CMCLRec: Cross-modal Contrastive Learning for User Cold-start Sequential Recommendation

Xiaolong Xu
njuxlxu@gmail.com
Nanjing University of Information Science and Technology
Nanjing, Jiangsu, China

Hongsheng Dong
202312210008@nuist.edu.cn
Nanjing University of Information Science and Technology
Nanjing, Jiangsu, China

Lianyong Qi*
lianyongqi@gmail.com
China University of Petroleum (East China)
Qingdao, Shandong, China

Xuyun Zhang*
xuyun.zhang@mq.edu.au
Macquarie University
Sydney, New South Wales, Australia

Haolong Xiang
hlx6700@gmail.com
Nanjing University of Information Science and Technology
Nanjing, Jiangsu, China

Xiaoyu Xia
xiaoyu.xia@rmit.edu.au
RMIT University
Melbourne, Victoria, Australia

Yanwei Xu
xuyanwei@tju.edu.cn
Tianjin University
Tianjin, China

Wanchun Dou
douwc@nju.edu
Nanjing University
Nanjing, Jiangsu, China

ABSTRACT

Sequential recommendation models generate embeddings for items through the analysis of historical user-item interactions and utilize the acquired embeddings to predict user preferences. Despite being effective in revealing personalized preferences for users, these models heavily rely on user-item interactions. However, due to the lack of interaction information, new users face challenges when utilizing sequential recommendation models for predictions, which is recognized as the cold-start problem. Recent studies, while addressing this problem within specific structures, often neglect the compatibility with existing sequential recommendation models, making seamless integration into existing models unfeasible. To address this challenge, we propose CMCLRec, a Cross-Modal Contrastive Learning framework for user cold-start RECommendation. This approach aims to solve the user cold-start problem by customizing inputs for cold-start users that align with the requirements of sequential recommendation models in a cross-modal manner. Specifically, CMCLRec adopts cross-modal contrastive learning to construct a mapping from user features to user-item interactions based on warm user data. It then generates a simulated behavior sequence for each cold-start user in turn for recommendation purposes. In this way, CMCLRec is theoretically compatible with any extant sequential recommendation model. Comprehensive experiments conducted on real-world datasets substantiate that, compared with state-of-the-art baseline models, CMCLRec markedly enhances the performance of conventional sequential recommendation models, particularly for cold-start users.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Sequential Recommendation, Cold-start, Cross-modal Contrastive Learning, Self-supervised Learning

ACM Reference Format:
Xiaolong Xu, Hongsheng Dong, Lianyong Qi, Xuyun Zhang, Haolong Xiang, Xiaoyu Xia, Yanwei Xu, and Wanchun Dou. 2024. CMCLRec: Cross-modal Contrastive Learning for User Cold-start Sequential Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3626772.3657839

1 INTRODUCTION

Recommender systems have been extensively applied across diverse online and mobile platforms, including but not limited to e-commerce, music streaming, and social media platforms [37]. In such platforms, user behavior evolves over time [2], and the number of items that typical users interact with usually represents only 1%-2% of the total items. This results in a highly sparse user-item interaction matrix, posing significant limitations on traditional recommendation algorithms, such as collaborative filtering [19] and two-tower models [36].

Recently, the rapid development of deep learning [4] has engendered substantial investigation into embedding-based sequential recommender systems [6]. These systems have been widely adopted

*Corresponding author.

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657839

in the industry, benefiting from their proficiency in accurately capturing the dynamic behaviors of users and providing high-quality recommendations. Sequential recommender systems typically capture sequential information from user-item interaction sequences to forecast the next potential item with which the user is predisposed to interact. However, despite the diversity of models, sequential recommender systems exhibit a strong dependency on user-item interaction sequences due to fixed recommendation patterns. This leads to suboptimal performance when recommending to new users, a challenge commonly known as the user cold-start problem.

To tackle this problem, various methods have been investigated, such as the content-based cold-start recommendation [30] and Dropout [24]. For content-based cold-start recommendation, it recommends items by analyzing item content information, as well as user personal characteristics and preferences. In this case, it can better understand user interests, especially in scenarios involving new users or a lack of user historical behavioral data. Dropout is another popular mechanism. During training, it randomly drops some neurons and user feature information to reduce the model's reliance on historical interaction data, which helps enhance the model's generalization ability and effectively improves its cold-start performance. This approach encourages cold-start recommendations based on alternative content information, mitigating the impact of suboptimal ID embeddings. Additionally, there are models based on meta-learning [25], active learning [39], and other methods to address the cold-start problem.

However, the above-mentioned methods face a common challenge in that they struggle to be integrated into these sequential recommendation models where specialized structural requirements for sequential recommendation models are needed to address specific data distributions [33]. This problem stems from substantial differences in the model structures, feature representations, and parameter settings of various cold-start algorithms compared to established sequential recommendation models. Besides, several approaches concentrate extensively on enhancing cold-start performance, frequently overlooking recommendations for regular users. Although state-of-the-art sequential recommendation models, such as SASRec [11] and MCLRec [16], exhibit suboptimal performance in recommending to cold-start users, they perform well with warm users. If a cold-start module could be seamlessly integrated, the performance is expected to show a significant improvement. To be compatible with existing sequential recommendation models, recommending to cold-start users requires generating simulated behavior sequences as model inputs. However, due to the substantial distribution gap between user features and behavior sequences, an effective alignment technique is urgently needed to tackle the mapping issue.

Recently, cross-modal contrastive learning [38], capable of generating mappings between multiple modalities, has garnered widespread attention in various fields, such as computer vision [42], natural language processing [35], etc. This approach reinforces semantic relationships by learning shared embedding representations of diverse modal data and facilitating closer proximity of similar content in this space. This makes it possible to generate simulated behavioral sequences.

Furthermore, the embedded modules should not impose additional training burdens on the overall model and should not demand an excessive amount of extra training data. For instance, Chen et al. [3] address the cold-start problem using generative adversarial networks to reduce the difference between cold and warm item embeddings. Lee et al. [12] employ meta-learning to estimate cold-start user preference. However, such methods are constrained by the need for additional training data, increasing the costs associated with data collection and processing accordingly. Interestingly, in [34], self-supervised learning is introduced into the recommender system. It directly masks Wikipedia for training data and utilizes contrastive learning techniques, without the need for extra preparation.

