Contrastive Translation With Dynamical Temperature for Sequential Recommendation

Aoran Zhang, Yonghong Yu, Li Zhang, *Senior Member, IEEE*, Rong Gao, and Hongzhi Yin, *Senior Member, IEEE*

***Abstract*—Contrastive learning is a promising solution to the problem of data sparsity in the field of recommendation system since it is able to extract self-supervised signals from raw data. The traditional contrastive learning-based sequential recommendation algorithms generate augmentations of original item sequences by utilizing crop, mask and reorder operations. However, those augmentation schemes destroy the underlying semantics of item sequences, resulting in difficulty in accurately defining positive and negative samples. To address this issue, we propose a contrastive translation based sequential recommendation algorithm, namely, CT4Rec. Specifically, CT4Rec generates augmented views of item sequences by injecting noises into embeddings of users and items, which is able to guarantee that the underlying semantics of augmented views are consistent with those of original item sequence. Hence, CT4Rec is able to effectively learn the invariances among the augmented views. In addition, the personalized translation operations are utilized to model the third-order relationships among entities. Moreover, it is difficult for contrastive learning-based recommendation algorithms with static temperature to simultaneously capture the differences among individual users/items and among the clusters of users/items. Hence, we utilize a dynamic temperature strategy to enhance CT4Rec, which endows CT4Rec with the capabilities of group-wise discrimination and instance discrimination. Our validation on five benchmark datasets shows that CT4Rec outperforms SOTA sequential recommendation methods. Our code is released at https://github.com/zar123123/CT4Rec.**

***Index Terms*—Contrastive learning, knowledge graph embedding, sequential recommendation.**

I. Introduction

TRADITIONAL recommendation systems [1], [2], [3] usually learn user preferences by modeling their historical behaviors and ignore the impact of temporal patterns. However, the decision-making of users is mainly influenced by user general preferences and sequential behavior patterns in practical recommendation scenarios. Compared to traditional item recommendations, sequential recommendation systems [4], [5], [6] generate recommendations via capturing user general preferences and sequential behavior patterns. In other words, the user long-term preference module and the sequential behavior module are two major components for the sequential recommendation algorithms. However, the traditional sequential recommendations (i.e., FPMC [4] and PRME [5]) neglect the correlation between two components or consider the correlation in the manner of tightly coupling (i.e., HRM [6]). On the other hand, TransRec [7] and ATM [8] employ the operations of personalized translation to model the third-order relationships, and explicitly take the correlation between the above two components in a unified framework. Their experimental results indicate that such a personalized translation operation is beneficial for sequential recommendations.

Recently, deep learning techniques (i.e., GNNs [9] and Transformers [10]) are widely adopted by sequential recommendation models. A GNN model is able to endow the recommendation models with the capability of capturing high-order collaborative signals. For instance, SURGE [11] and HGNN [12] reconstruct the item–item interest graphs according to the item sequences, and utilize the GNN and HGNN to extract the users' interests, respectively. However, for GNN-based sequential recommendation methods, the process of generating the item-item interest graphs for each user is time-consuming, and may distort the original sequential relationships. Moreover, since a Transformer model is able to effectively capture sequential behavior patterns, some researchers proposed several sequential recommendation models built on Transformers. For example, SASRec [13] and TiSARec [14] employ the Transformer module to capture user's dynamic interest, boosting the performance of sequential recommendations. However, Transformer-based sequential recommendation models ignore modeling the users' preferences, and the optimization of Transformer networks is time-consuming. Compared to GNNs, Transformers are more suitable for the sequential recommendation task due to their intrinsic properties [15], [16], [17].

Contrastive learning [18], [19], [20] extracts self-supervised signals from raw data, which alleviates the problem of data sparsity. Inspired by contrastive learning, some researchers [16], [17], [21] have proposed several contrastive learning-based sequential recommendation models.

Received 18 November 2024; accepted 26 February 2025. Date of publication 26 March 2025; date of current version 19 May 2025. This work was supported in part by the Qing Lan Project of Jiangsu Province; in part by the Future Network Scientific Research Fund Project under Grant FNSRFP-2021-YB-54; in part by the Tongda College of Nanjing University of Posts and Telecommunications under Grant XK203XZ21001; and in part by the Chunhui Plan Collaborative Research Project, Ministry of Education, China, under Grant HZKY20220350. This article was recommended by Associate Editor L. C. Rego. *(Corresponding author: Yonghong Yu.)*

Aoran Zhang and Yonghong Yu are with the College of Tongda, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: JUSTzhangaoran@gmail.com; yuyh@njupt.edu.cn).

