Target Interest Distillation for Multi-Interest Recommendation

Chenyang Wang
DCST, BNRist, Tsinghua University
Beijing 100084, China
wangcy18@mails.tsinghua.edu.cn

Zhefan Wang
DCST, BNRist, Tsinghua University
Beijing 100084, China
wzf19@mails.tsinghua.edu.cn

Yankai Liu
China Mobile Research Institute &
THU-CMCC Joint Institute
Beijing 100084, China
liuyankai@chinamobile.com

Yang Ge
China Mobile Research Institute
Beijing 100084, China
geyang100299@163.com

Weizhi Ma
AIR, Tsinghua University
Beijing 100084, China
mawz@tsinghua.edu.cn

Min Zhang*
DCST, BNRist, Tsinghua University &
THU-CMCC Joint Institute
Beijing 100084, China
z-m@tsinghua.edu.cn

Yiqun Liu
DCST, BNRist, Tsinghua University
Beijing 100084, China
yiqunliu@tsinghua.edu.cn

Junlan Feng
China Mobile Research Institute
Beijing 100084, China
fengjunlan@chinamobile.com

Chao Deng
China Mobile Research Institute
Beijing 100084, China
dengchao@chinamobile.com

Shaoping Ma
DCST, BNRist, Tsinghua University
Beijing 100084, China
msp@tsinghua.edu.cn

ABSTRACT

Sequential recommendation aims at predicting the next item that the user may be interested in given the historical interaction sequence. Typical neural models derive a single history embedding to represent the user's interests. Moving one step forward, recent studies point out that multiple sequence embeddings can help to better capture multi-faceted user interests. However, when ranking candidate items, these methods usually adopt the *greedy inference strategy*. This approach uses the best matching interest for each candidate item to calculate the ranking score, neglecting the target interest distribution in different contexts, which might lead to incompatibility with the current user intent. In this paper, we propose to enhance multi-interest recommendation by predicting the target user interest with a separate interest predictor and a specifically designed distillation loss. The proposed framework consists of two modules: the 1) *multi-interest extractor* to generate multiple embeddings regarding different user interests; and the 2) *target-interest predictor* to predict the interest distribution in the current context, which will be further utilized to dynamically aggregate multi-interest embeddings. To provide explicit supervision signals to the target-interest predictor, we devise a target-interest distillation loss that uses the similarity between the target item and multi-interest embeddings as the soft label of the target interest. This helps the target-interest predictor to accurately predict the user interest at the inference stage and enhances its generalization ability. Extensive experiments on three real-world datasets show the effectiveness and flexibility of the proposed framework.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557464

CCS CONCEPTS

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

KEYWORDS

recommender systems, sequential behaviors, multi-interest recommendation, knowledge distillation

ACM Reference Format:
Chenyang Wang, Zhefan Wang, Yankai Liu, Yang Ge, Weizhi Ma, Min Zhang, Yiqun Liu, Junlan Feng, Chao Deng, Shaoping Ma. 2022. Target Interest Distillation for Multi-Interest Recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557464

1 INTRODUCTION

Recommender systems play a crucial role in many online services, such as e-commerce, advertising, and so on. Traditional recommendation methods are mainly based on collaborative filtering [14, 26], which assumes that similar users will have similar preferences. Many recent works formalize recommendation as a sequential prediction task [2, 22, 32], aiming to predict the next item given the

*Corresponding author.

This work is supported by the Natural Science Foundation of China (Grant No. U21B2026, 62002191) and Tsinghua University Guoqiang Research Institute.

historical interaction sequence. Most advanced models adopt neural networks (e.g., recurrent neural network [10], self-attention [13]) to represent a user's history as an embedding vector, which is subsequently utilized to predict the next item.

Recently, researchers point out that it might not be adequate to use a single embedding to represent diverse interests of a user in different aspects [2, 17]. For example, in the emerging recommendation scenario on smart TV, different members in a family share the same TV but prefer different types of programs. A user in this scenario corresponds to a family instead of a single person. Besides, a user in e-commerce may also have multi-faceted preferences (e.g., jewelry, handbags, and cosmetics) depending on the context. Hence, a proposal of representing a user with multiple embeddings has been experimented to capture diverse user interests and proved effective. Many techniques (e.g., dynamic routing [17], self-attention [2]) have been explored to generate multi-interest embeddings from a user's interaction history. At the inference stage, each embedding is able to retrieve a set of candidate items independently. To generate the final recommendation, the common practice is to choose the overall Top-K items with the maximal matching scores, which is referred as *greedy inference strategy* [2, 3].

However, we argue that this greedy inference strategy fails to take full advantage of multi-interest user embeddings, which is equivalent to only considering the best matching interest for each candidate item. As a result, this approach inclines to retrieve items that match either of the user's interests, disregarding the target interest distribution in different contexts. In this paper, we use *interest distribution* to represent the user's dynamic preference over different interests. For example, we find it is common that an interaction sequence on a smart TV consists of many children programs and a few movies. Although children programs are likely to have higher matching scores most of the time, there might be some patterns that indicate a stronger intention for movies. In e-commerce, after a user bought a cellphone, it should be factored in that this user is less likely to be interested in purchasing another cellphone within a short period of time. Hence, we are motivated to model the target interest distribution in different contexts, which helps to capture dynamic user intent and hence make more accurate recommendations at the inference stage.

In this paper, we propose a **T**arget-**i**nterest distillation framework for **M**ulti-**i**nterest **Rec**ommendation, called TiMiRec. Specifically, TiMiRec consists of two modules: the 1) multi-interest extractor generates multiple interest embeddings from the user's interaction sequence, while the 2) target-interest predictor estimates the interest distribution in the current context. The basic idea of TiMiRec is to distill the knowledge of predicting the target interest distribution to a separate module, so that multi-interest embeddings can be dynamically aggregated by the predicted interest distribution. Notice that there is generally no labelling data for the actual user interest, which makes it hard to provide appropriate supervision signals to the target-interest predictor. To solve this problem, we use the similarity between the target item and each multi-interest embedding as the soft label of the target interest during training. A knowledge distillation loss is devised to match the predicted interest distribution and the soft label. This will facilitate the generalization ability of the target-interest predictor to infer the target interest distribution in different contexts. We conduct extensive experiments on three datasets, including two public data and one industrial dataset. Our proposed framework is flexible to work with various multi-interest recommendation methods and obtains significantly superior performances compared to state-of-the-art methods. The main contributions of this paper can be summarized as follows:

- We propose that it is sub-optimal to adopt the greedy inference strategy in multi-interest recommendation. The target interest distribution provides additional supervision signals during training but is usually ignored.
- A simple but effective framework, TiMiRec, is devised to adaptively aggregate multiple interests based on the target user interest. A separate target-interest predictor together with a knowledge distillation loss are introduced to infer the target interest distribution in different contexts.
- We conduct extensive experiments on three real-world data sets, showing that the proposed method achieves significant performance improvements compared to state-of-the-art multi-interest recommendation methods.