Considering the preceding discourse, we introduce a Cross-Modal Contrastive Learning framework for sequential RECommendation (CMCLRec) to address the user cold-start problem in this paper. The fundamental concept underlying CMCLRec is to utilize the cross-modal contrastive learning method to establish a mapping between user features and user-item interactions. This mapping is employed to generate simulated behavior sequences for cold-start users, and advanced sequential recommendation models subsequently utilize the generated sequences to provide recommendations for cold-start users. The model is composed of three main modules: the data augmentation module, the cross-modal contrastive learning module, and the sequential recommendation module. The data augmentation module employs contrastive learning to enhance user features and user-item interaction sequences, encouraging the model to bring embeddings of similar users closer in the embedding space to extract more enriched hidden features. The cross-modal contrastive learning module utilizes auto-encoder techniques to map user features and user-item interaction sequences into the same embedding space. It learns the mapping from user features to user-item interaction sequences based on warm user data, enabling the construction of simulated behavior sequences for cold-start users. The sequential recommendation module is agnostic to the underlying model and can be instantiated using any embedding-based sequential recommendation model. Our framework preserves the advanced overall recommendation performance by leveraging the advantages of cutting-edge sequential recommendation models while augmenting their effectiveness in cold-start scenarios. In addition, the initial two modules adopt self-supervised learning, eliminating the need for additional labeled data preparation.

The main contributions of our work are as follows:

- We design a novel framework, named Cross-Modal Contrastive Learning Recommendation (CMCLRec), to mitigate the user cold-start problem in recommender systems. CMCLRec generates simulated behavior sequences based on user features for cold-start users, facilitating their incorporation into any sequential recommendation model to enhance the performance in cold-start scenarios.
- We employ self-supervised training in CMCLRec to enhance the recommendation performance of the model for cold-start users without requiring supplementary label data.
- We conduct experiments based on two publicly available datasets. The results illustrate that CMCLRec outperforms the most competitive model across all scenarios, and the ablation study further confirms the effectiveness of the cross-modal construction of simulated behavior sequences.

2 RELATED WORK

2.1 Sequential Recommendation

Sequential recommendation, as investigated in previous studies [6, 28], utilizes user-item interactions to formulate embeddings for users. It forecasts the subsequent item that is most likely to be interacted with by the user. This paradigm has undergone extensive scrutiny and practical application in both academic and industrial contexts. Traditional Sequential recommendation adopts Markov chain models [8, 17], which have significant advantages in modeling user-item interaction in a sequence. However, Markov properties can only capture short-term and point dependencies, making them less suitable for real-world scenarios. Amidst the rapid evolution of deep learning, neural networks have progressively been integrated into recommender systems to address the limitations of traditional algorithms. GRU4Rec [9] employs Recurrent Neural Networks (RNN) to predict the next possible interaction by capturing the sequential relationships within a given user-item interaction sequence, introducing positional information of items into the model. Caser [20], inspired by the computer vision field, utilizes Convolutional Neural Networks (CNN) with a focus on short-term behavioral preferences that have a more significant impact on users. SR-GNN [32], rooted in Graph Neural Networks, conceptualizes interactions as nodes within a graph, subsequently mapping each sequence to paths in the graph. Ultimately, it acquires embeddings for users or items within the graph. The advent of the Transformer architecture has elevated the self-attention mechanism to a mainstream approach in recommender systems. SASRec [11] introduces the Transformer into sequential recommendation, employing the self-attention mechanism to model user-item interaction sequences and extract more valuable features.

2.2 Cold-start Recommendation

Despite the substantial success achieved by embedding-based recommendation models in the realm of recommender systems, they encounter challenges in delivering accurate recommendations for cold-start users devoid of user-item interaction sequences. This limitation contributes to a notable decline in user retention rates. Efficiently utilizing side information such as attribute features, knowledge graphs, and auxiliary domains becomes a common solution for cold-start scenarios without user-item interaction sequences. DropoutNet [24] introduces a dropout mechanism during training, significantly reducing the model's dependence on ID embedding and enhancing the weights of other content features. This approach allows cold-start users to be recommended mainly based on other content features, mitigating the impact of poor ID embeddings. MetaEmbedding [14] leverages item features, excluding ID, and incorporates a generator network to produce the initialization values for ID embeddings. In the case of cold-start items, the generator forecasts their initial ID embeddings, and subsequent training and recommendations are executed based on these embeddings. MWUF [40] generates scaling and shifting functions from item features using meta-learning, which are employed to transform features for cold-start items, mapping them to another feature space to enhance prediction accuracy. In the context of the cold-start scenario with limited user-item interaction sequences, efficiently utilizing the scarce data is a crucial challenge. Vartak et al. [23] addressed cold-start items on Twitter by training a classifier on items that users have interacted with and then using this classifier to determine whether a user is interested in a cold-start item. MetaTL [27] adopts a sequential recommendation model, utilizing few-shot learning to recommend to cold-start users, and leveraging meta-learning to enhance the accuracy of recommendations. MML[15] integrates side information of items into the meta-learning process to improve the recommendation effect of cold-start items. MeLU (Meta-Learned User Preference) [12] employs the Model-Agnostic Meta-Learning (MAML) algorithm for the purpose of meta-learning a shared set of initialization parameters. For each cold-start user, MeLU fine-tunes the initialized model using the limited user-item interaction data to obtain a user-customized model for recommending items to cold-start users.

Recently, contrastive learning [29] has been widely applied, achieving unprecedented success and providing an alternative approach to addressing the cold-start problem. CLCRec [31] maximizes the dependence between item content and collaborative signals based on a contrastive learning objective function, enabling the model to retain interactive information in the content representation of cold-start items. CPKSPA [13] introduces an effective combination of a rating prediction module, embedding distribution alignment module, and contrastive augmentation module to reduce differences between potential embedding distributions across domains. This results in more stable and robust embeddings for cold-start items. Socially-aware dual contrastive learning [5] introduces an approach that integrates user-user relationships, user-item interactions, and item-item similarity to adapt representations within a semi-supervised environment. Cold-start users leverage their social relationships for modeling warm users without necessitating additional user-item interaction records.

However, the aforementioned studies face challenges in seamless integration with existing efficient recommendation models, thereby significantly compromising their flexibility. Moreover, these studies have not fully harnessed the implicit relationships between user-item interactions and user features.

3 METHOD

3.1 Overview

In this section, we introduce the CMCLRec framework to enhance conventional sequential recommendation algorithms to achieve improved accuracy in recommending items for cold-start users. CMCLRec consists of three modules, including the data augmentation module, cross-module contrastive learning module, and sequential recommendation module.

