%
	NDCG@5	0.0044	0.0251	0.0463	0.0224	0.0297	0.0533	0.0244	0.0570	**0.0600***	5.31%
	HR@10	0.0090	0.0564	0.0925	0.0420	0.0597	0.0989	0.0589	0.1062	**0.1130***	6.42%
	NDCG@10	0.0052	0.0311	0.0547	0.0260	0.0354	0.0615	0.0312	0.0659	**0.0696***	5.56%
	HR@20	0.0143	0.0795	0.1212	0.0597	0.0834	0.1299	0.0857	0.1399	**0.1503***	7.41%
	NDCG@20	0.0065	0.0370	0.0619	0.0305	0.0414	0.0693	0.0379	0.0743	**0.0789***	6.23%
Yelp	HR@5	0.0056	0.0290	0.0272	0.0199	0.0211	0.0232	0.0264	0.0270	**0.0320***	10.18%
	NDCG@5	0.0036	0.0184	0.0173	0.0129	0.0134	0.0151	0.0169	0.0169	**0.0205***	11.74%
	HR@10	0.0096	0.0486	0.0433	0.0334	0.0367	0.0379	0.0441	0.0446	**0.0517***	6.36%
	NDCG@10	0.0049	0.0246	0.0224	0.0172	0.0184	0.0198	0.0226	0.0225	**0.0269***	8.95%
	HR@20	0.0158	0.0790	0.0695	0.0535	0.0603	0.0623	0.0737	0.0721	**0.0832***	5.24%
	NDCG@20	0.0065	0.0323	0.0290	0.0222	0.0244	0.0259	0.0300	0.0294	**0.0348***	7.62%

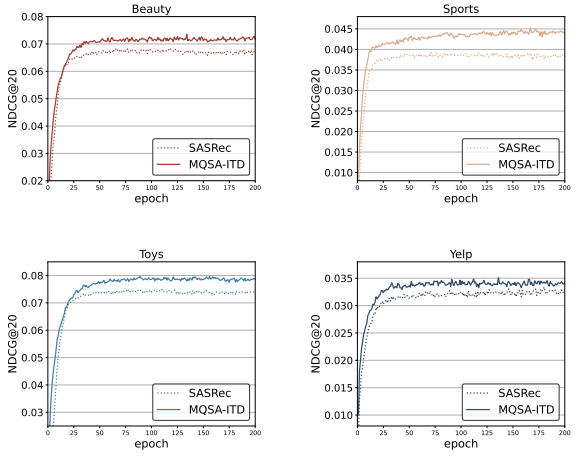


Figure 3: Performance curves of SASRec and our proposed MQSA-TED on four datasets.

- **FMLP-Rec** [32]: an *MLP*-based method that is currently the state-of-the-art sequential recommendation model based on filter-enhanced MLP.

4.1.3 Evaluation Metrics. We adopt Hit Ratio@N (HR@N) and NDCG@N to evaluate the performance of the methods on the sequential recommendation task [31, 32]. We set $N = 5, 10, 20$ by default and report the average scores of users. For each user, we rank all items except for the positive ones in their training or validation data [11]. To ensure the robustness of the results, we randomly initialize each model five times and report the average performance.

4.1.4 Implementation and Hyperparameter Settings. We implement all models with TensorFlow and use the cross-entropy loss for all models for a fair comparison, which has been proved to outperform the negative sampling-based losses significantly [13]. For common hyperparameters in all models, the maximum sequence length is set to 50, the embedding size d is set to 64, the learning rate is tuned in {5e-3, 1e-3, 5e-4, 1e-4}, and the l_2 regularization is tuned in {0, 1e-6, 1e-5, 1e-4, 1e-3}. All models are trained with mini-batch Adam [9], in the batch sizes of 256. Other hyperparameters of different models are tuned on the validation set according to the suggestions in their respective papers. The results of baseline methods under their optimal hyperparameter settings are reported.

4.2 Main Results (RQ1)

Table 2 presents a performance comparison of different methods. The results show that, on Amazon datasets, sequential methods such as FPMC, SASRec, and FMLP-Rec outperform the non-sequential method LightGCN significantly. Among the sequential methods, FMLP-Rec performs the best. However, on the Yelp dataset, LightGCN outperforms the sequential methods due to the weak sequentiality of user interactions on Yelp [34]. Furthermore, our proposed method significantly outperforms all baseline methods, with