Li Zhang is with the Department of Computer Science, Royal Holloway University of London, TW20 0EK Egham, U.K. (e-mail: Li.Zhang@rhul.ac.uk).

Rong Gao is with the School of Computer Science, Hubei University of Technology, Wuhan 430068, China (e-mail: gaorong@hbut.edu.cn).

Hongzhi Yin is with the School of Electrical Engineering and Computer Science, University of Queensland, Brisbane, QLD 4072, Australia (e-mail: h.yin1@uq.edu.au).

Digital Object Identifier 10.1109/TSMC.2025.3550701

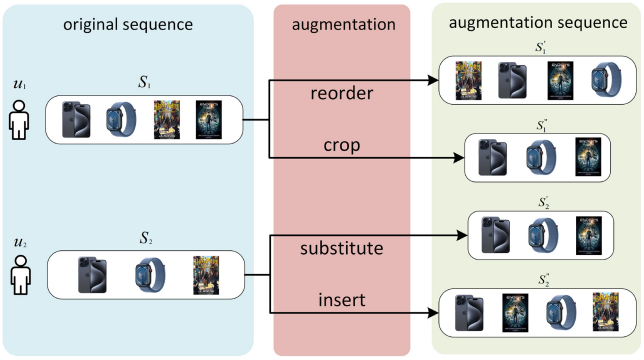


Fig. 1. Illustration of augmentation operations of reorder, crop, substitute and insert.

For instance, CL4SRec [21] explores the usage of the contrastive learning into the Transformer-based sequential recommendation model. Liu et al. [16] proposed CoSeRec. Their work utilized substitute and inserted operations to generate robust augmented sequences. Although their experimental results have demonstrated the effectiveness of contrastive learning for sequential recommendations, their schemes of generating augmented sequences have the following issues.

1) These augmentation schemes change the sequential pattern of item sequences and destroy the underlying semantics of item sequences.
2) These augmentation schemes introduce false negative samples when they perform contrastive learning.

As shown in Fig. 1, the original item sequence S_1 of user u_1 is {iPhone, iWatch, Harry Potter, Ender's Game}, and the original item sequence S_2 of user u_2 is {iPhone, iWatch, Harry Potter}. After augmenting S_1 by cropping and reordering, we have two augmented sequences of S_1, i.e., S_1' ={Harry Potter, iPhone, Ender's Game, iWatch} and S_1'' ={iPhone, iWatch, Ender's Game}. In addition, the augmented sequences of S_2, i.e., S_2' ={iPhone, iWatch, Ender's Game} and S_2'' ={iPhone, Ender's Game, iWatch, Harry Potter}, are generated by substitute and insert operations, respectively. As shown in S_1, the main reason of user u_1 choosing iWatch is that he/she just purchased an iPhone. In other words, the user's purchasing decision largely depends on the previously purchased items, implying that temporal patterns of item sequences are crucial for sequential recommendations. However, if the augmentation views are generated by crop, reorder, substitute and insert operations, the sequential patterns of item sequences may be dramatically changed, destroying the underlying semantics of the original item sequences. For example, the augmented sequence S_1' is obtained by reordering the original sequence S_1, in which we observe that the purchasing for iWatch depends on the previous purchasing behavior, i.e., user u_1 just purchased Harry Potter. Moreover, S_2 adopts the insert operation to insert Ender's Game before iWatch to generate S_2'', making the user's purchasing behavior of the iWatch relies on previously Ender's Game. The semantics of these augmented item sequences greatly deviate from that of original sequence, i.e., augmentation operations destroy the sequential pattern of original item sequences. In addition, these augmentation operations introduce some false negative samples, resulting in difficulty in accurately defining positive and negative samples. For instance, although S_1' and S_1'' are generated by different augmentation operations, S_1' and S_1'' should be a positive pair because they are derived from the same item sequence. Meanwhile, since S_1' and S_2' are derived from different item sequences, S_1' and S_2' should be a negative pair. However, as shown in Fig. 1, S_1' is the same as S_2', which is contradictory with the underlying principle of augmentation operations. Hence, it is hard for traditional contrastive learning-based sequential recommendation models to distinguish positive samples from negative samples.