3.2 Problem Formulation

This study aims to generate simulated behavioral sequences for cold-start users, for whom interaction sequences are unavailable, relying on their feature information. This facilitates the seamless integration of CMCLRec into pre-existing sequential recommendation models without compromising the original model's effectiveness for warm users. Furthermore, the incorporation of self-supervised learning and contrastive learning into the framework is executed

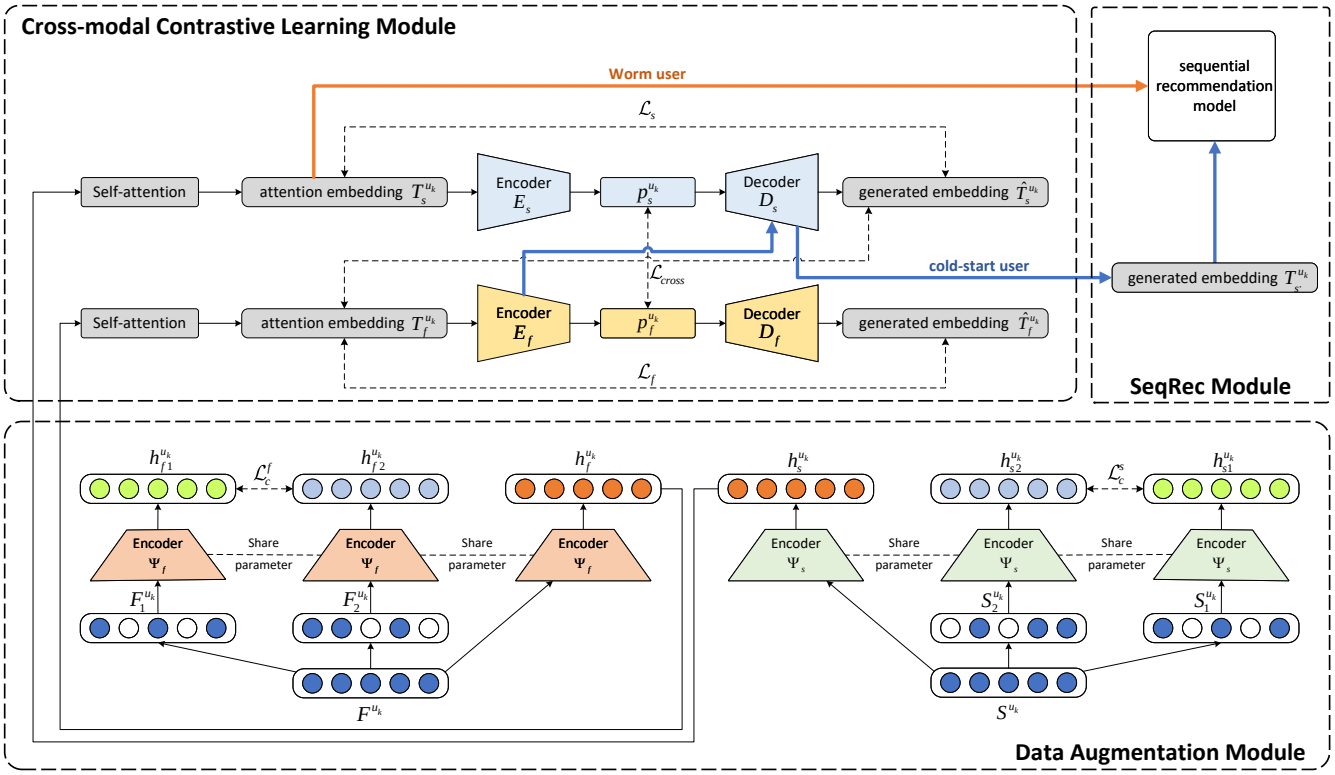


Figure 1: The overall architecture of CMCLRec.

without a concomitant escalation in overall training complexity or necessitating additional labeled data.

Let U denote the user set, I denote the item set, and $u \in U$ denote a user. The historical interaction sequence for user u is denoted as $S^u = \{i_1^u, i_2^u \dots i_n^u\}$, where n is the length of the interaction sequence, and $i_t^u \in I(1 <= t <= n)$ represents the item at position t interacted with by user u. Additionally, user u is associated with feature information F^u. The goal of our framework is to use the feature information F^u of a cold-start user to derive an embedding for the interaction sequences. This derived embedding is then utilized to generate recommendations for the cold-start user.

Given that F^u may not be sufficient, utilizing these features to infer behavior sequences proves challenging. Therefore, a contrastive learning approach is initially employed to enhance F^u for $u \in U$, aiming to acquire more comprehensive user features. Using cross-modal contrastive learning, we construct a cross-modal mapping from F^u to S^u and generate simulated behavior sequences for cold-start users. For warm users, direct recommendations are made using the enhanced S^u; for cold-start users, the generated simulated behavior sequences are used to predict their preferences.

3.3 Data Augmentation Module

Owing to the incompleteness of registration information for a considerable number of users and the inadequate richness of feature content, the data augmentation model employs a contrastive learning methodology to augment the features of the data, as illustrated in the data augmentation module depicted in Figure 1. It incentivizes the model to constrict the proximity of embeddings corresponding to similar users within the embedding space, while concurrently amplifying the separation between embeddings associated with dissimilar users.

Given a batch size of N, for each user $u_k (1 \le k \le N)$, the feature F^{u_k} is subjected to different data augmentation methods, resulting in partially masked features $F_1^{u_k}$ and $F_2^{u_k}$. It is ensured that the features masked in $F_1^{u_k}$ are inconsistent with those masked in $F_2^{u_k}$, where the data augmentation function is denoted as Φ. The augmented feature representation is then presented by:

$$F_1^{u_k} = \varphi_{f1}(F^{u_k}), F_2^{u_k} = \varphi_{f2}(F^{u_k}), s.t.\ \varphi_{f1}, \varphi_{f2} \in \Phi, \tag{1}$$

where both φ_1 and φ_2 are distinct data augmentation functions, and $F_1^{u_k}$ and $F_2^{u_k}$ represent the different feature embeddings generated from F^{u_k} through these two functions, respectively.

Based on the enhanced $F_1^{u_k}$ and $F_2^{u_k}$, an encoder, denoted by $\Psi_f(\cdot)$, is applied with a similar structure. This results in the encoded embeddings $h_{f1}^{u_k}$ and $h_{f2}^{u_k}$, ensuring parameter consistency throughout the training process. In this way, the encoded embeddings $h_{f1}^{u_k}$ and $h_{f2}^{u_k}$ can be expressed as:

$$h_{f1}^{u_k} = \Psi_f(F_1^{u_k}),\ h_{f2}^{u_k} = \Psi_f(F_2^{u_k}), \tag{2}$$

where $F_1^{u_k}$ and $F_2^{u_k}$ can both be regarded as containing partial information from F^{u_k}.

The purpose of the data augmentation model is to expand user features as much as possible, which can be analogized as restoring F^{u_k} from $F_1^{u_k}$ and $F_2^{u_k}$. Therefore, for their embeddings $h_{f1}^{u_k}$ and $h_{f2}^{u_k}$, it is necessary to minimize the distribution gap between each embedding and its corresponding feature, while simultaneously maximizing the distribution gap between embeddings of different users. In detail, we employ cosine similarity to represent the similarity between embeddings, aiming to minimize the similarity among similar users (as shown in the numerator of Eq. (3)) while maximizing the dissimilarity among dissimilar users (as shown in the denominator of Eq. (3)). The contrastive loss $\mathscr{L}_c^f$ can be calculated as:

$$\mathscr{L}_c^f = -\frac{1}{N}\sum_{k=1}^{N}\log\left(\frac{\exp\left(s(h_{f1}^{u_k}, h_{f2}^{u_k})/\tau\right)}{\sum_{t=1}^{N}\exp\left(s(h_{f1}^{u_k}, h_{f2}^{u_t})/\tau\right)}\right), \tag{3}$$

where the function $s(\cdot)$ represents the similarity between two vectors, serving to quantify the distribution gap between them, and τ is a pre-defined hyperparameter that governs the model's discriminative capacity concerning negative samples. A too-large τ value may result in insufficient discrimination between positive and negative samples, thereby contributing to suboptimal model performance. Conversely, a too-small τ value may cause the model to overly focus on negative samples, making it challenging for the model to converge. It is evident that by minimizing the contrastive loss, the distances between positive samples can be reduced while simultaneously increasing the distances between negative samples.

