FPAdaMetric: False-Positive-Aware Adaptive Metric Learning for Session-Based Recommendation

Jongwon Jeong[1], **Jeong Choi**[2], **Hyunsouk Cho**[3*†], **Sehee Chung**[1]

[1] Knowledge AI Lab, NCSOFT Corp., Republic of Korea
[2] NAVER Corp., Republic of Korea
[3] Ajou University, Republic of Korea
jwjeong@ncsoft.com, jeong.choi@navercorp.com, hyunsouk@ajou.ac.kr, seheechung@ncsoft.com

Abstract

Modern recommendation systems are mostly based on implicit feedback data which can be quite noisy due to false positives (FPs) caused by many reasons, such as misclicks or quick curiosity. Numerous recommendation algorithms based on collaborative filtering have leveraged post-click user behavior (*e.g.*, skip) to identify false positives. They effectively involved these false positives in the model supervision as negative-like signals. Yet, false positives had not been considered in existing session-based recommendation systems (SBRs) although they provide just as deleterious effects. To resolve false positives in SBRs, we first introduce **FP-Metric** model which reformulates the objective of the session-based recommendation with FP constraints into metric learning regularization. In addition, we propose **FP-AdaMetric** that enhances the metric-learning regularization terms with an adaptive module that elaborately calculates the impact of FPs inside sequential patterns. We verify that **FP-AdaMetric** improves several session-based recommendation models' performances in terms of Hit Rate (HR), MRR, and NDCG on datasets from different domains including music, movie, and game. Furthermore, we show that the adaptive module plays a much more crucial role in **FP-AdaMetric** model than in other baselines.

Introduction

Recommendation Systems based on collaborative filtering have received great attention due to their outstanding performance in numerous personalized services (Zhang et al. 2019). Although explicit feedback data such as user ratings can give direct supervision regarding user preferences, they are often expensive and lack in size in real world scenarios (Rendle et al. 2012). Alternatively, implicit feedback data have been widely adopted as the main resource for training recommendation models (Hu, Koren, and Volinsky 2008; Yi et al. 2014; He et al. 2017; Guo et al. 2017). They are solely based on user behavior logs and thus much easier to collect (Koren, Bell, and Volinsky 2009; Rendle et al. 2012).

However, in multimedia streaming services such as music or movies, the user preference may not be revealed in the click logs because users often make decisions after they try consuming an item to some degree. Users show their preference by either continuing or skipping in the early section. This behavior is even stronger in subscription-based services because users have full access of items at zero additional cost. For example, 50% of positive signals are false positives (FPs) in music streaming services, such as skips or backward button clicks within 10 seconds, meaning that half of click logs are triggered out of simple curiosity (Wen, Yang, and Estrin 2019). Therefore, several works on collaborate filtering dealt with how to consider FPs better by dwell-time (Yi et al. 2014), skip (Wen, Yang, and Estrin 2019) or other ways (Wu et al. 2020; Wang et al. 2021b).

Meanwhile, the sequential patterns in user feedbacks have become a major factor of the recommendation problem, and have been extensively studied in the field of session-based recommendation system (SBRs) (Fang et al. 2020; Wang et al. 2021a). Since gradual changes take place in user preferences over time, SBRs splits each user's entire log sequence into a number of session-level segments to conjugate both local and global preferences. Although the SBRs models have been successful in capturing user preferences, the noisiness or the underlying FPs of the implicit feedback data are mostly neglected. We argue that they could be just as harmful in SBRs tasks as in non-sequential cases (Yi et al. 2014; Wen, Yang, and Estrin 2019; Wu et al. 2020; Wang

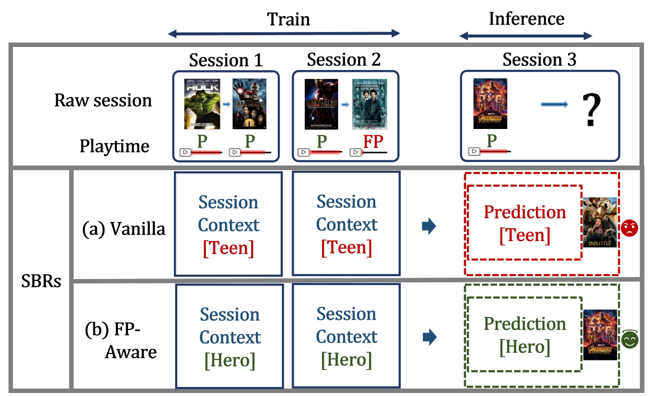


Figure 1: Example about Importance of False-Positive(FP) on SBRs in Movie service. FPs depend on playtime of items, and the user does not like "Teen" movie.

*This work was done while working at NCSOFT.
†Corresponding author

et al. 2021b). A few works have taken FPs into account in certain sequential recommendation tasks (Zhao et al. 2018b,a; Xie et al. 2020; Wang and Cao 2021; Bian et al. 2021), however, to the best of our knowledge, no work has been done towards the SBRs problems. As shown in Figure 1, FPs could play crucial roles especially in SBRs scenarios. Figure 1.(a) depicts a user scenario where FPs are not considered. Session context contained "Hero" hurts by "Teen" movie as in session #2 so that recommended item can be "Teen" movie. However, as shown in Figure 1.(b), we get better recommendation results when FPs are considered. A FP-aware method helps to avoid further FPs in the sequential recommendation, resulting in "Hero" to be recommended, which better meets the true user preference.

To directly resolve the FPs in the SBRs problem, we first define a general objective function with constraints where the FP item embedding should necessarily be far away from the current sequence embedding. This constraint can be applied due to our assumption 1 of FPs in SBRs. Then, we show that this optimization is equivalent to learning an embedding function that maps data into our desired metric space by using a triplet loss objective (Kaya and Bilge 2019). We call this **FP-Metric**. In addition, we propose **FP-AdaMetric**, a novel architecture which better represents FP items during training by learning an additory function that adapts the input and positive embeddings for applying degree of FPs differently. We show that this adaptive module improves the triplet loss-based metric learning procedure.