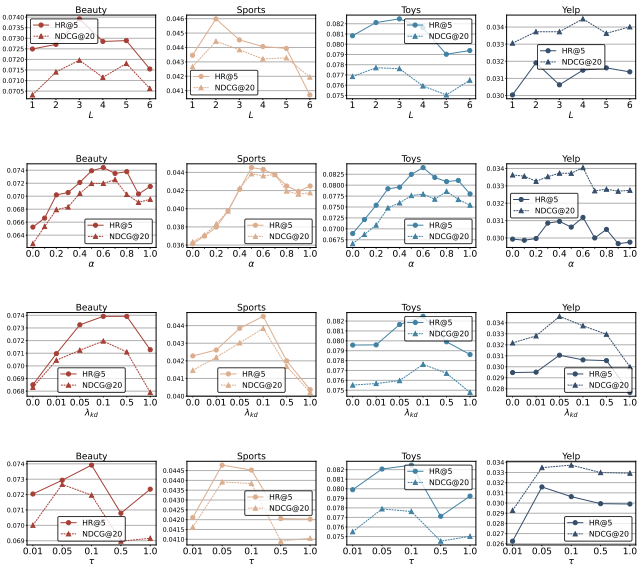


Figure 4: Performance of the proposed MQSA-TED w.r.t. various hyperparameters on four datasets.

an average improvement of 6.24% in Hit Ratio@20 and 7.64% in NDCG@20 compared to the best baseline.

Figure 3 shows the performances of SASRec and our proposed method with respect to the training epochs. One can observe that our proposed method consistently outperforms SASRec by a notable margin, showing the effectiveness of the proposed modules.

4.3 Hyperparameter and Ablation Studies (RQ2)

Figure 4 presents the performance of our proposed method with respect to various hyperparameters and modules:

4.3.1 Length of Long-Query Self-Attention L. It can be observed the best L depends on the datasets and the model generally performs well when L is in the range of $[2, 4]$, showing the effectiveness of long-query self-attention in capturing collaborative signals.

4.3.2 Balance of Long and Short-Query Self-Attention α. The results show that when α is approximately 0.5, the model achieves the best performance, indicating a proper bias-variance trade-off in modeling user interests. Notably, when $\alpha = 1$, the model degrades to SASRec with TED. Therefore, the proposed multi-query self-attention significantly outperforms the short-query self-attention used in SASRec with a proper α.

4.3.3 Weight of Embedding Distillation λ_{kd}. It can be seen that the model performs better when λ_{kd} is approximately 0.1, demonstrating the effectiveness of the TED module. Note that when $\lambda_{kd} = 0$, our proposed method degrades to the MQSA model without TED, resulting in a significant drop in performance.

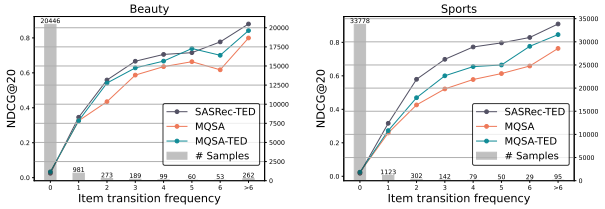
4.3.4 Temperature of Embedding Distillation τ. The results suggest that the model requires relatively hard pseudo-labels of item transition distributions for effective knowledge distillation, as the best performance is achieved when $\tau = 0.05$ or $\tau = 0.1$.

Table 3: Performance comparison of the proposed TED module with graph-based methods on four datasets. The best results are in boldface and the second best are underlined.

Dataset	Metric	MQSA	+GES	+GraReg	+TED
Beauty	NDCG@10	0.0599	<u>0.0623</u>	0.0611	**0.0627**
	NDCG@20	0.0694	<u>0.0724</u>	0.0708	**0.0726**
Sports	NDCG@10	0.0344	<u>0.0370</u>	0.0351	**0.0380**
	NDCG@20	0.0408	<u>0.0434</u>	0.0416	**0.0446**
Toys	NDCG@10	0.0654	<u>0.0672</u>	0.0667	**0.0696**
	NDCG@20	0.0749	<u>0.0765</u>	0.0755	**0.0789**
Yelp	NDCG@10	0.0255	0.0244	<u>0.0257</u>	**0.0269**
	NDCG@20	0.0327	0.0320	<u>0.0330</u>	**0.0348**

Table 4: Performance comparison of LightGCN and FMLP-Rec w/ and w/o the proposed TED module on four datasets. The best results under each backbone are in boldface.