Subsequently, the same methodology is applied to process user behavior sequences. For a user $u_k (1 \le k \le N)$ and its behavior sequence S^{u_k}, the final encoded embeddings $h_{s1}^{u_k}$ and $h_{s2}^{u_k}$ are generated. The contrastive loss can be calculated with:

$$\mathscr{L}_c^s = -\frac{1}{N}\sum_{k=1}^{N}\log\left(\frac{\exp\left(s(h_{s1}^{u_k}, h_{s2}^{u_k})/\tau\right)}{\sum_{t=1}^{N}\exp\left(s(h_{s1}^{u_k}, h_{s2}^{u_t})/\tau\right)}\right). \tag{4}$$

Due to the powerful capabilities of contrastive learning and the relatively uncomplicated nature of user feature content, a basic Multi-Layer Perceptron (MLP) is utilized as the encoder for contrasting user features. However, a basic MLP cannot handle more complex user behavior sequences. Here, we utilize a Transformer Encoder with enhanced expressive capacity for encoding. The incorporation of this amalgamation of contrastive learning and self-supervised learning facilitates the extraction of more comprehensive information features from both user features and behavior sequences, obviating the necessity for additional data. This streamlines the subsequent implementation of cross-modal fusion.

3.4 Cross-modal Contrastive Learning Module

User features F^{u_k} and behavior sequences S^{u_k} are directly input into the data augmentation model, enhancing user feature embeddings denoted as $h_f^{u_k}$ and $h_s^{u_k}$.

Conventional deep networks struggle to attend to the entire sequence information effectively, and they may not fully utilize the comprehensive information within $h_f^{u_k}$ and $h_s^{u_k}$ along with their implicit combined information. Thus, a self-attention mechanism is employed for embedding construction. Taking the user behavior sequence as an example, for $h_s^{u_k} = \{a_1, a_2, \ldots, a_n\}$, where $a_i (1 \le i \le n)$ is a column vector representing the embedding of the corresponding item at position i. Introducing transformation matrices M_1, M_2, and M_3, subject to parameter updates through learning, the transformation for a_i is as follows:

$$\sigma_1^i = M_1 a_i,\ \sigma_2^i = M_2 a_i,\ \sigma_3^i = M_3 a_i, \tag{5}$$

where σ_1^i and σ_2^j are used to compute the similarity between a_i and a_j, while σ_3^i encapsulates the original information of a_i.

Subsequently, the similarity between a_i and a_j, denoted as $\alpha_{i,j}$, is calculated using the following formula:

$$\alpha_{i,j} = \frac{\exp\left(\sigma_1^i \cdot \sigma_2^j/\sqrt{d}\right)}{\sum_{k=1}^{n}\exp\left(\sigma_1^i \cdot \sigma_2^k/\sqrt{d}\right)}, \tag{6}$$

where d represents the dimensionality of σ_1^i and σ_2^j. Since the obtained $\alpha_{i,j}$ after the inner product tends to increase with the dimensionality, normalization is required to prevent unnecessary errors caused by dimensionality.

Through the self-attention operation, $h_s^{u_k}$ is transformed into $T_s^{u_k} = \{b_1, b_2, \ldots, b_n\}$, where $b_i (1 \le i \le n)$ is represented as:

$$b_i = \sum_{j=1}^{n}\alpha_{i,j} \cdot \sigma_3^i. \tag{7}$$

Evidently, each b encompasses all the information from a. Furthermore, in this layer, there are only three transformation matrices: M_1, M_2, and M_3. Without significantly increasing the training complexity, a weighted approach is applied to the distinct features of items, allowing for the collection of combined information among various items within $h_s^{u_k}$. This significantly reduces the subsequent challenges in modal fusion. Similarly, for $h_f^{u_k}$, it undergoes a self-attention operation and transforms into $T_f^{u_k} = \{b_1, b_2, \ldots, b_n\}$.

Based on the $T_s^{u_k}$ and $T_f^{u_k}$ generated by self-attention, a mapping from user features to behavior sequences is constructed using autoencoders and cross-modal learning methods. E_s and E_f denote the encoders for $T_s^{u_k}$ and $T_f^{u_k}$, respectively, while D_s and D_f represent their corresponding decoders. Let $\widehat{T}_s^{u_k} = D_s(E_s(T_s^{u_k}))$, $\widehat{T}_f^{u_k} = D_f(E_f(T_f^{u_k}))$, ensuring that the encoded-decoded results preserve the information from $T_s^{u_k}$ and $T_f^{u_k}$ as much as possible. The autoencoder loss can be expressed as:

$$\begin{aligned}\mathscr{L}_s &= \sum_{u_k \in U_b} ||\widehat{T}_s^{u_k} - T_s^{u_k}||_2^2, \\ \mathscr{L}_f &= \sum_{u_k \in U_b} ||\widehat{T}_f^{u_k} - T_f^{u_k}||_2^2.\end{aligned} \tag{8}$$

Let $p_s^{u_k} = E_s(T_s^{u_k})$, $p_f^{u_k} = E_f(T_f^{u_k})$. This establishes the foundation for the implementation of cross-modal learning methods. In this context, the model is trained to acquire the mapping from $p_f^{u_k}$ to $\widehat{T}_s^{u_k}$ while concurrently minimizing the distribution gap between $p_s^{u_k}$ and $p_f^{u_k}$. This enables both vectors to exhibit the capacity for predicting $\widehat{T}_s^{u_k}$. Despite the significant distribution difference between $p_s^{u_k}$ and $p_f^{u_k}$, there exists a correlation between them as both are generated by the user u_k. Therefore, adopting a transfer

learning-like approach, coupled with the Max Mean Discrepancy (MMD) loss function, facilitates cross-modal learning by minimizing the distribution gap between $p_s^{u_k}$ and $p_f^{u_k}$. The loss function $\mathscr{L}_{cross}$ is calculated as follows:

$$\begin{aligned}\mathscr{L}_{cross} &= \mathrm{MMD}_{\mathcal{H}}\left(P_s, P_f\right) + \sum_{u_k \in U_b} \left\| \widehat{T}_s^{u_k} - T_f^{u_k} \right\|_2^2 \\ &= \left\| \frac{1}{N}\sum_{i=1}^{N} \phi(p_s^{u_i}) - \frac{1}{N}\sum_{i=1}^{N} \phi(p_f^{u_i}) \right\| + \sum_{u_k \in U_b} \left\| \widehat{T}_s^{u_k} - T_f^{u_k} \right\|_2^2 \\ &= \left\| \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \kappa(p_s^{u_i}, p_s^{u_j}) - \frac{2}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \kappa(p_s^{u_i}, p_f^{u_j}) \right. \\ &\quad \left. + \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \kappa(p_f^{u_i}, p_f^{u_j}) \right\| + \sum_{u_k \in U_b} \left\| \widehat{T}_s^{u_k} - T_f^{u_k} \right\|_2^2 .\end{aligned} \tag{9}$$

