Mining Interest Trends and Adaptively Assigning Sample Weight for Session-based Recommendation

Kai Ouyang[*]
Xianghong Xu[*]
oyk20@mails.tsinghua.edu.cn
xxh20@mails.tsinghua.edu.cn
SIGS, Tsinghua University
Shenzhen, China

Miaoxin Chen
cmx20@mails.tsinghua.edu.cn
Shezhen International Graduate School, Tsinghua University
Shenzhen, China

Zuotong Xie
xiezt20@mails.tsinghua.edu.cn
Shezhen International Graduate School, Tsinghua University
Shenzhen, China

Hai-Tao Zheng[†]
zheng.haitao@sz.tsinghua.edu.cn
Shezhen International Graduate School, Tsinghua University
Shenzhen, China

Shuangyong Song
songshy@chinatelecom.cn
China Telecom Corporation Ltd. Data&AI Technology Company
Beijing, China

Yu Zhao
zhaoy11@chinatelecom.cn
China Telecom Corporation Ltd. Data&AI Technology Company
Beijing, China

ABSTRACT

Session-based Recommendation (SR) aims to predict users' next click based on their behavior within a short period, which is crucial for online platforms. However, most existing SR methods somewhat ignore the fact that user preference is not necessarily strongly related to the order of interactions. Moreover, they ignore the differences in importance between different samples, which limits the model-fitting performance. To tackle these issues, we put forward the method, **M**ining Interest **T**rends and **A**daptively Assigning Sample **W**eight, abbreviated as **MTAW**. Specifically, we model users' instant interest based on their present behavior and all their previous behaviors. Meanwhile, we discriminatively integrate instant interests to capture the changing trend of user interest to make more personalized recommendations. Furthermore, we devise a novel loss function that dynamically weights the samples according to their prediction difficulty in the current epoch. Extensive experimental results on two benchmark datasets demonstrate the effectiveness and superiority of our method.

CCS CONCEPTS

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

KEYWORDS

Session-based Recommendation, Weight Assignment, Attention

ACM Reference Format:
Kai Ouyang, Xianghong Xu, Miaoxin Chen, Zuotong Xie, Hai-Tao Zheng, Shuangyong Song, and Yu Zhao. 2023. Mining Interest Trends and Adaptively Assigning Sample Weight for Session-based Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3539618.3592021

[*]Both authors contributed equally to this research.
[†]Corresponding author.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9408-6/23/07.
https://doi.org/10.1145/3539618.3592021

1 INTRODUCTION

In many real-world recommendation scenarios, users are usually unable to be identified or tracked due to privacy policies. Therefore, recommendation systems need to discern latent user interests based on sparse interaction data in the absence of user information. To address this challenge, Session-based Recommendation (SR) has been developed. With the boom of e-commerce platforms, SR has attracted increasing attention from academia and industry.

Most existing research efforts in SR regard the session as a strictly ordered sequence [19]. At the beginning of SR research, many RNN-based models [1–3, 14] have been proposed. For example, GRU4Rec [2] captures user latent interest by modeling the sequence information of user interactions. In recent years, researchers have proposed many GNN-based SR models [10–12, 15, 16, 18]. They model the session as a directed graph and focus on modeling the transitions between adjacent interactions with the pairwise relationships of nodes. Besides, most SR methods treat all samples equally during model training. They are optimized by treating samples indiscriminately with cross-entropy or InfoNCE [9] loss function. These methods achieve considerable performance improvement due to their effective capture of sequence information and optimization way like contrastive learning.

However, they have two defects. **(a)** These SR models implicitly follow a strong assumption, *i.e.*, the relative order of adjacent interactions is strongly associated with the users' interests. However, the relative order of users' behaviors does not have an absolute correlation with their interests, and some behaviors may just be noise signals. For instance, some user interactions may result from accidental clicking or the random recommendation function of APPs. Thus, strictly modeling timing can increase the impact of these noises, ultimately limiting the model's performance. **(b)** They ignore the difference between samples. In reality, different sessions have varying numbers and credibility of interactions, resulting in different prediction difficulties. This, in turn, makes their importance in model fitting different. Hence, treating all samples equally reduces the fitting performance of the model.