In addition, most of existing contrastive learning-based sequential recommendation algorithms [16], [17], [21] utilize the InfoNCE loss with static temperature to define their respective objective functions. Based on the principle of InfoMax, the InfoNCE loss treats two augmentations of the same instance as positive pairs, and all others as negative ones. The temperature is a important parameter for InfoNCE loss, which enable models achieve the goal of instance discrimination or group-wise discrimination. As reported by [22] and [23], InfoNCE loss with a smaller value of temperature is able to achieve instance discrimination. On the contrary, if we expect to reach the goal of group-wise discrimination, InfoNCE loss needs a higher value of temperature. In fact, the main aim of group-wise discrimination is to discover the differences among clusters of samples. Particularly, for contrastive learning-based recommendation models, it is important to simultaneously capture the differences among individual users/items and among the clusters of users/items. Therefore, it is unreasonable for contrastive learning-based recommendation models to set temperature as a fixed hyperparameter, which makes the recommendation models lack the capability to simultaneously achieve group-wise discrimination and instance discrimination.

To tackle the above issues, we propose a contrastive translation-based sequential recommendation algorithm, namely, CT4Rec. Specifically, CT4Rec generates augmented views of item sequences by injecting noises into embeddings of users and items, which guarantees that the underlying semantics of augmented views are consistent with those of the original item sequence. In addition, we utilize a dynamic temperature strategy in contrastive loss, which enables CT4Rec to adaptively achieve group-wise discrimination and instance discrimination. Finally, we employ the personalized translation operations to model the third-order relationships and learn the long-term preferences of user and sequential patterns in a unified framework. We summarizes the main contributions of this article as follows.

1) We utilize a representation augmentation approach to generate the augmented views of item sequences, which effectively guarantees that the underlying semantics of augmented views are consistent with those of original item sequence and alleviates the problem of false negative samples caused by inappropriate augmentation strategies.
2) We employ the dynamic temperature strategy to dynamically adjust the temperature parameter of the contrastive loss, which enables CT4Rec to adaptively achieve group-wise discrimination and instance discrimination.
3) Extensive experiments are conducted on five benchmark datasets and the experimental results verify the superior

performance of our proposed CT4Rec compared with traditional sequential recommendation models.

II. Related Work

In this section, we briefly review traditional sequential recommendation, deep learning-based recommendation and contrastive learning-based recommendation algorithms.

A. Traditional Sequential Recommendation

The key of sequential recommendation is to learn user preferences and capture sequential behavior patterns. For instance, FPMC [4] utilizes a first-order Markov chain to model sequential behaviors and learns user long-term preferences via matrix factorization. To enhance the generalization capability of FPMC, Feng et al. [5] proposed PRME. PRME measures the similarities between different POIs via computing the Euclidean distance. To capture the correlation between two components, i.e., the user long-term preference modeling and the sequential behavior modeling, Wang et al. [6] proposed HRM. HRM introduces aggregation operations of pooling to model complex interaction behaviors in sequences. However, the classic sequential recommendation models (i.e., FPMC and PRME) do not take the correlation between two components into account or implicitly consider the correlation (i.e., HRM). Furthermore, He et al. [7] proposed TransRec, which explicitly implements the user long-term preferences modeling and sequential patterns modeling in a unified framework. Unlike TransRec that only considers the impact of low-order interaction information, Wu et al. [8] proposed ATM, which utilizes high-order Markov chains to capture high-order interaction information.

B. Deep Learning-Based Recommendation

Since a GNN model is able to endow the recommendation models with the capability of capturing high-order collaborative signal, it is widely adopted in the field of recommendation system. For GNN-based sequential recommendations, Chang et al. [11] proposed sequential recommendation with GNN, namely, SURGE. SURGE reconstructs the item–item interest graphs according to the item sequence, and uses the GNN to extract the users' interests. Moreover, since the GNN only models the item sequences as a flat graph without hierarchy, Xue et al. [12] utilized a hierarchical GNN to model factorial user preferences. In addition, Guo et al. [24] proposed TiDA-GCN, which utilizes a domain-aware GNN to enrich the embeddings of entities. However, for GNN-based methods, the process of generating the item–item interest graphs for each user is time-consuming, and may distort the sequential patterns hidden in original item sequences.