For a batch, $P_s = \{p_s^{u_1}, p_s^{u_2}, \ldots, p_s^{u_N}\}$, $P_f = \{p_f^{u_1}, p_f^{u_2}, \ldots, p_f^{u_N}\}$, where N denotes the batch size. Let k denote the Gaussian kernel function, $\phi : x \to \mathcal{H}$ denote the feature mapping, and $\mathcal{H}$ denote the reproducing kernel Hilbert space corresponding to κ. The kernel function κ can be expressed as:

$$\kappa(x, y) = \exp\left\{-\frac{(x-y)^2}{2\sigma^2}\right\}, \tag{10}$$

where the parameter σ regulates the range of the Gaussian kernel function with an increased value signifying a more extensive local impact range for the Gaussian kernel function.

The overall loss function for this module is as follows:

$$\mathscr{L}_{c2l} = \mathscr{L}_s + \mathscr{L}_f + \lambda \mathscr{L}_{cross}, \tag{11}$$

where λ represents the weight of cross-modal learning, and an excessively large value can impede model convergence, while an overly small value can result in poorer reconstruction ability, insufficient information in the simulated behavior sequence, and suboptimal recommendation performance.

Following the completion of cross-modal learning, $p_s^{u_k}$ and $p_f^{u_k}$ are similar in terms of distributions and possess predictive capabilities for $\widehat{T}_s^{u_k}$. Therefore, for warm users, direct recommendations can be made using the behavior sequence $T_s^{u_k}$. For cold-start users, $T_f^{u_k}$ are used to calculate $p_f^{u_k}$ through encoder E_f, and the simulated behavior sequence $\widehat{T}_{s'}^{u_k}$ is obtained through decoder D_s for user recommendations.

3.5 Sequential Recommendation Module

After the cross-modal learning module, the model has acquired a mapping from user features to behavior sequences. Therefore, the next step involves making recommendations to users. The method we propose does not impose specific requirements on the implementation of the sequential recommendation model. In theory, any existing sequential recommendation model can be employed, enhancing its recommendation performance in cold-start scenarios. The sequential recommendation model is represented by the function $\mathrm{R}(\cdot)$.

As the recommendation approaches for warm users and cold-start users differ in the model, $flag \in \{0, 1\}$ is introduced to distinguish between them:

$$flag = \begin{cases} 0, \text{ warm user} \\ 1, \text{ cold-start user} \end{cases}. \tag{12}$$

Regarding the input to the recommendation network, it can be differentiated based on the $flag$. For a batch of size N, where the users within the batch are denoted as $U_b = \{u_1, u_2 \ldots, u_N\}$, the behavior sequence of a warm user u_k can be represented as $T_s^{u_k} = \{i_1^{u_k}, i_2^{u_k} \ldots, i_n^{u_k}\}$. The simulated behavior sequence for a cold-start user u_j can be represented as $\widehat{T}_{s'}^{u_j} = \{i_1^{u_j}, i_2^{u_j} \ldots, i_n^{u_j}\}$. For the input X to the network, it can be expressed as:

$$X^{u_k} = flag \cdot \widehat{T}_{s'}^{u_k} + (1 - flag) \cdot T_s^{u_k}. \tag{13}$$

Following the recommendation model, its output is denoted as $G^{u_k} = \mathrm{R}(X^{u_k})$, where $G^{u_k} = \{g_1^{u_k}, g_2^{u_k}, \ldots, g_n^{u_k}\}$. The loss function can be expressed as:

$$\mathscr{L}_{rec} = -\frac{1}{N}\sum_{u_k \in U_b} \left\{ \mathrm{S}(g_n^{u_k}) - \ln\left(\sum_{i \in I} \exp\left(\mathrm{S}(i_e)\right) \right) \right\}, \tag{14}$$

where i_e denotes the embedding of item i, and $\mathrm{S}(\cdot)$ signifies the softmax function. The optimization of this cross-entropy loss function is directed towards maximizing the probability of accurate predictions.

3.6 Training Strategy

Within the data augmentation and cross-modal contrastive learning modules, CMCLRec aims to explore latent user features and extract the mapping relationship between user features and user-item interactions. These modules operate independently of the sequential recommendation module. Additionally, the first two modules employ self-supervised training, obviating the need for labeled data, rendering them apt for independent large-scale training. Consequently, the training of the comprehensive framework is conducted in two distinct stages as shown in Algorithm 1.

In the first stage, pre-training is conducted for the initial two modules, updating the parameters of the encoders Ψ_f and Ψ_s using the contrastive loss $\mathscr{L}_c^s$ (Eq. (3)) and $\mathscr{L}_c^s$ (Eq. (4)). Regarding $\mathscr{L}_2$ in Eq. (11), its update is performed in conjunction with the contrastive loss functions due to its dependence on feature enhancement and strong correlation with the data augmentation module. This stage's overall loss function can be expressed as follows:

$$\mathscr{L}_{pre} = \alpha \mathscr{L}_c^f + \beta \mathscr{L}_c^s + \mathscr{L}_{c2l}, \tag{15}$$

where the parameters α and β serve to adjust the magnitude of enhancement for user features and user-item interaction. In particular, training data in this stage is exclusively sourced from warm users.

In the second stage, the initial two modules are incorporated into the sequential recommendation module and are subjected to fine-tuning. This phase primarily enhances the model's recommendation capabilities, utilizing $T_s^{u_k}$ for warm user input, as discussed in Section 3.4, and $\widehat{T}_{s'}^{u_k}$ for cold-start user input. The overall loss

function for this stage can be expressed as:

$$\mathscr{L}_{fine} = \eta \mathscr{L}_{pre} + \mathscr{L}_{rec}, \tag{16}$$

where the parameter η is utilized to regulate the fine-tuning magnitude for the first two modules. An excessively large parameter may lead to overfitting issues, diminishing the model's generalization capability. Conversely, an overly small parameter might result in insufficient expressive power, causing the simulated sequences to inadequately fit cold-start users and thereby reducing recommendation performance.

The comprehensive training procedure is delineated in Algorithm 1.

Algorithm 1 CMCLRec

Input: F^u and S^u for $u \in U(flag = 0)$, F^u for $u \in U(flag = 1)$, learning rate lr, hyperparameters $\alpha, \beta, \lambda, \eta$.
Output: Global model parameters $\mathscr{W}$ (composed $\mathscr{W}_{pre}$ of and $\mathscr{W}_{rec}$).