We evaluate our proposed method in the datasets from different multimedia streaming services, *i.e.*, LastFM, Spotify, Amazon Movies, and FUSER[1]. Then, we quantitatively verify that our proposed method can improve the SBRs' performance in terms of Hit Ratio, MRR and NDCG. In all four datasets, **FP-AdaMetric** outperforms the baselines. In addition, by investigating the visualization of the learned embedding space, we show that our proposed method better discriminate FP embedding from the session embedding than other baselines.

To sum up, our contributions are as follows:

- We highlight the importance of FPs in SBRs. We revisit the FP constraints in the optimization problem by transforming them into the metric-learning regularization.
- To better represent FPs, we propose an adaptive embedding modules in our metric-learning architecture.

Related Works

Neural Session-Based Recommendation

Session-based recommendation, which aims at both sequential and session-aware recommendation, learns patterns from consecutive item consumption logs in a certain period of time defined as a session (Fang et al. 2020; Wang et al. 2021a). GRU4REC and their variants (Hidasi et al. 2015; Hidasi and Karatzoglou 2018) introduce GRU (Chung et al. 2014) to learn sequential patterns in sessions. A recent work NARM (Li et al. 2017) introduces the attention method in GRU4REC to aggregate global and local user preferences. STAMP (Liu et al. 2018) uses attention and memory networks to capture better long-term preferences. With the improvement of GNN, SRGNN (Wu et al. 2019) and TAGNN (Yu et al. 2020) try to model the sessions as the graphs to improve performance. SASREC (Kang and McAuley 2018) and BERT4REC (Sun et al. 2019) apply the Transformer architecture (Vaswani et al. 2017) to effectively catch long-term preferences although they don't take the concept of sessions into account. Most of these SBRs models consider implicit click signals only as positive signals.

On the other hand, there are studies that involve the negative feedback to improve the sequential recommendation problem. DEERS (Zhao et al. 2018b,a) tries formulating the implicit feedback recommendation as a reinforcement learning problem that utilizes both positive and negative feedback. DFM (Xie et al. 2020) and DUMN (Bian et al. 2021) tackle Click Through Rate (CTR) problem based on explicit and implicit sequences. HAEM (Wang and Cao 2021) introduce next basket prediction (NBP) learning by intra- and inter-coupling of the baskets of clicked and uncliked items. However, to our knowledge, our work is the first trial to directly apply FPs (*e.g.*, skip) into the SBRs, especially in the next-item prediction problem.

Deep Metric Learning in Recommendation

Deep metric learning is a method for learning a projection function that effectively separates positive and negative samples in the desired metric space (Hoffer and Ailon 2015). In the field of recommendation systems, metric learning is widely studied for collaborate filtering (CF) of implicit feedback (Zhang et al. 2019). One of the most pronounced works, CMF (Hsieh et al. 2017) first introduces metric-learning approaches to learn user and item relations by Mahalanobis distance in the collaborate filtering. To improve geometric flexibility over Mahalanobis distance, LRML (Tay, Anh Tuan, and Hui 2018) introduces latent relational vectors to learn user and item embedding. To catch better item user relationships, NGCF (Wang et al. 2019) applies Graph Neural Network (GNN) to learn better user and item embedding. Several works (Wang et al. 2018; Tran et al. 2019) tries to improve sampling strategies of negative items for training efficiency.

However, only few studies have been conducted in engaging the metric learning paradigm in the session-based or sequential recommendation situations. SML (Twardowski, Zawistowski, and Zaborowski 2021) is the first attempt to apply a metric learning loss function in the SBRs. They transform the original loss, such as BPR and TOP1 (Hidasi et al. 2015; Hidasi and Karatzoglou 2018) into a metric learning-based one. One that is the most related to our work is XDM (Lv et al. 2020), which utilizes unclick behaviors (implicit negatives) in the sequential recommendation problem using asymmetric metric learning model with a confidence fusion layer that takes the unclick sequences as input. Our model differs in that it considers FP items in SBRs. We also provide theoretical insight into the metric-learning regularization and introduce the adaptive module to improve our regularization term.

[1]This is a community-based music creation game service.

Preliminaries

Problem Definition: Next Item Prediction at Each Time-Step in a Session

Let $U = \{u_1, ...u_{|U|}\}$ be the user set and $I = \{i_1, ...i_{|I|}\}$ be the item set where $|U|$ and $|I|$ is the number of users and items respectively. In SBRs, a click sequence, which is an item consumption sequence during a certain period of time (session), can be defined as $S_k^u = \{x_l\}_{l=1}^{|S_k^u|}$, where $x_l \in I$. Here, S_k^u is the k^{th} session sequence for user u and x_l is the l^{th} item clicked in the session. Since each user have multiple sessions, the sessions that belong to a single user u are defined as follows: $S^u = \{S_k^u\}_{k=1}^{|S^u|}$.

Simple Objective We now define the SBRs as a problem of the next-item prediction in the corresponding session. The objective of the SBRp at each time step is to find the most probable item which follows the true next-item distribution in the session. Following the recent works (Li et al. 2017; Wu et al. 2019; Song et al. 2019; Kang and McAuley 2018; Sun et al. 2019), we mathematically define the SBRp objective by negative log liklihood (NLL) loss (a.k.a Cross Entropy loss) as in (1).

$$\min_{\theta} \mathop{\mathbb{E}}_{u} \left[\mathop{\mathbb{E}}_{S^u} \left[\sum_{k=2}^{|S^u|} -log\left(p_\theta^u(x = x_k | x_1, \ldots, x_{k-1})\right) \right] \right] \quad (1)$$

, where $p_\theta^u\left(\cdot | x_1, \ldots x_{k-1}\right)$ is the target probability function parameterized by θ with a given item sequence $x_1, \ldots, x_{k-1}$ in S^u.