To tackle these issues, we propose capturing the changing trend of user interest, rather than focusing on modeling the relative order between interactions. Moreover, we suggest assigning importance weight to the samples, rather than treating them equally. Hence, we put forward method, **M**ining Interest **T**rends and **A**daptively Assigning Sample **W**eight, abbreviated as **MTAW**. Specifically, we capture the instant interest at the current moment according to user interactions at the previous moments and their position information. We integrate these instant interests discriminatively to mine the changing trend of user interest. Compared to order information, trend information on interest changing is more reliable. Additionally, we devise the Adaptive Weight (AW) loss function, which adaptively assigns weights to different samples according to their prediction difficulty in the current epoch. This makes the model pay more attention to hard samples in the model-fitting process, improving its effectiveness. To summarize, we make the following contributions:

- We model the changing trends of users' interests. Technically, we track users' instant interests at each moment and then integrate these interests discriminatively to achieve more personalized recommendations.
- We devise the AW loss function. It adaptively assigns weights to different samples and enables us to focus more on difficult samples to enhance the model-fitting effect.
- Experimental results on two datasets demonstrate that MTAW overwhelmingly outperforms the state-of-the-art (SOTA) methods. Moreover, MTAW is far more efficient and requires fewer parameters than SOTA methods.

2 METHOD

The basic idea of MTAW is to mine the trend of user dynamic interest and assign different importance weights to different samples. The architecture of MTAW is depicted in Figure 1. It can be mainly divided into the following parts: **(1)** User Interest Modeling, which takes two steps to capture the user's interest evolving: *(i)* Interest Tracking Layer. *(ii)* Interest Enhancing Layer. **(2)** Recommendation and Optimization, which makes recommendations based on the session representation and assigns different weights to samples according to their difficulty.

2.1 Problem Statement

Let $I = \{i_1, i_2, \ldots, i_N\}$ denote the set of all unique items involved in all the sessions, where N is the total number of items. Each session is represented as a list $s = [i_{s,1}, i_{s,2}, \ldots, i_{s,m}]$ ordered by timestamps, where m is the length of the session s and $i_{s,k} \in I (1 \leq k \leq m)$ represents a clicked item of the user within the session s. The goal of the task is to predict the next click for the session s, *i.e.*, the sequence label $i_{s,m+1}$. For the session s, we calculate the probabilities $\hat{y}$ for all possible items, where the recommendation score of an item is the corresponding element of vector $\hat{y}$. The items corresponding to the top-K scores will be recommended.

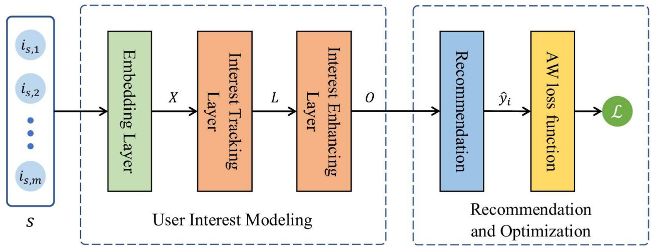

Figure 1: The architecture of MTAW.

2.2 User Interest Modeling

To convert the input session into vectors, we construct the Embedding Layer. For each item i in the input session, the hidden representation is:

$$x_i = e_i + p_i, \tag{1}$$

where $e_i \in \mathbb{R}^d$ is item embedding, d is the embedding size, $p_i \in \mathbb{R}^d$ is the position embedding, and $x_i \in \mathbb{R}^d$ denotes the hidden representation of item i. Besides, we use $X = \{x_1, x_2, \ldots, x_m\}$ to denote the embedding set of session $s = [i_1, i_2, \ldots, i_m]$.

2.2.1 Interest Tracking Layer. User interests are usually dynamic, and user behavior sequence is the carrier of user interest. Therefore, to capture user real interest, we need to mine the instant interest sequence based on the user behavior sequence. Specifically, we employ the Attention network to extract user instant interest at the current moment according to its previous interactions. Formally, the Attention network can be defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{Q^\top K}{\sqrt{d}})V, \tag{2}$$

where Q, K, and V are the input matrices. To ensure that the extracts for t-th item can depend only on its previous items, we derive m slices from X in chronological order: $X'_1 = \{x_1\}$, $X'_2 = \{x_1, x_2\}, \ldots, X'_m = \{x_1, x_2, \ldots, x_m\}$. Then, for each slice like $X'_t = \{x_1, x_2, \ldots, x_t\}$, where $t \leq m$ represents the serial number of slices, we adopt the Attention network to extract user instant interest:

$$\begin{aligned} Q_x &= \text{ReLU}(\text{MLP}(x_t)), \\ l'_t &= \text{Attention}(Q_x, X'_t, X'_t), \end{aligned} \tag{3}$$

where $l'_t \in \mathbb{R}^d$, MLP denotes the multi-layer perceptron, and ReLU is the activation function. We perform the above process in parallel by using a mask matrix in the Attention network. The output is hidden state set $L' = \{l'_1, l'_2, \ldots, l'_m\}$ for session s. Moreover, we apply the *Position-wise Feed-Forward Network (FFN)* to endow the model with more non-linearity:

$$L = FFN(L') = \text{MLP}(\text{ReLU}(\text{MLP}(L'))), \tag{4}$$

where two MLPs represent two different multi-layer perceptrons. Then, we add a residual connection and layer normalization on the result to alleviate the instability of the model training. We also add the dropout mechanism to alleviate the overfitting. For simplicity, we denote the *Interest Tracking Layer* as ITL, *i.e.*, $L = \text{ITL}(X)$, where $L = \{l_1, l_2, \ldots, l_m\}$, $l_m \in \mathbb{R}^d$ is the final output of this layer, and each of them denotes the user instant interest of current interaction.

2.2.2 *Interest Enhancing Layer.*

The evolution of user interest will directly affect the user's choice of the next item. Therefore, we design the Interest Enhancing Layer to conduct in-depth excavation and analysis of the evolution process of the user's interest. Besides, this layer injects information about interest evolution trends into the session representation.

Specifically, to obtain the changing trend of users' interest in the next item, we adopt the same Attention network to discriminatively integrate the instant interests:

$$O = \text{Attention}(Q_l, L, L), \tag{5}$$

where Q_l is the last element l_m of L, and $O \in \mathbb{R}^d$ represent the session representation. In a word, we use the user's last instant interest to be the query, and the whole instant interest sequence to be the key. It can learn the changing trend that hides behind the evolution process of user interest.

2.3 Recommendation and Optimization

In this layer, we complete the prediction of the current session and the model optimization process.

Recommendation. To make recommendations, for each item $i \in I$, we get the ranking score as follows:

$$\begin{aligned} \hat{O} &= \text{L2Norm}(O),\ \hat{x}_i = \text{L2Norm}(x_i), \\ \hat{y}_i &= \text{softmax}(\hat{O}^T \hat{x}_i), \end{aligned} \tag{6}$$

where x_i is the embedding of item i, L2Norm is the L2 Normalization function and $\hat{y}_i$ denotes the final probability of item i.

Optimization. There are differences between samples and the difficulty of models in predicting different sessions. Besides, easily classified negatives comprise the majority of the loss and dominate the gradient [7]. Therefore, we propose to assign different weights to different samples. Inspired by Focal loss [7], we assign weights based on the prediction deviation of samples in the current epoch.

More formally, we propose the Adaptive Weight (AW) loss function which adds a modulating factor to the cross-entropy loss function. The AW Loss can be formulated as follows:

$$\begin{aligned} p_i &= \begin{cases} \hat{y}_i, & \text{if } y = 1, \\ 1 - \hat{y}_i, & \text{otherwise}, \end{cases} \\ \mathcal{L} &= -\sum_{i=1}^{M} (2 - 2p_i)^{\gamma} \log(p_i), \end{aligned} \tag{7}$$

where γ is the temperature coefficient and as γ is increased the effect of the modulating factor is increased. y is the ground truth probability distribution of the next item, which is a one-hot vector. M is the total number of samples. Since $p_i \in [0, 1]$, $(2 - 2p_i)$ is around 1. $(2 - 2p_i)^{\gamma}$ indicates the deviation between the predicted value and the ground truth of the sample, *i.e.*, the difficulty of the sample in the current epoch. Intuitively, the modulating factor reduces the loss contribution of simple samples and expands that of hard samples.

3 EXPERIMENTS

3.1 Settings

3.1.1 *Datasets and Metrics.*

For a fair comparison, we conduct the experiment on the two public datasets: **Tmall**[1] comes from IJCAI-15 competition, which contains anonymous shopping logs on Tmall online platform. **RetailRocket**[2] comes from a Kaggle contest and contains the browsing activity of users within six months. For a fair comparison, we implement our model on the public pre-processed version datasets provided by S^2-DHCN [3]. The statistics of the two datasets after preprocessing are represented in Table 1.