Remove user-item interaction of some users to simulate cold start users.
Set $flag$ for all users by Eq. (12).
/* Self-supervised learning stage. */
for $i \leftarrow 1 : E_1$ (number of pre-train epochs) **do**
  for $j \leftarrow 1 : B_1$ (number of pre-train batch size) **do**
    Calculate $\mathscr{L}_c^f$ for users ($flag = 0$) by Eq. (3).
    Calculate $\mathscr{L}_c^s$ for users ($flag = 0$) by Eq. (4).
    Calculate $\mathscr{L}_s$, $\mathscr{L}_f$ for users ($flag = 0$) by Eq. (8).
    Calculate $\mathscr{L}_{cross}$ for users ($flag = 0$) by Eq. (9).
    Set $\mathscr{L}_{pre}$ by Eq. (15).
    Apply Adam optimizer to $\mathscr{L}_{pre}$.
    Perform back-propagation to $\mathscr{L}_{pre}$ getting gradients $\mathscr{G}$.
    Update $\mathscr{W}_{pre}$ based on $\mathscr{G}$.
  end for
end for
/* Recommended training stages. */
for $i \leftarrow 1 : E_2$ (number of fine-train epochs) **do**
  for $j \leftarrow 1 : B_2$ (number of fine-train batch size) **do**
    Calculate $\mathscr{L}_{pre}$ according to the previous stage.
    Get X^{u_k} by Eq. (13).
    Calculate $\mathscr{L}_{rec}$ for all user by Eq. (14).
    Set $\mathscr{L}_{fine}$ by Eq. (16).
    Apply Adam optimizer to $\mathscr{L}_{fine}$.
    Perform back-propagation to $\mathscr{L}_{fine}$ getting $\mathscr{G}$.
    Update $\mathscr{W}_{pre}$ and $\mathscr{W}_{rec}$ based on $\mathscr{G}$.
  end for
end for
return $\mathscr{W}$

4 EXPERIMENT

We conducted comparison experiments and ablation experiments on publicly available datasets to address the following three research questions:

- **RQ1:** Can CMCLRec achieve the best cold start and overall recommendation performance compared to state-of-the-art cold-start solutions?
- **RQ2:** Whether CMCLRec reduces the distribution gap between cold-start and warm users during the self-supervised learning phase?
- **RQ3:** Can integrating CMCLRec into a regular sequential recommendation model effectively improve recommendation performance for cold-start users, and what are the effects of different key components?

4.1 Experimental Setup

Datasets. We conduct experiments to evaluate CMCLRec's performance on two publicly available datasets: KuaiRec [7] and XING [1]. KuaiRec constitutes a real-world dataset sourced from recommendation logs within the mobile video-sharing application, i.e., Kuaishou, containing 1141 users and 3327 items with 4,676,570 user-item interactions. XING is a subset derived from the ACM RecSys 2017 challenge dataset, comprising 106,881 users, 20,519 jobs, and 4,306,183 interactions. A 2,738-dimensional vector is employed to represent the job content, capturing diverse attributes including career level, tags, and supplementary information. 30% of users are partitioned into the test set, with 15% undergoing no adjustments and the remaining 15% having their user-item interactions removed to simulate cold-start users for each dataset. The division of the remaining users into training and validation sets adheres to an 8:2 ratio.

Evaluation Metrics. We conduct separate evaluations for the overall performance, warm recommendation performance, and cold-start recommendation performance. The evaluation is performed using two widely used evaluation metrics, including Recall@K and Normalized Discounted Cumulative Gain (NDCG)@K. The higher Recall@K and NDCG@K indicate the higher the rating prediction accuracy and the better the ranking performance for most of the preferred items. Similar to [10, 18], K is set to 20 by default. For each metric, the results were averaged across all users and averaged over five independent experiments.

Baselines. In this experiment, CMCLRec uses SASRec [11] as its recommendation model. In assessing the effectiveness of CMCLRec for recommending both cold-start and warm users, we conducted comparisons with five cold-start recommendation models across two datasets.

- **DropoutNet** [24] improves the cold-start problem by randomly discarding embeddings to reduce the model's dependence on user-item interactions.
- **DeepMusic** [21] employ a deep convolutional neural network to project users and items into a low-dimensional implicit space. Recommendations are then generated by assessing the positional relationships in this space.
- **MeLU** [12] A predicts preferences for cold-start users, leveraging meta-learning, based on consumed items, and strategically addresses the user cold-start issue through the Model-Agnostic Meta-Learning (MAML) approach.
- **Heater** [41] introduces a Mixture-of-Experts Transformation mechanism to Enhance DropoutNet, providing 'personalized' transformation functions.
- **PDMA** [26] enhances a preference learning decoupling framework using meta-augmentation to improve user cold-start recommendation.

Table 1: Recommendation performance comparison against baselines. The improvements are calculated by comparing CMCLRec with the corresponding best baselines (underlined).

Method	Overall Recommendation				Cold-Start Recommendation				Warm Recommendation			
	KuaiRec		XING		KuaiRec		XING		KuaiRec		XING	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
DeepMusic	0.0375	0.0186	0.1876	0.1683	0.0394	0.0192	0.2691	0.1606	0.0414	0.0285	0.4205	0.2946
DropoutNet	0.0252	0.0092	0.1733	0.1507	0.0306	0.0143	0.2773	0.1953	0.0367	0.0271	0.3034	0.2182
MeLU	0.0351	0.0115	0.1829	0.1713	0.0418	0.0162	0.2842	0.2304	0.0392	0.0294	0.4116	0.3247
Heater	0.0392	0.0153	0.2095	0.1830	0.0463	0.0179	0.3271	0.2124	0.0449	0.0327	0.3982	0.2708
PDMA	0.0437	0.0176	0.2343	0.2157	0.0413	0.0209	0.3679	0.2416	0.0324	0.0319	0.3823	0.2886
SDCRec	0.0397	0.0209	0.2231	0.1973	0.0492	0.0187	0.3018	0.2037	0.0417	0.0348	0.3421	0.2947
CMCLRec	**0.0481**	**0.0221**	**0.2476**	**0.2282**	**0.0563**	**0.0231**	**0.4116**	**0.2606**	**0.0484**	**0.0356**	**0.4556**	**0.3418**
improv.	10.07%	5.74%	5.68%	5.80%	14.43%	10.53%	11.88%	7.86%	7.80%	2.30%	8.35%	5.27%

- **SDCRec** [5] seamlessly models cold-start users as warm users using a social ensemble, without using additional user-item interaction records, to improve referrals for cold-start users.