For GNN-based session recommendations, SR-GNN [25] utilizes the GNN to capture complex transitions of items among the session graph. However, traditional session-based recommendations are restricted by the session of current user. To address this issue, Pang et al. [26] proposed HG-GNN, utilizing heterogeneous global GNNs to learn user preferences from the current and historical sessions. In addition, Yin et al. [27] proposed H3GNN. H3GNN utilizes hybrid hierarchical hypergraph neural network to effectively capture the hierarchical information and sequential relationships among sessions.

Inspired by the great success of Transformers [10], several Transformers-based sequential recommendations have been proposed. For example, SASRec [13] captures user's dynamic interests via self-attention module. To tackle the limitation of unidirectional structure of SASRec in capturing user behavior sequences, BERT4Rec [15] adopts a deep bidirectional self-attention module to model user behavior sequences. Moreover, Li et al. [14] proposed TiSASRec. TiSASRec utilizes a time interval aware self-attention mechanism to model the relative time intervals and absolute positions among items. The above Transformer-based sequential recommendation algorithms basically capture the sequential patterns of users by utilizing self-attention mechanisms. However, the Transformer-based methods tend to have high computational complexity, which hinders their applications in real-world scenarios. Our proposed method utilizes a personalized translation operation to model the sequential patterns of users, which avoids the additional computational cost of optimizing Transformers.

C. Contrastive Learning-Based Recommendation

Contrastive learning has received widely attention in computer vision [28], [29] and natural language processing [30], [31]. For recommendation systems, contrastive learning is a promising solution to data sparsity since it is able to extract self-supervised signals from raw data. Recently, researchers have explored the application of contrastive learning in the field of recommendation [18], [20], [21]. For item recommendations, Wu et al. [18] proposed SGL. However, SGL generates the adjacency matrix in each training epoch, which is a time-consuming process for generating contrastive views. To enhance the consistency of representations, Yu et al. [20] proposed SimGCL. To alleviate popularity bias, Liu et al. [32] proposed PopDCL, which adaptively corrects the positive and negative scores via the users' and items' popularities.

For session recommendations, Wang et al. [33] proposed STGCR, which employs the temporal graph modeling method to capture the dynamic users' global preference. Moreover, to filter out false negative samples, Liu et al. [34] proposed SCLRec. Additionally, Wan et al. proposed RESTC [35], which utilizes an auxiliary cross-view contrastive learning mechanism to enhance the learning process of session-based recommendation models.

For sequential recommendations, Zhang et al. [36] proposed GCL4SR. GCL4SR utilizes a weighted item transition graph to augment the representation of each interaction sequence and employ the contrastive loss to learn the consistency among augmented views. In addition, in order to disentangle of user real interest and address popularity bias, Yang et al. [37] proposed DCRec. Moreover, Liu et al. [38] proposed SelfGNN, which utilizes the GNN to learn the short-term and long-term collaborative signals and employs a personalized self-augmented