2 RELATED WORK

2.1 Sequential Recommendation

Different from general recommendation methods [9, 34], sequential recommendation leverages users' historical sequences to better capture dynamic user intent, which attracts increasing attention in recent years. Traditional sequential methods depend on Markov chains to model the transition pattern between items [24, 27]. Recently, with the development of deep learning, a lot of works utilize different neural networks to encode the historical sequence to a hidden vector [10, 16, 31, 38, 39]. GRU4Rec [10] first introduces Recurrent Neural Network (RNN) to the sequential recommendation domain and achieves impressive performance improvements compared to traditional methods. Caser [28] and NextItNet [39] utilizes Convolution Neural Network (CNN) based methods to capture high-order Markov chains by applying convolutional operations on historical sequences. Besides, inspired by the effectiveness of attention mechanism in other domains [1, 37], SASRec [13] first applies self-attention to model the mutual influence between historical interactions, achieving remarkable performance gains. TiSASRec [19] further introduce time intervals into the calculation of self-attention. Despite the great success of deep-learning sequential recommendation models, most of them focus on the structure of the sequence encoder and only give an overall embedding from the user's interaction sequence, which is not enough to cultivate multi-faceted user interests.

2.2 Multi-Interest Recommendation

Recent studies begin to focus on diverse user interests [2, 17, 41] to better understand user intent in practice. Some early studies use extra side information to capture dynamic user intents [29]. Inspired by the structure of capsule network [25], MIND [17] uses capsules to represent multiple user interests based on the dynamic routing mechanism. ComiRec [2] devises a multi-interest extraction layer to derive multiple embeddings from the user interaction sequence. The extraction method can be based on either dynamic capsule routing or attention mechanism. Some follow-up studies [22] also adopt multi-interest extraction to cultivate different user interests.

Different from a single sequence embedding in traditional methods, each interest vector here can retrieve a set of items based on the matching score. To generate the final candidate items, the common practice is to choose the items with the overall maximal matching scores across interests [2, 3]. Some studies [17] use attention mechanism to aggregate multi-interest embeddings, but the aggregation weight is solely instructed by the recommendation loss. None of them considers the additional supervision signal from the target interest distribution addressed in our TiMiRec framework.

2.3 Knowledge Distillation

Knowledge distillation [11] is first proposed for model compression and acceleration, which aims to learn a small student model from a large teacher model. The distillation loss is usually defined to match the output logits between the teacher model and the student model [6]. However, the application scope of knowledge distillation is not restricted to model compression. This technique has been shown to be effective in many tasks such as self-distillation [40] and learning from noise labels [20]. It is also beneficial to leverage this idea to enhance the model effectiveness and robustness in recommendation. For example, some privileged features are only available in the offline setting but absent for online serving. Recent work [36] tries to distill the knowledge brought by privileged features to another same-structure network with only regular features, which is proven to be effective in practice. In the multi-interest recommendation scenario, the target interest distribution is also informative but only available during training. This inspires us to utilize the knowledge distillation technique to leverage the soft label of the target interest as an additional supervision signal, which helps to accurately predict the user interest at the inference stage.

3 METHODOLOGY

3.1 Preliminaries

3.1.1 Problem Formulation. Let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item set, respectively. For each user $u \in \mathcal{U}$, we are given a chronologically ordered list $[i_{u,1}, i_{u,2}, \cdots, i_{u,N_u}]$, where each element $i_{u,t} \in \mathcal{I}$ is the interacted item at time step t and N_u is the length of the interaction sequence. Then the task of sequential recommendation is: given the historical sequence before the target time step t, denoted as $S_{u,t}$, generating an ordered list of items that the user u may be interested in.

3.1.2 Multi-Interest Recommendation. Different from typical sequential recommendation that gives an overall embedding from a user's behavior sequence, multi-interest recommendation methods produce K interest embeddings based on the user historical interaction sequence $S_{u,t}$:

$$\mathbf{V}_{u,t} = [\mathbf{v}_{u,t}^1, \cdots, \mathbf{v}_{u,t}^K] \in \mathbb{R}^{D \times K}, \tag{1}$$

where D is the dimension of the embedding space. Given a candidate item $i_{u,t}$, the interest embedding with the maximal matching score is utilized as the user representation [2]:

$$\mathbf{v}_{u,t} = \mathbf{V}_{u,t}[:, \operatorname{argmax}\left(\mathbf{V}_{u,t}^T \mathbf{i}_{u,t}\right)], \tag{2}$$

where $\mathbf{i}_{u,t}$ is the embedding of the candidate item. Then the ranking score can be calculated as the dot product between the user representation and target item embedding (i.e., $f(u, i_{u,t}) = \mathbf{v}_{u,t}^T \mathbf{i}_{u,t}$). This encourages different sequence embeddings to represent different aspects of interests, cause only the most relevant interest embedding will be updated each time. At the inference stage, the *greedy inference strategy* is usually adopted that derives the ranking score in a similar way with training. The best matching interest embedding will be used to calculate the ranking score.

3.2 TiMiRec Framework Overview

Figure 1 shows the overall structure of the proposed framework. There are two major modules in our TiMiRec: the 1) multi-interest extractor $F_{\Phi_E}(\cdot)$ parameterized by Φ_E and the 2) target-interest predictor $G_{\Phi_P}(\cdot)$ parameterized by Φ_P. Previous multi-interest recommendation methods mainly focus on how to derive multiple sequence embeddings to represent diverse user interests (i.e., structure of multi-interest extractor), and then use the most compatible interest when retrieving candidate items. Differently, we introduce a target-interest predictor to infer the interest distribution according to the current context:

$$\mathbf{z}_{u,t}^q = G_{\Phi_P}(S_{u,t}) \in \mathbb{R}^K, \tag{3}$$

which is utilized to aggregate multi-interest embeddings as the user representation:

$$\mathbf{v}_{u,t} = \mathbf{V}_{u,t}\ \operatorname{softmax}(\mathbf{z}_{u,t}^q). \tag{4}$$

This predicted interest distribution $\mathbf{z}_{u,t}^q$ is supposed to be proportional to the probability of activating each interest in the current context. Considering that user intents are latent and generally hard to predict, we argue that it is not adequate to learn the target-interest predictor only with the common next-item recommendation loss. To solve this problem, we propose a target-interest distillation loss to facilitate the generalization capability of the target-interest predictor. We use the similarity between the target item and multi-interest embeddings as a soft label of the target interest, and the predicted interest distribution $\mathbf{z}_{u,t}^q$ is encouraged to be close to the target interest distribution (denoted as $\mathbf{z}_{u,t}^p$). This additional supervision signal helps the target-interest predictor accurately predict the target interest in different contexts.