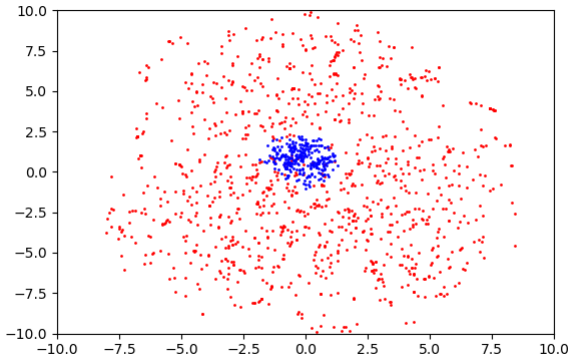
Implementation Details. We use the officially provided source code to implement Baselines. In this experiment, CMCLRec adopts SASRec [11] as its recommendation model. The Embedding dimension is set to 64 for all baseline models and CMCLRec. We employ the Adam optimizer with a learning rate of 1×10^{-4}. For Eq. (11), the default value for λ is set to 0.3. For Eq. (15), the default values for α and β are both set to 0.5. For Eq. (16), the default value for η is set to 0.7. In the interest of fairness, all baseline models are configured with identical hyperparameters and designs as specified in their respective articles.

4.2 Performance Comparison (RQ1)

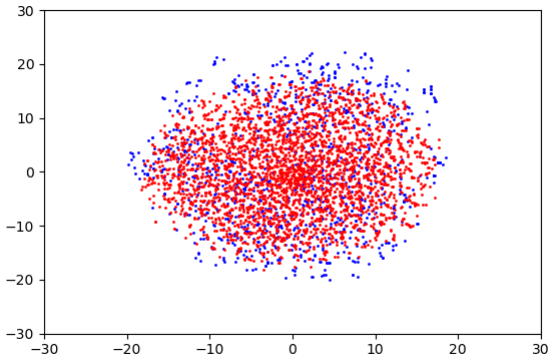
The comparative evaluation of CMCLRec with other baseline models on the two datasets is presented in Table 1, encompassing assessments of overall recommendation performance, cold-start user recommendation performance, and warm user recommendation performance. To evaluate the effectiveness of CMCLRec, we conducted comparative experiments with a generative model (DeepMusic), two dropout models (DropoutNet, Heater), two meta-learning models (MeLU, PDMA), and a contrastive learning model (SDCRec). The improvement can be calculated by comparing CMCLRec with the optimal baseline (underline).

Upon scrutinizing the table, it is evident that in the overall recommendation scenario, CMCLRec exhibited superior performance compared to other baseline models across both datasets. Specifically, on the NDCG@20 metric, CMCLRec achieved an improvement of 5.74% (KuaiRec) and 5.80% (XING) compared to the best baseline model PDMA. Moreover, on the Recall@20 metric, CMCLRec achieved an improvement of 10.07% (KuaiRec) and 5.68% (XING) compared to the best baseline model SDCRec.

For the cold-start recommendation scene, CMCLRec exhibits more significant improvements compared to baseline models. Specifically, compared to the best baseline model, CMCLRec achieved an increase of 10.53% (KuaiRec) and 7.86% (XING) in NDCG@20, and an increase of 14.43% (KuaiRec) and 11.88% (XING) in Recall@20. For the warm recommendation scene, in comparison to the best baseline models, CMCLRec showed an increase of 2.30% (KuaiRec) and 5.27% (XING) in NDCG@20, and an increase of 7.80% (KuaiRec) and 8.35% (XING) in Recall@20.



(a) Distribution of $\widehat{T}_{s'}^{u_k}$ and $\widehat{T}_{s}^{u_k}$ before the self-supervised training.



(b) Distribution of $\widehat{T}_{s'}^{u_k}$ and $\widehat{T}_{s}^{u_k}$ after the self-supervised training.

Figure 2: Comparison between $\widehat{T}_{s'}^{u_k}$ (cold-start user, blue) and $\widehat{T}_{s}^{u_k}$ (worm user, red).

Table 2: Results of Ablation Study

Method	Overall Recommendation				Cold-Start Recommendation				Warm Recommendation			
	KuaiRec		XING		KuaiRec		XING		KuaiRec		XING	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
CMCLRec-NoDA	0.0468	0.022	0.2318	0.2204	0.0541	0.0218	0.4012	0.2421	0.0471	0.0312	0.4248	0.3257
CMCLRec-NoCL	0.0476	0.0217	0.2361	0.2143	0.0422	0.0162	0.3758	0.227	0.0473	0.0334	0.4409	0.3215
CMCLRec-None	0.0437	0.0209	0.2242	0.1972	0.0417	0.0148	0.3596	0.2147	0.0451	0.0267	0.4185	0.2803
CMCLRec	0.0481	0.0221	0.2476	0.2282	0.0563	0.0231	0.4116	0.2606	0.0484	0.0356	0.4556	0.3418

In all scenes, CMCL outperforms the best baseline model, which is attributed to the success of cross-modal contrastive learning in establishing a mapping between user features and behavior sequences . This effectively improves the feature distribution of cold-start users. Simultaneously, for warm users, CMCL also enhances their behavior sequence embeddings effectively, resulting in a noticeable improvement in recommendation performance.

4.3 Visualization Study (RQ2)

In this section, we utilize t-SNE [22] to reduce the distribution dimensions of the simulated behavior sequence embedding $\widehat{T}_{s'}^{u_k}$ and the behavior sequence embedding $\widehat{T}_{s}^{u_k}$ defined in Section 4.3 to 2 dimensions and conduct visual analysis, as shown in Figure 2, to verify whether the disparity between cold-start users ($\widehat{T_{s'}^{u_k}}$) and warm users ($T_s^{u_k}$) has effectively been reduced.

From sub-figure (a), we observe that, for the model, there is a significant distribution disparity between cold-start users and warm users before training, and CMCLRec shows almost no distinguishability for cold-start users. Such phenomenon is attributed to the absence of historical behavioral sequence for cold-start users, thereby compelling the model, which heavily relies on such interactions, to treat all cold-start users uniformly. After self-supervised training, the distribution of cold-start users and warm users becomes nearly identical, and the improvement in the distribution of cold-start users is significant, as shown in sub-figure (b). This signifies that the model has gained distinctiveness for the post-training cold-start user data, generating different simulated behavior sequences for various cold-start users. Therefore, during self-supervised learning, CMCLRec effectively reduces the distribution gap between cold-start users and warm users, which confirms the rationality of using simulated behavioral sequences to recommend cold-start users.

4.4 Ablation Study (RQ3)

In comparison to traditional sequential recommendation models, CMCLRec incorporates additional data augmentation and cross-modal contrastive learning modules. Through the Ablation Study, we investigate the effectiveness of these two modules and design the following three variant models for comparison based on CMCLRec.

- **CMCLRec-NoDA:** Removed the data augmentation module and directly employed the original embeddings for cross-modal contrastive learning.
- **CMCLRec-NoCL:** Removed the cross-modal contrastive learning module and directly applied linear mapping to the augmented data.
- **CMCLRec-None:** Removed the data augmentation module as well as the cross-modal contrastive learning module and performed a direct linear mapping on the original embeddings.