We use the same evaluation metrics as the previous works [16, 17]: MRR@K (Mean Reciprocal Rank at K) and P@K (Precision at K). The values of K include 10 and 20.

Table 1: Statistics of RetailRocket and Tmall Datasets

Dataset	# training	# test	# items	Avg. Len.
RetailRocket	433,643	15,132	36,968	5.43
Tmall	351,268	25,898	40,728	6.69

3.1.2 *Baselines and Implementation Details.*

We compare our model with the following representative SR methods: **FPMC** [13] is a sequential method based on Markov Chain. **GRU4REC** [2], which is a representative sequential model based on Gated Recurrent Unit (GRU). **NARM** [6] utilizes the self-attention mechanism and GRU to capture the main purpose of the session. **STAMP** [8], which uses the self-attention mechanism to represent the intent of the session, and emphasizes the importance of the last click in each session. **SASRec** [4] solely uses self-attention mechanism. **NextItNet** [20], which is the best performing CNN-based SBR model. **SR-GNN** [16] models separated session sequences into graph-structured data and uses graph neural networks to capture complex item transitions. **GC-SAN** [18], which combines GNN and multi-layer self-attention to make recommendations. **GCE-GNN** [15] is a widely compared GNN-based SBR model that learns global and local information of sessions. **S^2-DHCN** [17] constructs two types of hypergraphs to learn inter- and intra-session information and introduces self-supervised learning.

For a fair comparison, we follow S^2-DHCN to make the following settings: We use Adam [5] optimizer with a learning rate of 0.001. We set the embedding size to 100, the number of epochs to 50, and the batch size to 100. Besides, we set γ to 2 for the Tmall dataset, and set γ to 6 for the RetailRocket dataset.

3.2 Results

3.2.1 *Overall Performance.*

The experimental results of all models are reported in Table 2. Based on the results, we can draw the following conclusions: **(a)** These traditional models (*e.g.*, Item-KNN, FPMC) can even outperform the first method based on RNNs (*i.e.*, GRU4REC) in terms of some metrics. It indicates that solely modeling a session as a strictly ordered sequence may result in limiting

[1]https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
[2]https://www.kaggle.com/retailrocket/ecommerce-dataset
[3]https://github.com/xiaxin1998/DHCN

Table 2: Performance comparison on two datasets (%). In each metric, the best result is highlighted in boldface and the second best is underlined. And † indicates statistic significant improvement over all baseline models for t-test with p-value < 0.01.

Method	Tmall				RetailRocket			
	P@10	MRR@10	P@20	MRR@20	P@10	MRR@10	P@20	MRR@20
FPMC [13]	13.10	7.12	16.06	7.32	25.99	13.38	32.37	13.82
GRU4REC [2]	14.16	6.56	18.20	6.85	34.41	15.06	44.89	15.77
NARM [6]	19.17	10.42	23.30	10.70	42.07	24.88	50.22	24.59
STAMP [8]	22.63	13.12	26.47	13.36	42.95	24.61	50.96	25.17
SASRec [4]	22.06	14.02	26.95	14.21	44.65	25.53	51.12	25.91
NextItNet [20]	22.67	13.12	27.22	13.32	41.12	23.99	48.26	24.48
SR-GNN [16]	23.41	13.45	27.57	13.72	43.21	26.07	50.32	26.57
GC-SAN [18]	21.32	12.43	25.38	12.72	43.21	26.07	50.32	26.57
GCE-GNN [15]	<u>28.02</u>	<u>15.08</u>	<u>33.42</u>	<u>15.42</u>	46.05	<u>27.48</u>	53.63	<u>28.01</u>
S^2-DHCN [17]	26.22	14.60	31.42	15.05	<u>46.15</u>	26.85	<u>53.66</u>	27.30
MTAW	**31.67**†	**18.90**†	**37.17**†	**19.14**†	**48.41**†	**29.96**†	**56.39**†	**30.52**†
Improv. (%)	13.03	25.33	11.22	24.12	4.90	9.02	5.09	8.96

the ability to capture users' genuine interest. GNNs-based models outperform most models because GNNs can model the transitions between adjacent items. However, they model the session as a directed graph, which actually models the session as a strictly ordered sequence, so their performance is surpassed by MTAW. **(b)** MTAW overwhelmingly outperforms all baseline models. This demonstrates the superiority of interest trend modeling and adaptive assignment of sample weights.