It is noteworthy that there are no requirements to the concrete structure of the backbone modules, which leads to a flexible framework that can improve various multi-interest recommendation models. Next, we will first introduce the target-interest distillation loss and the training procedure of TiMiRec in Section 3.3 and 3.4, and then give an example instantiation for each module in Section 3.5.

3.3 Target-Interest Distillation Loss

Although our TiMiRec uses a separate target-interest predictor to generate the interest distribution in different contexts, there is no explicit supervision on the predicted interest distribution. If it is only used to aggregate multi-interest embeddings and optimized by the final next-item recommendation loss $\mathcal{L}_{rec}$, the model might find some shortcuts to mainly update multi-interest embeddings rather than the target-interest predictor. This motivates us to find other supervision signals to assure the rationality of the predicted interest distribution.

Note that given the well-learned multi-interest extractor and the target item (only available during training), we can measure the

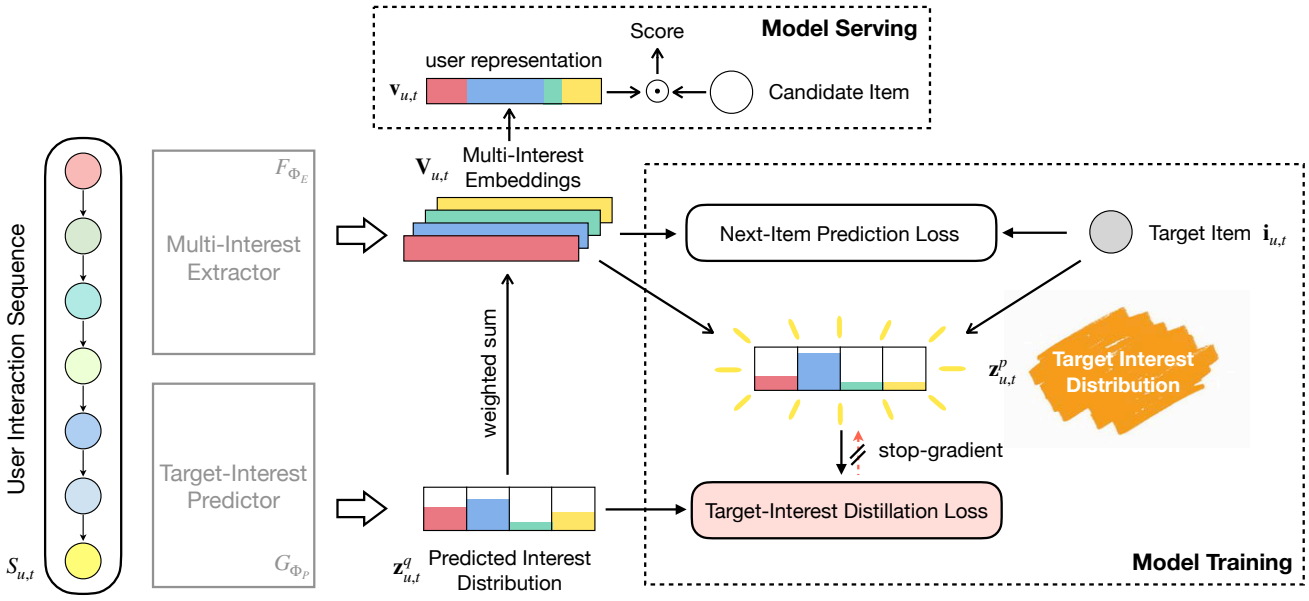

Figure 1: Overview of the proposed TiMiRec framework. TiMiRec mainly consists of two modules: 1) multi-interest extractor and 2) target-interest predictor. The former derives multiple interest embeddings from a user's interaction sequence, while the latter gives the predicted interest distribution in the current context. Then we use the predicted interest distribution to aggregate multi-interest embeddings and calculate the next-item prediction loss. Besides, a target-interest distillation loss is devised to instruct the target-interest predictor, where the soft label of the target interest is derived by the compatibility (cosine similarity) between the target item and multi-interest embeddings, serving as an additional supervision signal.

target interest distribution by the similarity between multi-interest embeddings $\mathbf{V}_{u,t} = [\mathbf{v}^1_{u,t}, \cdots, \mathbf{v}^K_{u,t}]$ and the target item $\mathbf{i}_{u,t}$:

$$\mathbf{z}^p_{u,t} = \text{sim}(\mathbf{V}_{u,t}, \mathbf{i}_{u,t}) = \left[\frac{\mathbf{v}^1_{u,t}}{||\mathbf{v}^1_{u,t}||_2} \frac{\mathbf{i}_{u,t}}{||\mathbf{i}_{u,t}||_2}, \cdots, \frac{\mathbf{v}^K_{u,t}}{||\mathbf{v}^K_{u,t}||_2} \frac{\mathbf{i}_{u,t}}{||\mathbf{i}_{u,t}||_2} \right]. \tag{5}$$

Here we use the cosine similarity to measure the compatibility between each interest and the target item. This target interest distribution $\mathbf{z}^p_{u,t} \in \mathbb{R}^K$ reflects how the actual interacted item is related to different interests of the user, which is suitable to be taken as an additional supervision signal (soft label) to instruct the target-interest predictor in our TiMiRec. As a result, we propose a target-interest distillation loss inspired by the knowledge distillation technique [6], which encourages the predicted interest distribution (output of the student model) to be close to the target interest distribution (output of the teacher model).