Considering the FPs Objective For a given user click sequence S_k^u, we sort out the true-positives and FPs using a predefined criterion (*e.g.*total consumption time or skip button click) as follows: $S_k^u = S_k^{u,p} \cup S_k^{u,fp}$. We also define an item click sequence of a single user u as $S^u = S^{u,p} \cup S^{u,fp}$, where $S^{u,p} = \{S_k^{u,p}\}_{k=1}^{|S^{u,p}|}$ and $S^{u,fp} = \{S_k^{u,fp}\}_{k=1}^{|S^{u,fp}|}$.

Since the goal of SBRp is to recommend only positive items to users, we reformulate our objective as follows.

$$\min_{\theta} \mathop{\mathbb{E}}_{u} \left[\mathop{\mathbb{E}}_{S^{u,p}} \left[\sum_{k=2}^{|S^{u,p}|} -log(p_\theta^u(x = x_k | x_1, \ldots x_{k-1}) \right] \right] \quad (2)$$

General Session-Based Recommendation Model

In this section, we formulate a general SBRs' model into an equation with trainable parameters of θ. A SBRs' model can be divided into 1) an embedding layer with θ_e, 2) a sequential layer with θ_{seq}, and 3) a recommendation layer (Jang et al. 2020; Lv et al. 2020). The trainable parameters are consist of the followings: $\theta = \{\theta_e, \theta_{seq}, \theta_{concat}(\text{optional})\}$, where $\theta_e = \left\{\theta_e^i, \theta_e^u\right\}$.

Embedding Layer In general, a unique integer id is given to every item and user (if exists). The embedding layer transforms each id into a d dimensional embedding vectors e, *i.e.*, $f_{\theta_e} : \mathbb{R} \rightarrow \mathbb{R}^d$. In this work, we use a lookup embedding matrix f_{θ_e}, following a commonly used technique in Natural Language Processing (NLP) domain. We define the item and user (if exists) embedding matrix as $f_{\theta_e^i}$ and $f_{\theta_e^u}$ respectively.

Sequence Embedding Layer The sequence embedding layer $f_{\theta_{seq}}(e_{x_1}, \cdots, e_{x_{k-1}}) : \mathbb{R}^{d \times k} \rightarrow \mathbb{R}^d$ maps a sequence of item embeddings into a single vector, where e_i is the item embedding, d is the dimension of the embedded vectors. The sequence embedding layer is a function that can be approximated with deep neural network architectures that are designed to capture sequential patterns, such as RNN (Chung et al. 2014; Hidasi et al. 2015), GNN (Wu et al. 2019; Qiu et al. 2020) Transformer (Kang and McAuley 2018; Sun et al. 2019) and so on. In the case that the users are also considered (Qiu et al. 2020), a concatenation layer $f_{\theta_c} : \mathbb{R}^{d \times 2} \rightarrow \mathbb{R}^d$ aggregates the output of sequence embedding layer with the user embeddings. The final output of sequential embedding e_{seq} is represented as follows:

$$e_{seq} = \begin{cases} f_{\theta_c}(f_{\theta_{seq}}(e_{x_1}, \cdots, e_{x_{k-1}}), e_u), & \text{if } u \text{ exists} \\ f_{\theta_{seq}}(e_{x_1}, \cdots, e_{x_{k-1}}), & \text{if } u \text{ not exists} \end{cases} \quad (3)$$

Recommendation Layer The purpose of a recommendation layer is to calculate the relevance score of each item r_i given sequential embedding e_{seq} as in (4). Among several options for the similarity measure D_{sim}, we choose dot product to calculate score of items due to its high performance and low complexity (Rendle et al. 2020). The recommendation score can be represented as follows:

$$r_i = D_{sim}(e_{seq}, e_i) = {e_{seq}}^T \cdot e_i \quad (4)$$

, where e_i is the embedding vector of item i and r_i is the final logit output.

To train this model using the equation (1), we define the probability $p_\theta^u(\cdot | x_1, \ldots, x_{k-1})$ as in equation (5), *i.e.*, a softmax layer.

$$p_\theta^u(i | x_1, \ldots x_{k-1}) = \frac{\exp\left(r_i\right)}{\sum_{x \in |I|} \exp\left(r_x\right)} \quad (5)$$

Methodology

Motivation: Direct Usage of FPs

Previous works (Li et al. 2017; Wu et al. 2019) consider FP feedback in a passive way, *i.e.*, only neglecting FPs. However, in the user scenarios of modern multimedia streaming services which the SBRs' models are mainly aimed at, users often give new items a try because they are not charged with any extra cost. Therefore, we argue that it is crucial to take FPs into account more actively to improve the quality of recommendation in such situations. To address this challenge, we introduce the FP-feedback constraint by using the common assumption 1 about FPs.

Assumption 1 (False-positive (FP) Characteristics)**.** *1) FPs are independent from the characteristics of the sequence. 2) FPs get lower relevance score than true-positives in general.*

The assumption 1-1) means that the FPs are consistent regardless of the characteristics of their corresponding sequences. Since FPs mean that the users do not prefer those items, FP property does not change dynamically with respect to other items in the sequence. The assumption 1-2) implies that FPs reveal more elaborate preference information of users by embracing their negative reaction towards certain items. Therefore, we need to set up a constraint where the embedding vectors of FPs are distant from the session embedding vector in terms of the given similarity measure. With these two assumptions, we formulate a constraint as shown in Definition 1:

Definition 1 (False-positive Constraint SBRp). *For the user set U and item set I, there exists a positive sequence set $S^{u,p} = \{S_k^{u,p}\}_{k=1}^{|S^p|}$ and a FP set $S^{u,fp} = \{S_k^{u,fp}\}_{k=1}^{|S^{u,fp}|}$. From the assumption 1, we can consider FP item sets: $FP^u = \{x_i | x_i \in S^{u,fp}\} \subset I$. Under these, the objective of SBRp can be shown as below:*

$$\min_{\theta} \mathbb{E}_{u} \left[\mathbb{E}_{S^{u,p}} \left[\sum_{k=2}^{|S^{u,p}|} -log(p_\theta^u(x_k|x_1, \dots x_{k-1}) \right] \right] \tag{6}$$

$$s.t.\ \mathbb{E}_{u} \left[\mathbb{E}_{FP^u, S^{u,p}} \left[\sum_{k=2}^{|S^{u,p}|} D_{sim}\big(f_{\theta_e^i}(fp), e_{seq}\big) \right] \right] \leq \epsilon \tag{7}$$

, where ϵ is a given margin and D_{sim} is a similarity measure (high is better), *e.g.*, dot product.