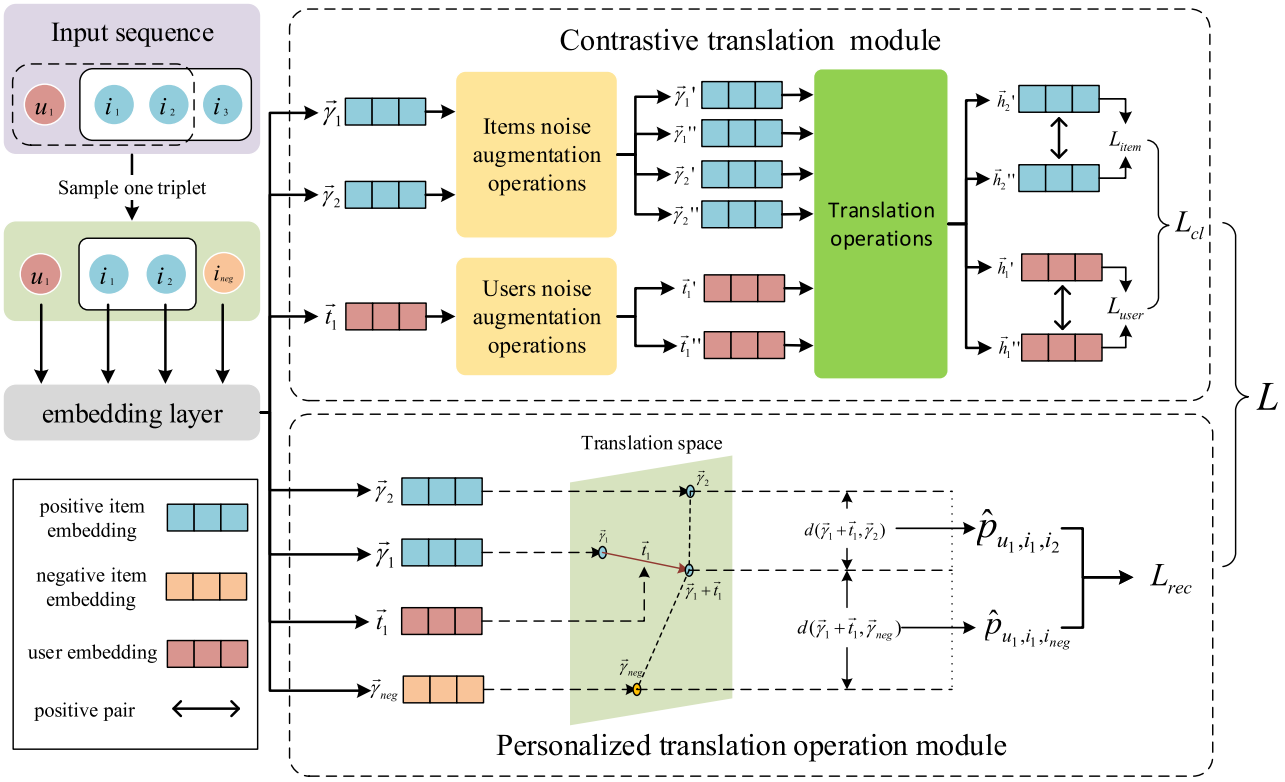


Fig. 2. Framework for CT4Rec.

learning structure to enhance model robustness. Xie et al. [21] proposed CL4SRec, exploring the usage of the contrastive learning into the Transformer-based sequential recommendation model. In addition, CoSeRec [16] utilizes a substitute and insert operation to generate robust augmented sequences. Moreover, ICLRec [17] injects the learned user intentions into contrastive learning-based sequential recommendation. Considering the high semantic similarity between the embeddings of items learned by the Transformer-based sequential recommendation algorithms, Qiu et al. [39] designed a contrastive regularization method to reshape the distribution of sequence representations. However, existing contrastive learning-based sequential recommendation algorithms generally adopt inappropriate augmentation strategies for generating augmented views, destroying the underlying semantics of item sequences and introducing false negative samples. Different from contrastive learning-based sequential recommendation methods, we adopt a noise augmentation scheme to generate augmented views, which is able to guarantee that the underlying semantics of augmented views are consistent with those of the original item sequence. Moreover, instead of keeping the parameter temperature unchanged, we utilize a dynamic temperature strategy to enhance our proposed model, which equips it with the capabilities of group-wise discrimination and instance discrimination.

III. Our Proposed Method

In this section, we describe the contrastive translation based sequential recommendation model in detail. Our proposed CT4Rec is mainly divided into two modules: 1) a personalized translation operation module and 2) a contrastive translation module. The major goals of personalized translation operation module are modeling user long-term preferences and capturing sequential patterns hidden in item sequences visited by users. Inspired by knowledge graph embedding [40], [41], [42], we treat each item as an entity node of knowledge graph and each user as a personalized relation between two items. Then, we utilize personalized translation operations to model the third-order interaction relationships. In contrastive translation module, extra supervised signals are extracted via contrastive learning, which are utilized to supplement the sparse user interaction behaviors. Specifically, we generate augmented views of item sequences by injecting noises into embeddings of users and items, which keeps underlying semantics of item sequences unchanged. In addition, we utilize the InfoNCE loss to learn the sequential patterns of item sequences, and adopt dynamic temperature strategy to enhance CT4Rec, which endows CT4Rec with the capability of group-wise discrimination and instance discrimination. Fig. 2 presents the the framework of CT4Rec.

A. Problem Description

In sequential recommendation systems, items that are visited by a user $u \in U$ in chronological order are represented as a sequence S^u, where U is the set of users. The set of action sequences of all users is denoted as $A = \{S^1, S^2, \ldots, S^{|U|}\}$. In addition, V is the set of items. The goal of sequential recommendation is to predict the probabilities of user u visiting candidate items and recommend the top-N items with the highest probabilities to user u.