Remember that the predicted interest distribution is denoted as $\mathbf{z}^q_{u,t}$. First we derive the normalized interest distribution for the predicted one and the target one:

$$\mathbf{q}_{u,t} = \frac{\exp(\mathbf{z}^q_{u,t} \,/\, T)}{\sum_{k=1}^K \exp(\mathbf{z}^q_{u,t}[k] \,/\, T)}, \tag{6}$$

$$\mathbf{p}_{u,t} = \frac{\exp(\mathbf{z}^p_{u,t} \,/\, T)}{\sum_{k=1}^K \exp(\mathbf{z}^p_{u,t}[k] \,/\, T)}. \tag{7}$$

Here T is a hyperparameter called distillation temperature, which controls the smoothness of the normalized distribution. Then the target-interest distillation loss can be directly defined as:

$$\mathcal{L}'_{\text{distill}} = -\sum_{u \in \mathcal{U}} \sum_{t=2}^{N_u} \mathbf{p}^T_{u,t} \log(\mathbf{q}_{u,t}). \tag{8}$$

This is a standard distillation loss [6] that intuitively pushes up the similarity between the predicted and the target interest distribution. The distillation loss makes use of multi-interest embeddings to measure the target interest distribution, which provides additional supervision signals to the target-interest predictor and endows the generalization ability to infer the interest distribution in different contexts.

Meanwhile, notice that the target interest distribution in our scenario is not a fixed target but derived by the similarity between multi-interest embeddings and the target item embedding. Directly optimizing the above distillation loss will also make the target interest distribution be updated to approach the predicted one, which might harm the well-learned multi-interest extractor. As a result, we add a stop-gradient operator to the normalized target interest distribution $\mathbf{p}_{u,t}$ as follows:

$$\mathcal{L}_{\text{distill}} = -\sum_{u \in \mathcal{U}} \sum_{t=2}^{N_u} \text{stopgrad}(\mathbf{p}^T_{u,t}) \log(\mathbf{q}_{u,t}). \tag{9}$$

In this way, the gradients of this loss will only affect the target-interest predictor as expected. Otherwise the rationale of the learned multi-interest embedding is likely to be influenced. We will show the influence of the stop-gradient operation in Section 4.4.

Algorithm 1 Learning algorithm of TiMiRec

Input: multi-interest extractor structure F_{Φ_E}, target-interest predictor structure G_{Φ_P}, interest number K
Output: parameters Φ_E, Φ_P

while not converged **do**
  $\mathbf{V}_{u,t} = F_{\Phi_E}(S_{u,t})$.
  $\mathbf{v}_{u,t} = \mathbf{V}_{u,t}[:, \text{argmax}\left(\mathbf{V}_{u,t}^T \mathbf{i}_{u,t}\right)]$.
  Pretrain multi-interest extractor with $\mathcal{L}_{\text{rec}}$.
end while
while not converged **do**
  $\mathbf{V}_{u,t} = F_{\Phi_E}(S_{u,t})$.
  $\mathbf{z}_{u,t}^q = G_{\Phi_P}(S_{u,t})$.
  $\mathbf{z}_{u,t}^p = \text{sim}(\mathbf{V}_{u,t}, \mathbf{i}_{u,t})$, i.e., Eq.(5).
  Calculate target-interest distillation loss $\mathcal{L}_{\text{distill}}$.
  $\mathbf{v}_{u,t} = \mathbf{V}_{u,t}\text{softmax}(\mathbf{z}_{u,t}^q)$.
  Calculate next-item recommendation loss $\mathcal{L}_{\text{rec}}$.
  Finetune Φ_E and Φ_P with $\mathcal{L}$, i.e., Eq.(11).
end while
return Φ_E, Φ_P

3.4 Learning Algorithm

Considering that the rationale of the target interest distribution relies on meaningful multi-interest embeddings, we adopt a two-stage learning strategy to facilitate the training process. Specifically, we first disregard the target-interest predictor and pretrain the multi-interest extractor in a similar fashion with previous multi-interest recommendation models [2]. The best matching interest embedding is taken as the user representation $\mathbf{v}_{u,t}$ (i.e., Eq.(2)) to accomplish the next-item prediction task. Following the common practice, the objective is defined as a pairwise ranking loss [23]:

$$\mathcal{L}_{\text{rec}} = -\sum_{u \in \mathcal{U}} \sum_{t} \log \sigma\left(\mathbf{v}_{u,t}^T \mathbf{i}_{u,t} - \mathbf{v}_{u,t}^T \mathbf{i}_{u,t}^-\right), \tag{10}$$

where $i_{u,t}^-$ is a negative item randomly sampled from items the user has not interacted with, and $\sigma(\cdot)$ is the sigmoid function. This pretraining stage makes sure the multi-interest extractor be able to generate meaningful interest embeddings according to the user interaction sequence.

At the secend stage, we turn to use the interest distribution generated by the target-interest predictor to derive the user representation (i.e., Eq.(4)). The multi-interest extractor and target-interest predictor will be finetuned by jointly optimizing the next-item prediction loss and the proposed target-interest distillation loss:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + T^2 \mathcal{L}_{\text{distill}}. \tag{11}$$

The coefficient T^2 of the target-interest distillation loss is to balance the two supervision signals. Studies in the literature of knowledge distillation [6] show that the gradient of the temperature-scaled distillation loss will be scaled by $1/T^2$. This coefficient makes the target-interest distillation loss adaptively compatible with the next-item prediction loss. The detailed learning algorithm is shown in Algorithm 1. We will compare different learning strategies in Section 4.4.

3.5 Module Instantiation

Note that TiMiRec is model-agnostic and there is no specific restricts to the concrete structure of each module. In this section, we give an example instantiation for each module in TiMiRec. The principle here is to keep simple and effective. We leave the investigation of more complex implementations as future work to center our contribution in the overall learning framework.

3.5.1 Multi-Interest Extractor. The multi-interest extractor is responsible for generating K interest embeddings $\mathbf{V}_{u,t} \in \mathbb{R}^{D \times K}$ based on the user interaction sequence $S_{u,t}$. Here we use a self-attentive method [21] as an example instantiation, which is also adopted in previous work about multi-interest recommendation [2].

The input item sequence $S_{u,t}$ with length n is first transformed into embeddings $\mathbf{H} \in \mathbb{R}^{D \times n}$ through an embedding layer. To make use of the order information, we add trainable positional embeddings [30] to the input item embeddings. Then we can get the multi-interest attention matrix as

$$\mathbf{A} = \text{softmax}\left(\mathbf{W}_2^T \tanh(\mathbf{W}_1 \mathbf{H})\right)^T, \tag{12}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are trainable parameters with size $D_a \times D$ and $D_a \times K$. The final multi-interest embeddings $\mathbf{V}_{u,t}$ is computed by

$$\mathbf{V}_{u,t} = \mathbf{H}\mathbf{A}. \tag{13}$$

This is like aggregating the historical item sequence in K different ways by changing the standard D_a-dimensional $\mathbf{W}_1$ to a parameter matrix with size $D_a \times K$, which is parameter efficient and shown to be effective [2].