Theoretical Analysis

We now analyze the problem setting in Definition 1. First, we show that the additional constraint does not hurt the optimal value in (2).

Proposition 1 (Optimality Equivalence). *Let the FPs be $fp \in FP^u$, $fp \notin S^{u,p}$, and θ^* be the optimal parameter of* (2). *Then, there exists ϵ' such that θ^* also is an optimal value in the problem defined in Definition 1.*

Proof. Proof by contradiction. Detailed in Appendix. □

From Prop.1, we now conclude that the problem defined in 1 is applicable to SBRp as well. Also, the constraint restricts the search space of θ, making it easier to compute the optimal value in real world situations. After this Proposition, we need to change the form of Definition 1 properly in practice. Prop. 2 provides the evidence about it.

Proposition 2 (Metric-learning View). *If there exists an optimal Lagrange multiplier $\lambda > 0$, the problem which is defined in Definition 1 is identical to objective as in the equation 8.*

$$\min_{\theta} \mathbb{E}_{u} \left[\mathbb{E}_{S^{u,p}} \left[\sum_{k=2}^{|S^{u,p}|} -log(p_\theta^u(x_k|x_1, \dots x_{k-1}) \right] \right] \tag{8}$$

$$+ \lambda \left[\mathbb{E}_{u} \left[\mathbb{E}_{FP^u, S^{u,p}} \left[\sum_{k=2}^{|S^{u,p}|} L_{met}(x_k, fp, seq; \theta) \right] \right] \right] \tag{9}$$

, *where* $L_{met}(x_k, fp, seq; \theta) = \max\left(-D_{sim}(p, seq) + D_{sim}(fp, seq) + m, 0\right)$, *and* $D_{sim}(p, seq) = D_{sim}\big(f_{\theta_e^i}(x_k), e_{seq}\big)$ *and* $D_{sim}(fp, seq) = D_{sim}\big(\mathbb{E}_{fp \sim p(FP^u)}\left[f_{\theta_e^i}(fp)\right], e_{seq}\big)$.

Proof. Base on KKT conditions (Boyd, Boyd, and Vandenberghe 2004) in convex optimization. Detailed in Appendix. □

Consequently, we introduce metric-learning regularization terms in the original problem (equation (6)), which we call **FP-Metric**. Metric-learning (Hoffer and Ailon 2015; Kaya and Bilge 2019) is well-known approach to learn appropriate representation via FP and positive samples in computer vision (Karpusha, Yun, and Fehervari 2020; Venkataramanan et al. 2021) and audio (Chung et al. 2020; Xu et al. 2020) domains. Also, metric-learning is getting great attention recently due to high-performance in self-supervised and unsupervised approaches (Jaiswal et al. 2021). Unlike the previous works that conjugate metric-learning directly for training objectives, we use it for the regularization of the original loss highlighting the effect of the FP items. We leave the proofs on other types of metric-learning loss (Sohn 2016; Chen et al. 2020) for future works.

Proposed Method: FP-AdaMetric

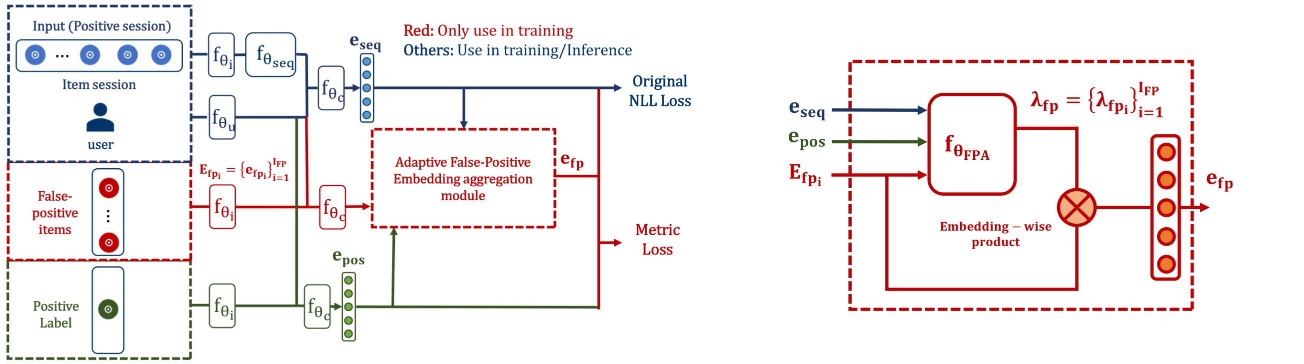
From the finding in the Proposition 2, we now propose the metric-learning regularization method to actively involve the FP items in the SBRs as shown in equation 8. We show its effectiveness in the experiment part.

Although this regularization shows effectiveness in both mathematical and experimental aspects, there are still limitations. Each of the FP items differs in the level of impact to the users. For example, when certain users have negative preference about scary movies, the level of scariness in each movie will decide how large of an impact the FP item would have. This property is summarized as in the remark 1.