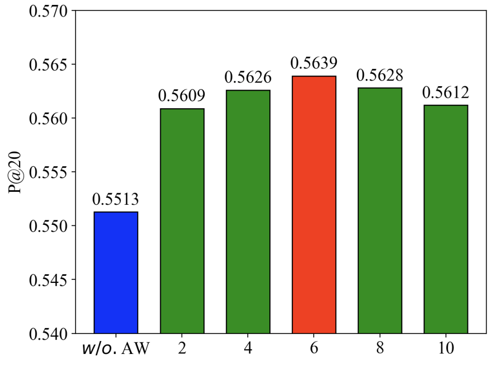

Figure 2: AW Loss Study on RetailRocket.

3.2.2 *Study on AW Loss Function.*

To verify the superiority of the AW loss function, and investigate the impact of hyper-parameter γ on the final performance, we conduct further experiments. We search the γ in the range of $[2, 4, \ldots, 10]$ in terms of P@20 on the RetailRocket dataset. Besides, we compare them with variant $w/o.$ AW which uses the normal cross-entropy loss function for optimization. The results are shown in Figure 2, and we can conclude that with the increase of the value γ, the performance of MTAW first increases and then decreases. When the γ is set to 6, MTAW performs best. Moreover, when MTAW is optimized without the AW loss function, the performance will decline, which shows the effectiveness of the AW loss function.

Table 3: Training time per epoch and the number of trainable parameters, where s, m, and M respectively represent second, minute, and million.

Method	Tmall		RetailRocket	
	Time	#Params	Time	#Params
NextItNet	34m51s	4.23M	64m27s	3.85M
SR-GNN	4m7s	4.23M	23m54s	3.86M
GC-SAN	3m42s	4.24M	12m20s	3.87M
GCE-GNN	2m20s	4.35M	16m02s	3.98M
S^2-DHCN	32m34s	4.31M	77m12m	3.94M
MTAW	**47s**	**4.02M**	**58s**	**3.69M**

3.2.3 *Efficiency Comparison.*

To evaluate the efficiency of MTAW, we compare the training time per epoch and trainable parameters with recent SOTA models on the same device. The results are shown in Table 3. We can observe that MTAW is far more efficient and requires fewer parameters than recent SOTA methods. Compared with GCE-GNN [15], MTAW achieves 16.59× speed up with fewer parameters on the RetailRocket dataset. We can conclude that MTAW is both efficient and effective, and modeling interest trends is a potential future work.

4 CONCLUSION

In this paper, we propose MTAW, which mines the changing trend of user interest and adaptively adjusts sample weights for SR. In MTAW, we use the attention mechanism to capture users' instantaneous interests. Furthermore, we integrate these interests to mine the trend of changing interests. Additionally, we devise the AW loss function to dynamically assign sample weights. Extensive experiments on two datasets demonstrate the superiority of MTAW.

ACKNOWLEDGMENT

This research is supported by National Natural Science Foundation of China (Grant No.62276154), Research Center for Computer Network (Shenzhen) Ministry of Education, Beijing Academy of Artificial Intelligence (BAAI), the Natural Science Foundation of Guangdong Province (Grant No. 2023A1515012914), Basic Research Fund of Shenzhen City (Grant No. JCYJ20210324120012033 and JSGG20210802154402007), and the Major Key Project of PCL for Experiments and Applications (PCL2021A06).

REFERENCES

[1] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *CIKM*. 843–852.

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

[3] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *RecSys*. 306–310.

[4] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[5] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[6] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *CIKM*. 1419–1428.

[7] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[8] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *SIGKDD*. 1831–1839.

[9] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[10] Kai Ouyang, Xianghong Xu, Chen Tang, Wang Chen, and Haitao Zheng. 2022. Social-aware Sparse Attention Network for Session-based Social Recommendation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2173–2183.

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

[12] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In *CIKM*. 579–588.

[13] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*. 811–820.

[14] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 17–22.

[15] 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 *SIGIR*. 169–178.

[16] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *AAAI*, Vol. 33. 346–353.

[17] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation. In *AAAI*. 4503–4511.

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

[19] Xianghong Xu, Kai Ouyang, Liuyin Wang, Jiaxin Zou, Yanxiong Lu, Hai-Tao Zheng, and Hong-Gee Kim. 2022. Modeling Latent Autocorrelation for Session-Based Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 4605–4609.

[20] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *WSDM*. 582–590.