3.5.2 Target-Interest Predictor. For fair comparisons with other multi-interest recommendation methods, we consider the same history behavior sequence $S_{u,i}$ as input to generate the predicted interest distribution $\mathbf{z}_{u,t}^q$. Specifically, we first encodes the history sequence to a summary embedding $\mathbf{s}_{u,t} \in \mathbb{R}^D$ with Gated Recurrent Unit (GRU) [4] as the sequence encoder:

$$\mathbf{s}_{u,t} = \text{GRU}(\mathbf{H}'), \tag{14}$$

where $\mathbf{H}'$ is transformed from $S_{u,t}$ with another embedding layer. We use the last hidden state of GRU as the final representation $\mathbf{s}_{u,t}$. Then the predicted interest distribution $\mathbf{z}_{u,t}^q \in \mathbb{R}^K$ can be derived by a 2-layer projection MLP head:

$$\mathbf{z}_{u,t}^q = \mathbf{W}_2^q \cdot \text{ReLU}\left(\mathbf{W}_1^q \cdot \mathbf{s}_{u,t} + \mathbf{b}_1\right) + \mathbf{b}_2, \tag{15}$$

where $\mathbf{W}_1^q \in \mathbb{R}^{D \times D}, \mathbf{W}_2^q \in \mathbb{R}^{K \times D}, \mathbf{b}_1 \in \mathbb{R}^D, \mathbf{b}_2 \in \mathbb{R}^K$ are trainable parameters. Despite that many other methods can be adopted to get the sequence embedding and derive the interest distribution, we empirically find a simple GRU with a 2-layer MLP head yields promising results most of the time. Besides, the structure of the target-interest predictor is also not restricted to ID-based sequential models. Content-based models such as DeepFM [7] can also serve as the target-interest predictor if other context information is available (e.g., time of the day, user profiles, item attributes). We leave other implementations with more context information as future work to center our contributions in the overall learning framework.

Table 1: Statistics of datasets.

Dataset	#user ($\|\mathcal{U}\|$)	#item ($\|\mathcal{I}\|$)	#inter ($\sum_u N_u$)	density
Beauty	22,363	12,101	198,502	0.07%
MovieLens	6,040	3,706	1,000,209	4.47%
CMCC	49,847	29,074	1,300,351	0.09%

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 Datasets. We conduct experiments on three real-world datasets, including two public benchmarks and one industrial data.

- **Beauty**[1]: This is one of the series of product review datasets crawled from Amazon [8].
- **MovieLens**[2]: This is a widely used recommendation dataset containing user's ratings for movies. We choose the 1M version and treat the ratings as implicit feedback.
- **CMCC**: This is an industrial dataset collected from China Mobile, containing video watching activities on smart TV. We randomly sampled 50,000 users and record their watching activities in June 2021. Then users and items with less than 5 associated interactions are discarded.

The statistics of datasets are summarized in Table 1.

4.1.2 Baselines. We compare TiMiRec with several representative models. The baselines are classified into two sets according to whether they include transformer layers [30] to encode the history sequence, because we find there are significant performance gaps between these two kinds of models. The methods without transformer layers include:

- **GRU4Rec** [10]: This is the first sequential recommendation algorithm that utilizes recurrent neural network (RNN) to model historical interactions.
- **YouTube** [5]: This is a popular sequential model for industrial recommendations that uses MLP to process the history.
- **MIND** [17]: This is a novel multi-interest recommendation method that is capable of extracting multiple interest vectors based on the capsule routing mechanism.
- **ComiRec** [2]: This is a state-of-the-art multi-interest recommendation method. We use the ComiRec-SA version based on attention mechanism because of its stable performance.

The methods with transformer layers include:

- **SASRec** [13]: This method utilizes self-attention to exploit the mutual influence between historical interactions.
- **TiSASRec** [19]: This method improves SASRec [13] by considering time intervals between historical interactions.
- **ComiRec+**: This is an enhanced version of ComiRec that first passes the historical item embeddings $\mathbf{H}$ to a transformer layer to get contextual item embeddings in the sequence. Then the same attentive method as ComiRec is utilized to derive multi-interest embeddings.

4.1.3 Evaluation Protocols. We adopt the leave-one-out strategy to evaluate model performance, which is widely used in previous work [19, 33, 35]. For each interaction sequence, we use the most recent interaction for testing, the second recent interaction for validation, and the remaining interactions for training. As for the candidate items, previous studies usually sample 100 negative items that the user has not interacted with and rank the ground-truth item together with these items. However, recent work [15] has demonstrated that sampled metrics may lead to inconsistent results when the number of negative items is small. To strike a balance between non-sampling evaluation and computational efficiency, we randomly sample 1000 items as negative items, and this setting is shown to be close to the non-sampling version [18].

We employ Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) [12] as evaluation metrics (abbreviated by H@K and N@K respectively). HR@K measures whether the target item appears in the Top-K recommendation list, while NDCG@K further concerns about its position in the ranking list. We abbreviate HR@K and NDCG@K to H@K and N@K in the following for convenience. Each experiment is repeated 5 times with different random seeds and we report the average score.

4.1.4 Implementation Details. We use Adam as the optimizer and search for hyper-parameters on the validation set. For fair comparisons, the batch size is set to 256, the embedding size is set to 64, and the maximum history length is set to 20 for all the methods. The learning rate is tuned between $[1e^{-3}, 5e^{-4}, 1e^{-4}]$; the weight decay is tuned between $[1e^{-4}, 1e^{-6}, 1e^{-8}, 0]$. For multi-interest recommendation methods, we tune the interest number K between $[2, 4, 6, 8]$. For TiMiRec, the temperature T is tuned between $[0.01, 0.1, 0.2, 0.5, 1, 2, 5, 10]$. We tune other baseline-specific hyper-parameters within the range suggested by their authors. The experiments are supported by JIUTIAN Artificial Intelligence Platform[3]. The codes are publicly available for reproducibility[4].

4.2 Overall Performance

For our proposed TiMiRec, we devise two versions to compare with the two kinds of baselines:

- **TiMiRec**: This is the standard version that uses the attentive method described in Section 3.5.1, which is equivalent to adopt ComiRec as the multi-interest extractor.
- **TiMiRec+**: This enhanced version adopts ComiRec+ as the multi-interest extractor, which includes a transformer layer in multi-interest extraction.

Table 2 summarizes the performance of different methods.