The observations derived from the data presented in Table 2 are summarized as follows. Firstly, the removal of the cross-modal contrastive learning module results in a notable decline in recommendation performance in the cold-start recommendation scene. This is because CMCLRec-NoCL directly uses linear mapping to generate simulated behavior sequences for cold-start users, which deviates substantially from their actual preferences, resulting in poor recommendation performance. Secondly, omitting the data augmentation module results in a reduction in recommendation effectiveness across all scenes. This is because CMCLRec-NoDA directly models the original data embeddings, which inadequately captures the preferences of users with fewer features, leading to decreased recommendation performance. Thirdly, upon removing the initial two modules and the cross-modal contrastive learning module, CMCLRec-None experiences a further decrease in recommendation accuracy compared to CMCLRec-NoDA and CMCLRec-NoCL. This suggests that both modules have a positive impact on user recommendations in all scenes.

5 CONCLUSION

In this paper, we propose CMCLRec that addresses the cold-start problem in sequential recommendation by generating simulated behavior sequences for cold-start users. CMCLRec leverages the data of warm users and employs the cross-modal contrastive learning method to construct a mapping from user features to behavior sequences. Since the embedding of simulated behavior sequences can be directly used as input for conventional sequential recommendation models for cold-start users, CMCLRec can be directly embedded to improve the recommendation performance of cold-start users for specially designed sequential recommendation models. Experimental results based on two publicly available datasets demonstrate that CMCLRec outperforms state-of-the-art baseline models in both cold-start and warm recommendation scenes. Besides, the ablation study confirms that CMCLRec effectively enhances the recommendation performance of sequential recommendation models for cold-start users.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant (no. 92267104 and 62372242).

REFERENCES

[1] Fabian Abel, Yashar Deldjoo, Mehdi Elahi, and Daniel Kohlsdorf. 2017. Recsys challenge 2017: Offline and online evaluation. In *Proceedings of the eleventh acm conference on recommender systems*. 372–373.

[2] Veselka Boeva and Christian Nordahl. 2019. Modeling evolving user behavior via sequential clustering. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 12–20.

[3] Hao Chen, Zefan Wang, Feiran Huang, Xiao Huang, Yue Xu, Yishi Lin, Peng He, and Zhoujun Li. 2022. Generative adversarial framework for cold-start item recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2565–2571.

[4] Shi Dong, Ping Wang, and Khushnood Abbas. 2021. A survey on deep learning and its applications. *Computer Science Review* 40 (2021), 100379.

[5] Jing Du, Zesheng Ye, Lina Yao, Bin Guo, and Zhiwen Yu. 2022. Socially-aware dual contrastive learning for cold-start recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1927–1932.

[6] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)* 39, 1 (2020), 1–42.

[7] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. KuaiRec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 540–550.

[8] Ruining He, Wang-Cheng Kang, Julian J McAuley, et al. 2018. Translation-based Recommendation: A Scalable Method for Modeling Sequential Behavior.. In *IJCAI*. 5264–5268.

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

[10] Feiran Huang, Zefan Wang, Xiao Huang, Yufeng Qian, Zhetao Li, and Hao Chen. 2023. Aligning Distillation For Cold-start Item Recommendation. (2023).

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

[12] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1073–1082.

[13] Weiming Liu, Xiaolin Zheng, Jiajie Su, Longfei Zheng, Chaochao Chen, and Mengling Hu. 2023. Contrastive Proxy Kernel Stein Path Alignment for Cross-Domain Cold-Start Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[14] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 695–704.

[15] Xingyu Pan, Yushuo Chen, Changxin Tian, Zihan Lin, Jinpeng Wang, He Hu, and Wayne Xin Zhao. 2022. Multimodal meta-learning for cold-start sequential recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*. 3421–3430.

[16] Xiuyuan Qin, Huanhuan Yuan, Pengpeng Zhao, Junhua Fang, Fuzhen Zhuang, Guanfeng Liu, Yanchi Liu, and Victor Sheng. 2023. Meta-optimized Contrastive Learning for Sequential Recommendation. *arXiv preprint arXiv:2304.07763* (2023).

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

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

[19] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)* 47, 1 (2014), 1–45.

[20] 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.

[21] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. *Advances in neural information processing systems* 26 (2013).

[22] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[23] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A meta-learning perspective on cold-start recommendations for items. *Advances in neural information processing systems* 30 (2017).

[24] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. Dropoutnet: Addressing cold start in recommender systems. *Advances in neural information processing systems* 30 (2017).

[25] Chunyang Wang, Yanmin Zhu, Haobing Liu, Tianzi Zang, Jiadi Yu, and Feilong Tang. 2022. Deep Meta-learning in Recommendation Systems: A Survey. *arXiv preprint arXiv:2206.04415* (2022).

[26] Chunyang Wang, Yanmin Zhu, Aixin Sun, Zhaobo Wang, and Ke Wang. 2023. A Preference Learning Decoupling Framework for User Cold-Start Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1168–1177.

[27] Jianling Wang, Kaize Ding, and James Caverlee. 2021. Sequential recommendation for cold-start users with meta transitional learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1783–1787.

[28] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830* (2019).

[29] Xiao Wang and Guo-Jun Qi. 2022. Contrastive learning with stronger augmentations. *IEEE transactions on pattern analysis and machine intelligence* 45, 5 (2022), 5549–5560.

[30] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2016. Collaborative filtering and deep learning based hybrid recommendation for cold start problem. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. IEEE, 874–877.

[31] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive learning for cold-start recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 5382–5390.

[32] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 346–353.

[33] Jie Xu, Tianwei Xing, and Mihaela Van Der Schaar. 2016. Personalized course sequence recommendations. *IEEE Transactions on Signal Processing* 64, 20 (2016), 5340–5352.

[34] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4321–4330.

[35] Rong Ye, Mingxuan Wang, and Lei Li. 2022. Cross-modal contrastive learning for speech translation. *arXiv preprint arXiv:2205.02444* (2022).

[36] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 269–277.

[37] Eva Zangerle and Christine Bauer. 2022. Evaluating recommender systems: survey and framework. *Comput. Surveys* 55, 8 (2022), 1–38.

[38] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. 2021. Cross-modal contrastive learning for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 833–842.

[39] Yu Zhu, Jinghao Lin, Shibi He, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2019. Addressing the item cold-start problem by attribute-driven active learning. *IEEE Transactions on Knowledge and Data Engineering* 32, 4 (2019), 631–644.

[40] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1167–1176.

[41] Ziwei Zhu, Shahin Sefati, Parsa Saadatpanah, and James Caverlee. 2020. Recommendation for new users and new items via randomized training and mixture-of-experts transformation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1121–1130.

[42] Mohammadreza Zolfaghari, Yi Zhu, Peter Gehler, and Thomas Brox. 2021. Crossclr: Cross-modal contrastive learning for multi-modal video representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1450–1459.