Remark 1 (Degree of Dislike). *The level of impact that each FP item has on the sequence varies. We call this property the degree of dislike.*

This property is not considered in the equation 8 . Therefore, we propose an addtional module that computes the different level of impact each FP has in an adaptive way. We include this adaptive module in the metric learning loss function in the remark 2. The summary of our proposed method is shown in Figure 2. We call this **FP-AdaMetric**, a False-positive-aware Adaptive Metric Learning model.

Remark 2 (**FP-AdaMetric**). *Let us define the adaptive module for all FPs as follows: $f_{\theta_{FPA}}(e_u, e_{fp}, e_{seq}) : \mathbb{R}^{d \times 3} \rightarrow \mathbb{R}$, where $e_{fp} = f_{\theta_e^i}(fp)$. Base on this, we propose **FP-AdaMetric** which computes the FP part of the embed-*



(a) OverallFramework

(b) Adaptive False-Positive module details

Figure 2: Proposed framework that involves FP items in SBRs. (a) Overall framework for FP-AdaMetric, (b) Adaptive FP embedding aggregation module details.

ding in an adaptive fashion.

$$\min_{\theta} \mathbb{E}_{u} \left[\mathbb{E}_{S^{u,p}} \left[\sum_{k=2}^{|S^{u,p}|} -log(p_\theta^u(x_k|x_1, \dots x_{k-1}) \right] \right] \tag{10}$$

$$+ \lambda \left[\mathbb{E}_{u} \left[\mathbb{E}_{FP^u, S^{u,p}} \left[\sum_{k=2}^{|S^{u,p}|} L_{ada}(x_k, fp, seq; \theta) \right] \right] \right] \tag{11}$$

, *where the proposed adaptive metric-learning loss function is defined as* $L_{ada}(x_k, fp, seq; \theta) = \max\left(-D_{sim}(p, seq) + D_{sim}(fp, seq) + m, 0\right)$ *and the similarity measures are* $D_{sim}(p, seq) = D_{sim}\left(f_{\theta_e^i}(x_k), e_{seq}\right)$ *and* $D_{sim}(fp, seq) = D_{sim}\left(\mathbb{E}_{fp \sim p(FP^u)}\left[f_\theta(e_u, e_{fp}, e_{seq}) \times e_{fp}\right], e_{seq}\right)$.

Experiment Setting

Datasets

To verify our proposed method on the various data types, we evaluate the model on four different datasets: LastFM, Spotify, AmazonMovie and FUSER. The detailed statistics of each dataset are summarized in table 1. We split each dataset into 80% of train, 10% of validation, and 10% of test sets. The more detailed information is provided in the appendix.

Statistics	**LastFM**	**Spotify**	**Amazon Movie**	**FUSER**
Num. Users	0.5 K	None	None	8 K
Consumption cost	Low	Low	High	High
Num. Items	18 K	136 K	35 K	18 K
Num. Sess	84 K	125 K	8 K	59 K
False-positive (FP) decision	skip	skip	low score	skip
FP ratio (%)	10	50	30	20

Table 1: Statistics on the datasets used in the experiments

Baseline [2] and Metric

Model We choose the following recent SBRs models as baselines: **NARM** (Li et al. 2017), **SRGNN** (Wu et al. 2019), **STAMP** (Liu et al. 2018), and **GRU4REC** (Hidasi et al. 2015). **NARM** introduces attention layer to learn global and local preference of user preference in the sessions. **SRGNN** proposes a Graph Neural Network (GNN) architecture to capture complex item consumption in the sessions. **STAMP** utilizes the short-term attention and memory priority to capture users' general interests and current interests better. We specifically report the results from using **NARM** and **SRGNN** to show how our proposed method overcome their limitations. More experimental results (*e.g.* **STAMP**, **GRU4REC**) are reported in the appendix. Due to the low performance of the traditional non-deep learning methods (Jang et al. 2020; Qiu et al. 2020) - Sequence Popularity (S-POP) or First-order Markov Chain (FOMC)- are excluded in the baseline.

Method We set up two baseline methods as follows:

- **Vanilla**: FP items are not removed as in the equation (1). All click signals are regarded as implicit positive signals.
- **FP-Simple**: FP items are simply removed as in the equation (2). FPs are selected by skip (for LastFM, Spotify and FUSER) or scores below than 2 (for AmazonMovie).

Evaluation Metric We choose HR@K (Hit Ratio), MRR@K (Mean Reciprocal Rank) and NDCG@K (Normalized Discounted Cumulative Gain) for our evaluation metrics. These metrics are widely used in many of previous works (Jang et al. 2020; Lv et al. 2020). We also compare the results using different $K \in \{10, 50, 100\}$ to verify robustness in various scenarios.

[2] All implementations of the experiments are available in https://github.com/jongwonJeong/FPAdaMetric. It will be appeared after company's permission