First, we can observe that multi-interest recommendation methods (e.g., MIND, ComiRec) perform better than traditional sequential models that only give an overall embedding for each user (e.g., GRU4Rec, YouTube) on Beauty and CMCC. This shows the importance to consider multi-faceted user interests in sequential recommendation. Besides, the proposed framework leads to consistently better results compared to the base multi-interest method (TiMiRec vs. ComiRec, TiMiRec+ vs. ComiRec+). TiMiRec and TiMiRec+ achieve the best performance within the corresponding model set,

[1] https://jmcauley.ucsd.edu/data/amazon/links.html
[2] https://grouplens.org/datasets/movielens/
[3] https://jiutian.10086.cn/
[4] https://github.com/THUwangcy/ReChorus/tree/CIKM22

Table 2: Top-K recommendation performance on the three datasets. TiMiRec and TiMiRec+ adopt ComiRec and ComiRec+ as the multi-interest extractor, respectively. The best results within the same set of methods are in bold face, and the overall best results are underlined. The superscripts * and ** indicate $p \leq 0.05$ and $p \leq 0.01$ for the paired t-test of TiMiRec/TiMiRec+ vs. the best baseline within the corresponding model set.

Setting		Models without Transformer Layer					Models with Transformer Layer			
Dataset	Metric	GRU4Rec	YouTube	MIND	ComiRec	TiMiRec	SASRec	TiSASRec	ComiRec+	TiMiRec+
Beauty	H@5	0.1072	0.1040	0.1193	0.1257	**0.1437****	0.1435	0.1529	0.1546	**0.1573***
	H@10	0.1552	0.1563	0.1727	0.1832	**0.2006****	0.2058	0.2084	0.2123	**0.2196***
	H@20	0.2107	0.2264	0.2492	0.2543	**0.2645****	0.2706	0.2760	0.2809	**0.2887****
	N@5	0.0719	0.0702	0.0809	0.0852	**0.1006****	0.1004	0.1087	0.1095	**0.1112***
	N@10	0.0873	0.0870	0.0981	0.1038	**0.1118****	0.1192	0.1266	0.1272	**0.1313***
	N@20	0.1013	0.1046	0.1173	0.1217	**0.1350****	0.1356	0.1436	0.1459	**0.1488***
MovieLens	H@5	0.2730	0.2336	0.1863	0.2513	**0.3091****	0.3124	0.3212	0.2745	**0.3333****
	H@10	0.3964	0.3406	0.2881	0.3659	**0.4310****	0.4407	0.4397	0.3906	**0.4556****
	H@20	0.5323	0.4719	0.4152	0.4937	**0.5625****	0.5674	0.5712	0.5091	**0.5843****
	N@5	0.1875	0.1597	0.1229	0.1708	**0.2136****	0.2177	0.2241	0.1875	**0.2346****
	N@10	0.2273	0.1942	0.1558	0.2078	**0.2529****	0.2593	0.2625	0.2249	**0.2741****
	N@20	0.2616	0.2274	0.1877	0.2400	**0.2861****	0.2910	0.2956	0.2549	**0.3067****
CMCC	H@5	0.3978	0.4170	0.4229	0.4547	**0.4812****	0.4681	0.4768	0.4831	**0.4886***
	H@10	0.5121	0.5328	0.5381	0.5716	**0.5934****	0.5828	0.5882	0.5960	**0.6020****
	H@20	0.6306	0.6453	0.6533	0.6845	**0.7018****	0.6853	0.6937	0.6997	**0.7091****
	N@5	0.2916	0.3064	0.3119	0.3356	**0.3636****	0.3533	0.3615	0.3662	**0.3690***
	N@10	0.3286	0.3438	0.3492	0.3735	**0.3999****	0.3905	0.3975	0.4027	**0.4057***
	N@20	0.3587	0.3723	0.3784	0.4020	**0.4273****	0.4164	0.4242	0.4290	**0.4329***

respectively. It is noteworthy that TiMiRec even performs better than SASRec on CMCC in the absence of transformer layers, and TiMiRec+ further outperforms state-of-the-art sequential recommendation methods. This shows the importance of taking target interest distribution into consideration. To measure how accurate is the target-interest predictor in TiMiRec, we additionally calculate the Jensen–Shannon divergence (lower is better) between the predicted interest distribution $\mathbf{q}_{u,t}$ and the target one $\mathbf{p}_{u,t}$ on the test set. The averaged result on Beauty and MovieLens is 0.0032 and 0.0005, respectively. This shows that TiMiRec is able to accurately predict the target interest, leading to better recommendations.

Second, we notice that previous multi-interest recommendation methods give poor performances on MovieLens, which may depend on the multi-interest characteristics in different datasets. If there is no obvious multi-faceted user interests in the dataset, the introduction of multiple interest embeddings and the greedy inference strategy might hurt the recommendation performance. Differently, it is noteworthy that TiMiRec still achieves the best performance on MovieLens and leads to substantial improvements compared to ComiRec. The target-interest predictor and the distillation loss help the model dynamically aggregate different interests and make it adaptive to various application scenarios.

4.3 Effect of Target-Interest Distillation

In this section, we conduct additional experiments to validate our motivation and better understand the rationale of the target-interest distillation in TiMiRec. Previous multi-interest recommendation methods mainly focus on how to extract multiple interest embeddings, while the greedy inference strategy may lead to incompatibility between the recommendation result and the actual user intent. The key insight of target-interest distillation in our TiMiRec is to dynamically aggregate multi-interest embeddings according to the current context. Here we compare different aggregation methods of multi-interest embeddings to show the superiority of target-interest distillation. Specifically, three pooling methods are compared here:

- **max pooling**: This is equivalent to the greedy inference strategy in previous works, which uses the best matching interest to calculate ranking scores.
- **mean pooling**: This variant directly uses the mean operation to aggregate multi-interest embeddings, which means all the interests are treated equally.
- **attn pooling**: This method uses the history embedding $\mathbf{s}_{u,t}$ as the query vector to aggregate multi-interest embeddings via attention mechanism, which can be taken as TiMiRec without the distillation loss (only $\mathcal{L}_{\text{rec}}$ is optimized).

Table 3 shows the performance of different methods on Beauty and MovieLens. We can see that max pooling and mean pooling yield similar results, while attention pooling leads to significantly better performance. This shows that it is crucial to consider the target interest distribution in different contexts, rather than just focusing on the best matching interest or treating each interest equally. Meanwhile, TiMiRec achieves further improvements compared to attention pooling. The reason is that user intent is generally

Table 3: Comparison with different aggregation methods of multi-interest embeddings. TiMiRec is significantly better than other aggregation methods with $p \leq 0.05$.