B. Personalized Translation Operation Module

In knowledge graph embedding, if a triple $< h, r, t >$ holds, the distance between the translated representation of the h via the latent representation of r and the latent representation of the t should be close, i.e., $\overrightarrow{h} + \overrightarrow{r} \approx \overrightarrow{t}$. Inspired by the knowledge graph [7], [8], [40], we use the personalized translation operation to model third-order relationship when the user u visited item i and then visited items j. Formally

$$\vec{\gamma}_i + \vec{t}_u \approx \vec{\gamma}_j \tag{1}$$

where $\vec{t}_u$ is the translation vector of the user u in the transition space Ω. And the embeddings of item i and item j are presented as $\vec{\gamma}_i$ and $\vec{\gamma}_j$, respectively. However, it is difficult for recommendation models to accurately learn personalized user preferences due to the lack of user interactions. Therefore, the user's translation vector are rewritten as

$$\vec{T}_u = \vec{t} + \vec{t}_u \tag{2}$$

where $\vec{t}$ denotes the average behaviors of all users. $\vec{t}_u$ represents the user-specific offset. Hence, (1) is rewritten as

$$\vec{\gamma}_i + \vec{T}_u \approx \vec{\gamma}_j. \tag{3}$$

The probability of user successively visiting two items is defined as

$$\begin{aligned} P(j|u,i) &\propto \beta_j - d\left(\vec{\gamma}_i + \vec{T}_u, \vec{\gamma}_j\right) \\ \vec{\gamma}_j &\in \Psi \in \Omega, i \text{ and } j \in V, u \in U \end{aligned} \tag{4}$$

where Ψ is a subspace of Ω, and β_j represents the popularity of item j. $d(\cdot,\cdot)$ indicates the Euclidean distance.

C. Contrastive Translation Module

1) Augmentation of Item Sequences: The augmentation schemes employed by traditional contrastive learning-based sequential recommendation algorithms destroy the underlying semantics of original item sequences and introduce the false negative samples, resulting in difficulty in accurately defining positive and negative samples. Unlike the traditional contrastive learning-based sequential recommendations [16], [17], [21], we generate augmented views of item sequences by injecting noises into embeddings of users and items, formalised as follows:

$$\begin{aligned} \vec{\gamma}_i + \Delta_i' &= \vec{\gamma}_i', \ \vec{\gamma}_i + \Delta_i'' = \vec{\gamma}_i'' \\ \vec{\gamma}_j + \Delta_j' &= \vec{\gamma}_j', \ \vec{\gamma}_j + \Delta_j'' = \vec{\gamma}_j'' \\ \vec{t}_u + \Delta_u' &= \vec{t}_u', \ \vec{t}_u + \Delta_u'' = \vec{t}_u'' \end{aligned} \tag{5}$$

where $\Delta_i', \Delta_i'', \Delta_j', \Delta_j'', \Delta_u', \Delta_u''$ are noise vectors, and $||\Delta||_2 = \epsilon$, $\Delta = \overline{\Delta} \odot \text{sign}(\vec{\gamma}_i)$ or $\overline{\Delta} \odot \text{sign}(\vec{\gamma}_j)$ or $\overline{\Delta} \odot \text{sign}(\vec{t}_u)$. ε controls the magnitude of noise Δ, and Δ is regarded as a point on a hypersphere with radius ε. $\overline{\Delta}$ follows a uniform distribution of interval (0, 1).

As shown in Fig. 3, when injecting noises into the embeddings of the original user u and item j, we obtain two augmented views $\vec{\gamma}_i'$ and $\vec{\gamma}_i''$ for item $\vec{\gamma}_i$ as well as two augmented views $\vec{t}_u'$ and $\vec{t}_u''$ for user $\vec{t}_u$. Each augmentation

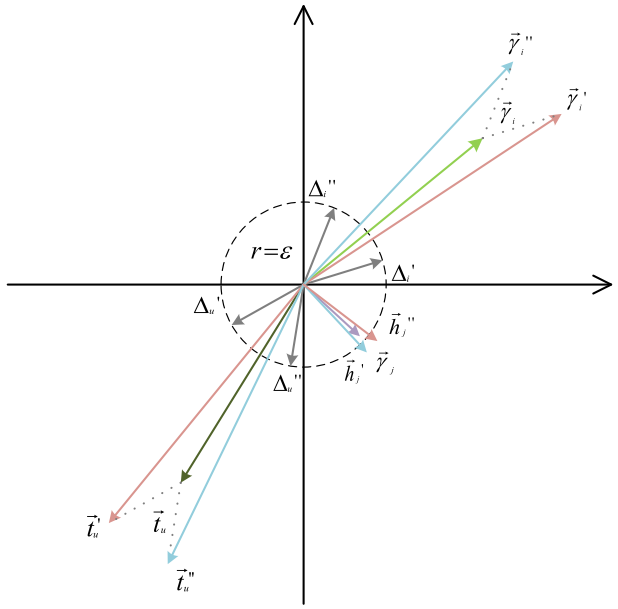


Fig. 3. Brief illustration of noise augmentation operation.

may be represented as a small angle rotation around the original vector, which preserves most of the original underlying semantics of original item sequences.