Dataset	Base Model	Method	HR@K (%) 10	50	100	MRR@K (%) 10	50	100	NDCG@K (%) 10	50	100
LastFM	NARM	Vanilla	<u>37.69</u>	48.02	53.03	29.44	29.94	30.01	31.39	33.68	34.49
		FP-Simple	<u>37.69</u>	<u>48.07</u>	<u>53.04</u>	<u>29.48</u>	<u>29.97</u>	<u>30.04</u>	<u>31.41</u>	<u>33.70</u>	<u>34.51</u>
		FP-AdaMetric	**37.86**	**48.24**	**53.17**	**29.50**	**29.99**	**30.06**	**31.47**	**33.77**	**34.56**
	SRGNN	Vanilla	<u>38.21</u>	48.37	53.30	<u>30.06</u>	<u>30.54</u>	<u>30.61</u>	<u>31.98</u>	<u>34.23</u>	<u>35.03</u>
		FP-Simple	38.17	<u>48.44</u>	<u>53.35</u>	30.03	30.52	30.59	31.95	34.21	35.01
		FP-AdaMetric	**38.25**	**48.49**	**53.38**	**30.09**	**30.57**	**30.64**	**32.02**	**34.27**	**35.07**
Spotify	NARM	Vanilla	29.93	43.11	47.58	15.70	16.38	16.44	19.06	22.04	22.77
		FP-Simple	<u>82.09</u>	<u>83.80</u>	<u>84.45</u>	<u>79.54</u>	<u>79.63</u>	<u>79.64</u>	<u>80.16</u>	<u>80.55</u>	<u>80.65</u>
		FP-AdaMetric	**82.22**	**83.90**	**84.54**	**79.97**	**80.06**	**80.06**	**80.52**	**80.89**	**81.00**
	SRGNN	Vanilla	63.81	72.14	74.78	35.14	35.56	35.60	42.30	44.19	44.62
		FP-Simple	<u>82.25</u>	<u>84.14</u>	<u>84.83</u>	<u>79.27</u>	<u>79.36</u>	<u>79.37</u>	<u>80.00</u>	<u>80.45</u>	<u>80.54</u>
		FP-AdaMetric	**82.49**	**84.41**	**85.11**	**79.46**	**79.56**	**79.57**	**80.21**	**80.64**	**80.75**
Amazon Movie	NARM	Vanilla	7.42	13.08	16.91	4.57	4.82	4.87	5.24	6.46	7.08
		FP-Simple	<u>9.33</u>	<u>14.61</u>	<u>20.16</u>	<u>7.37</u>	<u>7.55</u>	<u>7.60</u>	<u>7.82</u>	<u>8.74</u>	<u>9.23</u>
		FP-AdaMetric	**9.38**	**14.85**	**20.51**	**7.51**	**7.69**	**7.73**	**7.94**	**8.83**	**9.32**
	SRGNN	Vanilla	7.35	12.36	15.89	5.05	5.27	5.32	5.58	6.67	7.24
		FP-Simple	**9.25**	<u>14.26</u>	<u>19.79</u>	**7.49**	**7.66**	**7.70**	**7.89**	**8.75**	<u>9.22</u>
		FP-AdaMetric	<u>9.24</u>	**14.44**	**20.24**	<u>7.47</u>	<u>7.64</u>	<u>7.69</u>	<u>7.88</u>	**8.75**	**9.23**
FUSER	NARM	Vanilla	4.16	14.19	22.04	1.40	1.82	1.93	2.04	4.18	5.44
		FP-Simple	<u>5.31</u>	<u>17.35</u>	<u>26.05</u>	<u>1.71</u>	<u>2.22</u>	<u>2.35</u>	<u>2.54</u>	<u>5.12</u>	<u>6.52</u>
		FP-AdaMetric	**5.48**	**17.81**	**26.75**	**1.80**	**2.33**	**2.45**	**2.64**	**5.28**	**6.72**
	SRGNN	Vanilla	3.43	12.05	19.67	1.09	1.45	1.55	1.63	3.46	4.68
		FP-Simple	<u>4.36</u>	<u>14.90</u>	<u>23.48</u>	<u>1.43</u>	<u>1.88</u>	<u>2.00</u>	<u>2.10</u>	<u>4.35</u>	<u>5.74</u>
		FP-AdaMetric	**4.54**	**15.49**	**24.19**	**1.49**	**1.95**	**2.07**	**2.18**	**4.51**	**5.92**

Table 2: Overall performance comparison on various datasets in terms of Hit Ratio, MRR, and NDCG@K where K$\in \{10, 50, 100\}$. The highest and second-highest scores are highlighted as bold and underline respectively.

Experimental Result

Our experiments are designed to verify the following research questions:

- **RQ1 (Performance)**: Can **FP-AdaMetric** improve the performance of SBRs by considering FPs?
- **RQ2 (Domain Difference)**: Are there differences in the effect of FPs across different data domains?
- **RQ3 (Ablation Study)**: What is the impact of the each module?
- **RQ4 (Embedding Analysis)**: Do users who are close to each other in the embedding space have similar preferences?

Dataset	Method	Metric@100 HR	MRR	NDCG
Spotify	FP-AdaMetric	84.54	**80.06**	**81.00**
	w/o Adaptive	**84.84**	79.95	80.97
	w/o Metric	84.45	79.64	80.65
	Vanilla	47.58	16.44	22.77
Amazon Movie	FP-AdaMetric	**20.51**	**7.73**	**9.32**
	w/o Adaptive	20.21	7.69	9.27
	w/o Metric	20.16	7.60	9.23
	Vanilla	16.91	4.87	7.08

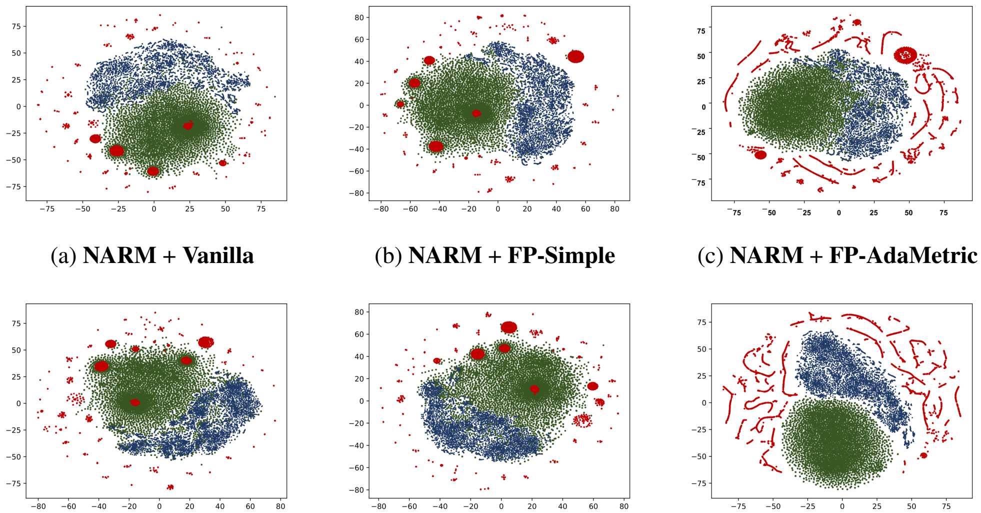
Table 3: Results on the ablation studies of FP-Metric and FP-Simple using NARM model.