Dataset	Metric	LightGCN	+TED	FMLP-Rec	+TED
Beauty	NDCG@10	0.0311	**0.0399**	0.0583	**0.0596**
	NDCG@20	0.0379	**0.0484**	0.0675	**0.0684**
Sports	NDCG@10	0.0212	**0.0246**	0.0346	**0.0356**
	NDCG@20	0.0260	**0.0298**	0.0409	**0.0423**
Toys	NDCG@10	0.0311	**0.0388**	0.0659	**0.0675**
	NDCG@20	0.0370	**0.0459**	0.0743	**0.0762**
Yelp	NDCG@10	**0.0246**	0.0236	0.0225	**0.0226**
	NDCG@20	**0.0323**	0.0312	0.0294	**0.0296**



Dataset	Trans. Freq.	SASRec-TED	MQSA	MQSA-TED	# Samples
Beauty	0	0.0254	0.0318	0.0329	20,446
	>0	0.5228	0.4665	0.4979	1,917
	All	0.0681	0.0691	0.0728	22,363
Sports	0	0.0183	0.0237	0.0244	33,778
	>0	0.4620	0.3635	0.3962	1,820
	All	0.0410	0.0411	0.0434	35,598

Figure 5: Performance of three methods w.r.t. item transition frequency on two datasets. MQSA-TED outperforms MQSA on test samples with frequent transitions and outperforms SASRec-TED on test samples lacking transition instances.

4.4 Comparison with Graph-Based Regularization Methods (RQ3)

We also compare the proposed Transition-Aware Embedding Distillation (TED) module with graph-based regularization methods in Table 3. The results show that most of the methods can improve the performance of MQSA. Specifically, GES performs better than GraReg on Amazon datasets but worse on the Yelp dataset. Moreover, our proposed TED method outperforms GES and GraReg

in most cases, indicating the effectiveness of learning global and accurate item transition patterns by knowledge distillation.

4.5 Transition-Aware Embedding Distillation for Various Base Models (RQ4)

We also compare the performance of various base models with and without our proposed Transition-Aware Embedding Distillation (TED) module in Table 4. The results demonstrate that TED can act as a domain adapter, which enhances the performance of the non-sequential method LightGCN on sequential recommendation tasks. Furthermore, the incorporation of TED yields remarkable improvement for the state-of-the-art sequential recommendation method FMLP-Rec. Notably, TED shows limited effects on the Yelp dataset due to the weak sequentiality of user interactions. In other words, transitional signals are less important in this dataset.

4.6 Performance Comparison by Groups (RQ5)

Figure 5 presents the performance of different methods on test samples grouped by transition frequencies observed in the training data from the validation item (the second last item) to the test item (the last item). We evaluate the SASRec model with the Transition-Aware Embedding Distillation (SASRec-TED), the Multi-Query Self-Attention model (MQSA), and the full MQSA-TED model. Compared with the results in Figure 1, the improvement of MQSA over SASRec mainly results from the improvement on test samples lacking transition instances. However, the integration of long-query self-attention may hurt the performance on test samples with frequent transitions. By incorporating the TED module as a calibrator, MQSA-TED performs better than MQSA mainly on test samples with high transition frequencies. As MQSA and TED focus on collaborative and transitional signals, respectively, their combination will result in a reasonable balance between the two signal types.

5 RELATED WORK

Sequential Recommendation. Sequential recommendation methods aim to capture dynamic user preferences [26]. Early efforts adopt Markov Chains (MCs) to learn item transition patterns, such as FPMC [18], which combines the Matrix Factorization (MF) with the first-order Markov chain. Fossil [3] fuses the similarity-based model with high-order Markov chains. Recent efforts incorporate deep learning-based models, such as GRU4Rec [6], which employs Gated Recurrent Units (GRU), and NARM [14], which enhances GRU with an attention mechanism. Caser [21] uses horizontal and vertical convolutional filters to learn sequential patterns. SASRec [8] and BERT4Rec [20] use unidirectional and bidirectional self-attention modules in Transformer [24] to capture long-term user interests, respectively. FMLP-Rec [32] is an all-MLP model with learnable filters in the frequency domain. However, previous efforts typically follow an auto-regressive framework, which neglects the valuable information in global item transition patterns. In this paper, we propose a Transition-Aware Embedding Distillation module to memorize and leverage the transitional signals.

Self-Attention in Recommendation. The Transformer architecture has achieved remarkable success in modeling long-term dependencies in Natural Language Processing (NLP) [1, 24]. Consequently, recent efforts employ self-attention networks for sequential recommendation tasks. For example, SASRec [8] and BERT4Rec [20] use unidirectional and bidirectional self-attention modules to capture long-term user interests, respectively. In addition, some efforts aim to enhance self-attention-based models by incorporating side information. For instance, TiSASRec [15] incorporates time interval embeddings into SASRec. S^3-Rec [31] introduces self-supervision tasks to learn correlations among attributes, items, sub-sequences, and sequences based on mutual information maximization. SASRec-GES [34] employs graph convolutions on sequential and semantic item graphs to generate smoothed item embeddings. Efforts have also been made to improve the efficiency or effectiveness of SASRec. CL4SRec [27] uses contrastive learning to derive self-supervision signals from user interaction sequences. DuoRec [17] develops a contrastive regularization with model-level augmentation and supervises positive sampling for contrastive samples. Despite these advances, previous studies paid less attention to the limitations of the conventional self-attention architecture in capturing collaborative signals. In this paper, we propose a Multi-Query Self-Attention method that combines long and short-query self-attentions to enhance its effectiveness in modeling user collaborations.