Aggregation Method	Beauty		MovieLens	
	H@10	N@10	H@10	N@10
max pooling (ComiRec)	0.1832	0.1038	0.3659	0.2078
mean pooling	0.1738	0.0975	0.3623	0.2086
attn pooling (w/o distill)	0.1791	0.1043	0.4258	0.2503
TiMiRec	**0.2006**	**0.1118**	**0.4310**	**0.2529**

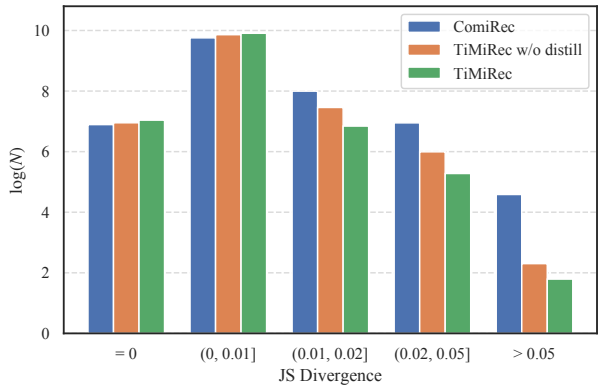

Figure 2: Distribution of Jensen–Shannon divergence between interest distributions of the target item and the top-1 recommended item for different methods on the test set.

hard to predict. It might be not adequate to only rely on the final recommendation loss. With the help of the proposed target-interest distillation loss, the target-interest predictor in TiMiRec is able to predict the interest distribution more accurately.

Further, we calculate the Jensen–Shannon divergence (lower is better) between interest distributions of the target item and the top-1 recommended item for different methods on the test set. Figure 2 gives the distribution of JS divergence on the Beauty dataset. Results show that the recommended top-1 item of TiMiRec is more compatible to the actual target item (i.e., higher for =0, $(0, 0.01]$), and there are obviously less incompatible cases (i.e., lower for $(0.01, 0.02]$, $(0.02, 0.05]$, >0.05). In particular, we find ComiRec leads to far more cases where the recommended item has a very different interest distribution from the target one (i.e., >0.05), which results from the greedy inference strategy to a large extent. A separate target-interest predictor (w/o distill) partially helps alleviate this issue, but it is inferior to TiMiRec without the help of the distillation loss. The proposed target-interest distillation loss leverages the target interest distribution as an additional supervision signal to instruct the predictor. Although the target interest distribution is only available during training, results show that this supervision signal enhances the generalization ability of the target-interest predictor, leading to less incompatibility between the target and the recommended item.

Table 4: Performance of TiMiRec variants.

Method	Beauty		MovieLens	
	H@10	N@10	H@10	N@10
joint train	0.1787	0.1030	0.4267	0.2483
w/o stopgrad	0.1971	**0.1157**	0.4260	0.2468
TiMiRec	**0.2006**	0.1118	**0.4310**	**0.2529**

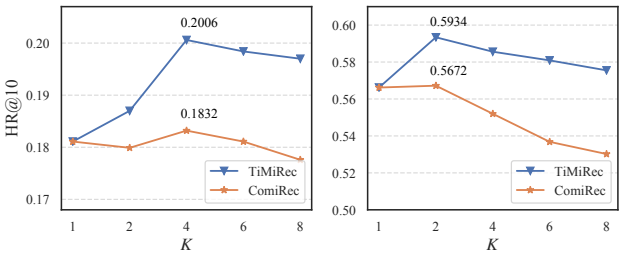

Figure 3: Parameter sensitivity analysis.

4.4 Further Analysis

To validate the effectiveness of other designs in our TiMiRec, here we compare with another two variants:

- **joint train**: This variant discards the pretraining process and jointly optimizes the final objective $\mathcal{L}$ from scratch.
- **w/o stopgrad**: This variant removes the stop-gradient operation in the target-interest distillation loss.

Table 4 shows the performance of these variants.

Firstly, joint training leads to much worse results on Beauty, but achieves a promising performance on MovieLens. Remember that ComiRec is effective on Beauty but performs badly on MovieLens. Hence, the pretraining might be more useful on datasets with obvious multi-interest characteristics, where the multi-interest extractor can be trained to get reasonable interest representations. This benefits the learning process by providing more accurate target interest distribution at the beginning of the finetuning stage.

Secondly, w/o stopgrad results in a little performance drop. Although the differences are not that obvious, we find w/o stopgrad will hurt the diversity of interest representations because multi-interest embeddings are updated to approach the predicted interest distribution. We calculate the averaged L2-distance between interest embeddings for each user on Beauty, the distance decreases from 4.08 (TiMiRec) to 0.69 (w/o stopgrad), which is not desirable for further usages of interest embeddings (e.g., show representative items of different interests to the user in consideration of the explainability). Differently, TiMiRec achieves better performances and maintains the difference between interests as expected.

4.5 Parameter Sensitivity

Figure 3 shows HR@10 of ComiRec and TiMiRec when changing the interest number K from 1 to 8 on Beauty and CMCC datasets. First, we can see that TiMiRec is consistently better than ComiRec under