Overall Performance

Overall results on all four benchmark datasets are presented in Table 2. Compared to **Vanilla** , **FP-Simple** shows higher HR, MRR, and NDCG by 22%, 101%, and 65% on the average. Especially, **FP-Simple** significantly improves the performance on the metrics regarding the ranking of true-positive items (*i.e.*, MRR and NDCG) than HR. From the results in general, we find that FP items in a session contaminate the true preference of items and eventually degrade the recommendation quality. It proves our claim that taking false positives into account is essential for SBRs. Also, we find a trend that the performance gain is larger on datasets with higher FP ratios, such as Spotify and Amazon Movie.

Next, from the comparison between all three methods, we verify that our proposed **FP-AdaMetric** outperforms all the other baselines in every evaluation metric across all data domains. This gives the answer to our **RQ1**.

When comparing the performance gain between **FP-Simple** and **FP-AdaMetric** in different datasets, the improvement in Amazon Movie and FUSER is larger than in LastFM and Spotify in general. We argue that this is because the item consumption cost in Movie and Game services is higher than in music streaming services, as shown in Table 1. Considering the FPs leads to a larger performance gain in such cases where users are more focused on the item contents and actively consume the items of their interests. This answers our **RQ2**.



(a) **NARM + Vanilla** (b) **NARM + FP-Simple** (c) **NARM + FP-AdaMetric**

(d) **SRGNN + Vanilla** (e) **SRGNN + FP-Simple** (f) **SRGNN + FP-AdaMetric**

Figure 3: Comparison of embedding spaces trained with Vanilla, FP-Simple and FP-AdaMetric (proposed method) using NARM and SRGNN on the Amazon Movie Test Dataset. Dimensions are reduced via PCA initilization and T-SNE. (blue: session-embeddings, green: true-positive embeddings, red: FP embeddings)

Ablation Study

We verify the effectiveness of each module in **FP-AdaMetric** in this section. **FP-AdaMetric** consists of 1) the metric learning module and 2) the adaptive module. To measure each module's effectiveness, we compare our original model **FP-AdaMetric** with a version without the metric learning module, **FP-Simple**, and a version without the adaptive module, **FP-Metric**. **FP-Simple** and **FP-Metric** are denoted as "w/o Adaptive" and "w/o Metric" respectively in Table 3. We report the performance of each setting on NARM in terms of HR, MRR, and NDCG@100 on Amazon Movie and Spotify. Table 3 shows that each module of our proposed method affects the performance differently. It can resolve **RQ3** as follows. The metric-learning module helps in every evaluation metric, especially in the Hit Ratio (HR). The constraint in the Definition 1 contributes to making FP embeddings far away from the sequential embedding, while not being concerned in attracting the positives. Therefore, it helps keeping the probability of FP in the top-k ranking low, resulting higher HR. On the other hand, the adaptive module (**FP-AdaMetric**) improves the performance in MRR and NDCG more than in HR. It means that the adaptive module helps to keep the rankings of true-positive items higher. We leave showing this relationship for future works.

Visualization Result

To qualitatively analyze the learned embedding spaces, we plot the embeddings of three different types (sequence, true-positive item, and FPs item) by T-SNE (Van Der Maaten 2014) with PCA initialization (Abdi and Williams 2010), which is a simple and high-quality visualization method widely used for the analysis of the embedding spaces (Kobak and Linderman 2021). Figure 3 shows the learned embedding space from three methods explained in table 2: **Vanilla** **FP-Simple** and **FP-AdaMetric** (proposed) on Amazon Movie dataset. Although the embeddings are not separated in **Vanilla** and **FP-Simple** method, we can see that **FP-AdaMetric** pushes away the FP embeddings (green) from true-positive and sequence embeddings. True-positive and sequence embeddings are placed very closely, and are not mixed with FPs. Therefore, the probability of recommending FPs in **FP-AdaMetric** is lower than in **Vanilla** or **FP-Simple** case. This answers **RQ4** that similar preference results in embeddings that are close to each other in our proposed method.

Conclusion

We study the impact of false positives on the SBRs, by carefully deriving the metric learning objective from the next-item prediction problem. We introduce the constraint optimization equation for SBRs and show that this equation can be transformed into a metric-learning regularization term. From the assumption that the degree of dislike in each false positive item differs, we propose the adaptive modules for false positives to improve our regularization effect. Diverse experiments show our proposed regularization including **FP-Metric** and **FP-AdaMetric** is effective in terms of several performance metrics, and our method plays a more crucial role in certain domains, such as Movie and Game.

Potential future research directions are as follows. Finding a proper criterion of false positives will contribute to a better results on methods that take false positives into account Wang et al. (2021b). Moreover, we can extend our theoretical analysis in N-pairs (Sohn 2016) or contrastive loss (Chen et al. 2020). Also, we can further explore the impact of false positives on other data domains (*e.g.*news).

Acknowledgments

We appreciate Hoyeop Lee from NCSOFT, Seongwon Jang from TMAP Mobility, and anonymous reviewers for the great discussion about our work.

References

Abdi, H.; and Williams, L. J. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4): 433–459.

Bian, Z.; Zhou, S.; Fu, H.; Yang, Q.; Sun, Z.; Tang, J.; Liu, G.; Liu, K.; and Li, X. 2021. Denoising User-aware Memory Network for Recommendation. *arXiv preprint arXiv:2107.05474*.