Knowledge Distillation in Recommendation. Knowledge distillation is a widely-used model compression technique in various fields [7], where a student model is trained with both a ground-truth label distribution and a smoothed pseudo-label distribution generated by a teacher modelRecent efforts apply this method to recommender systems, such as Ranking Distillation [22], which trains a student model to rank items based on both training data and teacher model predictions. Collaborative Distillation [12] uses probabilistic rank-aware sampling with teacher-guided and student-guided training strategies. Other existing methods aim to distill side information into recommendation models to enhance their performance and interpretability. For instance, SCML [33] combines the item-based CF model with the social CF model by embedding-level and output-level mutual learning. DESIGN [23] integrates information from the user-item interaction graph and the user-user social graph and makes them learn from each other. Zhang et al. [29] propose a joint learning framework to distill structured knowledge from a path-based model into a neural model. However, knowledge distillation has received less attention in the context of sequential recommendation. In this paper, we distill the knowledge of item transitions into sequential models to enhance their performances.

6 CONCLUSION

In this paper, we investigate the limitations of existing sequential recommendation methods in capturing collaborative and transitional signals in user interaction sequences. To overcome these limitations, we propose a new method called Multi-Query Self-Attention with Transition-Aware Embedding Distillation (MQSA-TED). To capture collaborative signals, we introduce an L-query self-attention module using flexible window sizes for attention queries and combine long and short-query self-attentions. In addition, we develop a transition-aware embedding distillation module that distills global item transition patterns into item embeddings, enabling the model to memorize and leverage transitional signals. Experimental results on four real-world datasets demonstrate the effectiveness of both modules in improving model performance.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[2] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3953–3957.

[3] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.

[4] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th international conference on world wide web*. 507–517.

[5] 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.

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

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

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

[9] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[10] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[11] Walid Krichene and Steffen Rendle. 2022. On sampled metrics for item recommendation. *Commun. ACM* 65, 7 (2022), 75–83.

[12] Jae-woong Lee, Minjin Choi, Jongwuk Lee, and Hyunjung Shim. 2019. Collaborative distillation for top-N recommendation. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 369–378.

[13] Fangyu Li, Shenbao Yu, Feng Zeng, and Fang Yang. 2023. Effective and Efficient Training for Sequential Recommendation Using Cumulative Cross-Entropy Loss. *arXiv preprint arXiv:2301.00979* (2023).

[14] 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.

[15] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th ACM international conference on web search and data mining*. 322–330.

[16] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th ACM SIGIR international conference on research and development in information retrieval*. 43–52.

[17] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 813–823.

[18] 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.

[19] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[20] 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.

[21] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

[22] Jiaxi Tang and Ke Wang. 2018. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2289–2298.

[23] Ye Tao, Ying Li, Su Zhang, Zhirong Hou, and Zhonghai Wu. 2022. Revisiting Graph based Social Recommendation: A Distillation Enhanced Social Graph Network. In *Proceedings of the ACM Web Conference 2022*. 2830–2838.

[24] 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).

[25] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. In *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1816–1825.

[26] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make it a chorus: knowledge-and time-aware item modeling for sequential recommendation. In *Proceedings of the 43rd ACM SIGIR International conference on research and development in Information Retrieval*. 109–118.

[27] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.

[28] Yuan Zhang, Fei Sun, Xiaoyong Yang, Chen Xu, Wenwu Ou, and Yan Zhang. 2020. Graph-based regularization on embedding layers for recommendation. *ACM Transactions on Information Systems (TOIS)* 39, 1 (2020), 1–27.

[29] Yuan Zhang, Xiaoran Xu, Hanning Zhou, and Yan Zhang. 2020. Distilling structured knowledge into embeddings for explainable and accurate recommendation. In *Proceedings of the 13th ACM international conference on web search and data mining*. 735–743.

[30] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

[31] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1893–1902.

[32] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*. 2388–2399.

[33] Tianyu Zhu, Guannan Liu, and Guoqing Chen. 2020. Social collaborative mutual learning for item recommendation. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14, 4 (2020), 1–19.

[34] Tianyu Zhu, Leilei Sun, and Guoqing Chen. 2021. Graph-based embedding smoothing for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (2021), 496–508.