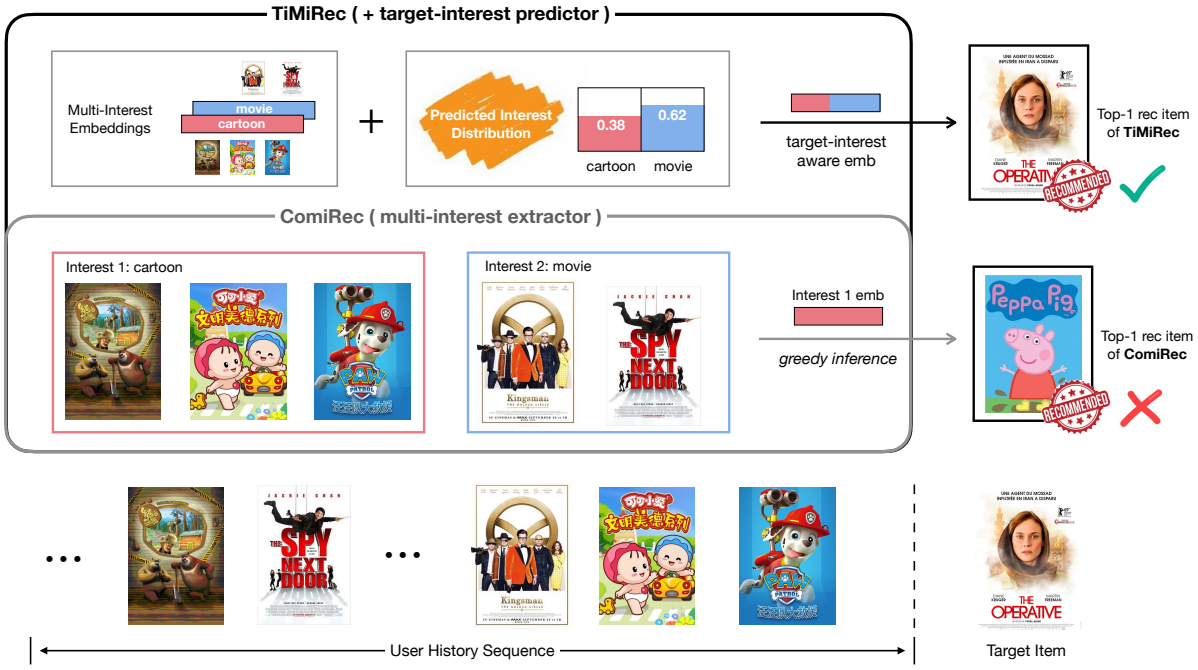

Figure 4: A case study on CMCC dataset. The multi-interest extractor generates two interests from the user history sequence: 1) cartoon and 2) movie. When making recommendations, ComiRec only considers the best matching interest for each candidate item, and hence wrongly recommends a cartoon because it gets the maximal matching score. However, after a series of cartoon watching, the interest for parents to watch movies may take advantage. TiMiRec can capture such dynamic intent with the target-interest predictor and gives the exactly correct recommendation.

different settings[5] of K, while the best choice may be determined by the dataset. Users in the Beauty dataset may have more diverse interests (e.g., jewelry, handbags, and cosmetics), and the best result is achieved when $K = 4$. In the scenario of video recommendation in CMCC, user interests are more concentrated and $K = 2$ leads to promising results. This is in general reasonable that each family has two aspects of interests (e.g., interests for parents and children).

4.6 Case Study

Figure 4 gives a case study of the top-1 recommendation results of ComiRec and our TiMiRec on the CMCC dataset when $K = 2$. Given the historical interaction sequence of this user, the multi-interest extractor generates two interests in terms of 1) cartoon and 2) movie. These two interests are probably corresponding to the children and parents in this family. It is noteworthy that only ID information is utilized in our model. Despite that, items in the history sequence are divided into reasonable groups, which validates the effectiveness of the multi-interest extractor.

When making recommendations, ComiRec only considers the maximal matching score of each candidate item towards different user interests. As a result, a cartoon similar with the recently watching ones is ranked the highest. However, after watching a series of cartoons continuously, the interest for parents to watch movies may take advantage. Due to the target-interest predictor and the distillation loss, the proposed TiMiRec successfully captures such dynamic intent. After aggregating matching scores w.r.t. multiple interests according to the predicted interest distribution, the movie (target item) gets a higher ranking score and our TiMiRec gives the exactly correct recommendation.

[5]When $K = 1$, TiMiRec is equivalent to ComiRec because there is only one interest.

5 CONCLUSION

In this paper, we propose an effective and flexible framework to make use of the target interest distribution for multi-interest recommendation, called TiMiRec. The proposed framework uses a separate target-interest predictor to infer the interest distribution according to the target context. This distribution will be utilized to adapatively aggregate different user interests. Considering that user intents are latent and hard to predict, a target-interest distillation loss is proposed to leverage the target interest distribution as an additional supervision signal. The target interest distribution (measured by the similarity between each interest embedding and the target item) is only available during training but is shown to be able to help the predictor make more accurate predictions in different contexts, which is usually neglected in previous studies. Experimental results show that our framework achieves significant performance improvements and is flexible to work with various multi-interest recommendation models. In the future, we plan to take further investigations on how to introduce temporal information into the target-interest predictor more effectively to make better interest predictions.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2942–2951.

[3] Gaode Chen, Xinghua Zhang, Yanyan Zhao, Cong Xue, and Ji Xiang. 2021. Exploring Periodicity and Interactivity in Multi-Interest Framework for Sequential Recommendation. *arXiv preprint arXiv:2106.04415* (2021).

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[5] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[6] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129, 6 (2021), 1789–1819.

[7] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[8] 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.

[9] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

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

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

[12] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

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

[14] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[15] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1748–1757.

[16] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 95–104.

[17] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2615–2623.

[18] Dong Li, Ruoming Jin, Jing Gao, and Zhi Liu. 2020. On sampling top-k recommendation evaluation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2114–2124.

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

[20] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. 2017. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*. 1910–1918.

[21] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130* (2017).

[22] Yujie Lu, Shengyu Zhang, Yingxuan Huang, Luyao Wang, Xinyao Yu, Zhou Zhao, and Fei Wu. 2021. Future-Aware Diverse Trends Framework for Recommendation. In *Proceedings of the Web Conference 2021*. 2992–3001.

[23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

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

[25] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829* (2017).

[26] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web*. Springer, 291–324.

[27] Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, Sep (2005), 1265–1295.

[28] 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.

[29] Md Mehrab Tanjim, Congzhe Su, Ethan Benjamin, Diane Hu, Liangjie Hong, and Julian McAuley. 2020. Attentive sequential models of latent intent for next item recommendation. In *Proceedings of The Web Conference 2020*. 2528–2534.

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

[31] Chenyang Wang, Weizhi Ma, and Chong Chen. 2022. Sequential Recommendation with Multiple Contrast Signals. *ACM Transactions on Information Systems (TOIS)* (2022).

[32] Chenyang Wang, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2020. Toward Dynamic User Intention: Temporal Evolutionary Effects of Item Relations in Sequential Recommendation. *ACM Transactions on Information Systems (TOIS)* 39, 2 (2020), 1–33.

[33] Chenyang Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2020. Make It a Chrous: Knowledge- and Time-aware Item Modeling for Sequential Recommendation. In *Proceedings of the 43th International ACM SIGIR conference*. ACM.

[34] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. *arXiv preprint arXiv:2206.12811* (2022).

[35] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2019. Modeling Item-Specific Temporal Dynamics of Repeat Consumption for Recommender Systems. In *The World Wide Web Conference*. ACM, 1977–1987.

[36] Chen Xu, Quan Li, Junfeng Ge, Jinyang Gao, Xiaoyong Yang, Changhua Pei, Fei Sun, Jian Wu, Hanxiao Sun, and Wenwu Ou. 2020. Privileged features distillation at Taobao recommendations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2590–2598.

[37] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. 2048–2057.

[38] Lu Yu, Chuxu Zhang, Shangsong Liang, and Xiangliang Zhang. 2019. Multi-order attentive ranking model for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5709–5716.

[39] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 582–590.

[40] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3713–3722.

[41] 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.