Boyd, S.; Boyd, S. P.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Chung, J. S.; Huh, J.; Mun, S.; Lee, M.; Heo, H. S.; Choe, S.; Ham, C.; Jung, S.; Lee, B.-J.; and Han, I. 2020. In defence of metric learning for speaker recognition. *arXiv preprint arXiv:2003.11982*.

Fang, H.; Zhang, D.; Shu, Y.; and Guo, G. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)*, 39(1): 1–42.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247*.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 173–182.

Hidasi, B.; and Karatzoglou, A. 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.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Hoffer, E.; and Ailon, N. 2015. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, 84–92. Springer.

Hsieh, C.-K.; Yang, L.; Cui, Y.; Lin, T.-Y.; Belongie, S.; and Estrin, D. 2017. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web*, 193–201.

Hu, Y.; Koren, Y.; and Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, 263–272. Ieee.

Jaiswal, A.; Babu, A. R.; Zadeh, M. Z.; Banerjee, D.; and Makedon, F. 2021. A survey on contrastive self-supervised learning. *Technologies*, 9(1): 2.

Jang, S.; Lee, H.; Cho, H.; and Chung, S. 2020. CITIES: Contextual Inference of Tail-item Embeddings for Sequential Recommendation. In *2020 IEEE International Conference on Data Mining (ICDM)*, 202–211. IEEE.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, 197–206. IEEE.

Karpusha, M.; Yun, S.; and Fehervari, I. 2020. Calibrated neighborhood aware confidence measure for deep metric learning. *arXiv preprint arXiv:2006.04935*.

Kaya, M.; and Bilge, H. Ş. 2019. Deep metric learning: A survey. *Symmetry*, 11(9): 1066.

Kobak, D.; and Linderman, G. C. 2021. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature biotechnology*, 39(2): 156–157.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.

Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1419–1428.

Liu, Q.; Zeng, Y.; Mokhosi, R.; and Zhang, H. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1831–1839.

Lv, F.; Li, M.; Guo, T.; Yu, C.; Sun, F.; Jin, T.; Wen, H.; Lin, G.; Yang, K.; and Ng, W. 2020. XDM: Improving Sequential Deep Matching with Unclicked User Behaviors for Recommender System. *arXiv preprint arXiv:2010.12837*.

Qiu, R.; Yin, H.; Huang, Z.; and Chen, T. 2020. Gag: Global attributed graph neural network for streaming session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 669–678.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Rendle, S.; Krichene, W.; Zhang, L.; and Anderson, J. 2020. Neural collaborative filtering vs. matrix factorization revisited. In *Fourteenth ACM Conference on Recommender Systems*, 240–248.

Sohn, K. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, 1857–1865.

Song, W.; Xiao, Z.; Wang, Y.; Charlin, L.; Zhang, M.; and Tang, J. 2019. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the Twelfth ACM international conference on web search and data mining*, 555–563.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 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.

Tay, Y.; Anh Tuan, L.; and Hui, S. C. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference*, 729–739.

Tran, V.-A.; Hennequin, R.; Royo-Letelier, J.; and Moussallam, M. 2019. Improving collaborative metric learning with efficient negative sampling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1201–1204.

Twardowski, B.; Zawistowski, P.; and Zaborowski, S. 2021. Metric Learning for Session-based Recommendations. *arXiv preprint arXiv:2101.02655*.

Van Der Maaten, L. 2014. Accelerating t-SNE using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1): 3221–3245.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Venkataramanan, S.; Psomas, B.; Avrithis, Y.; Kijak, E.; Amsaleg, L.; and Karantzalos, K. 2021. It Takes Two to Tango: Mixup for Deep Metric Learning. *arXiv preprint arXiv:2106.04990*.

Wang, Q.; Yin, H.; Hu, Z.; Lian, D.; Wang, H.; and Huang, Z. 2018. Neural memory streaming recommender networks with adversarial training. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2467–2475.

Wang, S.; Cao, L.; Wang, Y.; Sheng, Q. Z.; Orgun, M. A.; and Lian, D. 2021a. A survey on session-based recommender systems. *ACM Computing Surveys (CSUR)*, 54(7): 1–38.

Wang, W.; and Cao, L. 2021. Interactive sequential basket recommendation by learning basket couplings and positive/negative feedback. *ACM Transactions on Information Systems (TOIS)*, 39(3): 1–26.

Wang, W.; Feng, F.; He, X.; Nie, L.; and Chua, T.-S. 2021b. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 373–381.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 165–174.

Wen, H.; Yang, L.; and Estrin, D. 2019. Leveraging post-click feedback for content recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, 278–286.

Wu, C.; Wu, F.; Qi, T.; and Huang, Y. 2020. User Modeling with Click Preference and Reading Satisfaction for News Recommendation. In *IJCAI*, 3023–3029.

Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; and Tan, T. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 346–353.

Xie, R.; Ling, C.; Wang, Y.; Wang, R.; Xia, F.; and Lin, L. 2020. Deep Feedback Network for Recommendation. In *IJCAI*, 2519–2525.

Xu, J.; Wang, X.; Feng, B.; and Liu, W. 2020. Deep multi-metric learning for text-independent speaker verification. *Neurocomputing*, 410: 394–400.

Yi, X.; Hong, L.; Zhong, E.; Liu, N. N.; and Rajan, S. 2014. Beyond clicks: dwell time for personalization. In *Proceedings of the 8th ACM Conference on Recommender systems*, 113–120.

Yu, F.; Zhu, Y.; Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2020. TAGNN: Target attentive graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1921–1924.

Zhang, S.; Yao, L.; Sun, A.; and Tay, Y. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1): 1–38.

Zhao, X.; Xia, L.; Zhang, L.; Ding, Z.; Yin, D.; and Tang, J. 2018a. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*, 95–103.

Zhao, X.; Zhang, L.; Ding, Z.; Xia, L.; Tang, J.; and Yin, D. 2018b. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1040–1048.