Similar to the translation operation module in Section III-B, we perform a personalized translation operation on the augmented representation $\vec{\gamma}_i', \vec{\gamma}_i'', \vec{\gamma}_j', \vec{\gamma}_j''$, as illustrated as follows:

$$\begin{aligned} \vec{\gamma}_i' + \vec{t} + \vec{t}_u' = \vec{h}_j', \ \vec{\gamma}_i'' + \vec{t} + \vec{t}_u'' = \vec{h}_j'' \\ \vec{\gamma}_j' - \vec{\gamma}_i' - \vec{t} = \vec{h}_u', \ \vec{\gamma}_j'' - \vec{\gamma}_i'' - \vec{t} = \vec{h}_u'' \end{aligned} \tag{6}$$

where $\vec{h}_j'$ and $\vec{h}_j''$ indicate two augmented representations of item j via translation operation, and $\vec{h}_u'$ and $\vec{h}_u''$ represent two augmented representations of user u. As shown in Fig. 3, although $\vec{h}_j'$ and $\vec{h}_j''$ are obtained by translation operations, the third-order relationship among (u, i, j) is still held, i.e., the differences between $\vec{\gamma}_j$ and $\vec{h}_j'/\vec{h}_j''$ are relatively small. In fact, similar to the encoders used by graph contrastive learning (e.g, GNN [9]) and visual contrastive learning (e.g, ResNet [43]), the translation operation can be treated as an encoder for contrastive learning-based sequential recommendation model.

2) Contrastive Loss: We define the contrastive loss based on the InfoNCE loss. Specifically, we treat two augmentations of the same item sequence as positive pairs and the augmentations of different item sequences as negatives. The contrastive loss of CT4Rec L_{cl} is defined as

$$\begin{aligned} L_{cl} = &\sum_{u \in U} -\ln \frac{e^{\text{sim}\left(\vec{h}_u', \vec{h}_u''\right)/\tau}}{e^{\text{sim}\left(\vec{h}_u', \vec{h}_u''\right)/\tau} + \sum_{g \in U \cap u \neq g} e^{\text{sim}\left(\vec{h}_u', \vec{\gamma}_g''\right)/\tau}} \\ &+ \sum_{j \in V} -\ln \frac{e^{\text{sim}\left(\vec{h}_j', \vec{h}_j''\right)/\tau}}{e^{\text{sim}\left(\vec{h}_j', \vec{h}_j''\right)/\tau} + \sum_{v \in U \cap j \neq v} e^{\text{sim}\left(\vec{h}_j', \vec{\gamma}_v''\right)/\tau}} \end{aligned} \tag{7}$$

where $\text{sim}(\cdot,\cdot)$ is the cosine similarity and τ denotes the temperature parameter. In L_{cl}, the first term computes the contrastive loss from the perspective of users, and the second

term calculates the contrastive loss from the side of items. Since the personalized translation operation module could not accurately learn user long-term preferences and sequential patterns with limited interaction data, the contrastive loss module provides extra supervised signals for personalized translation operation module via learning invariances across the augmentations derived from the original item sequence.

3) Dynamic Temperature Strategy: As an important parameter in InfoNCE, τ greatly affects the training process of L_{cl}. For instance, the derivative of l_{cl}^{u} for the user u with respect to $\vec{t}_g''$ is computed as follows:

$$\begin{aligned}\frac{\partial l_{cl}^{u}}{\partial \vec{t}_g''} &= \frac{\partial l_{cl}^{u}}{\partial \mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)} \cdot \frac{\partial \mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)}{\partial \vec{t}_g''} = \left(\frac{\partial - \ln\left(e^{\mathrm{sim}(\vec{h}_u', \vec{h}_u'')/\tau}\right)}{\partial \mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)} \right. \\ &\quad \left. + \frac{\partial \ln\left(e^{\mathrm{sim}\left(\vec{h}_u', \vec{h}_u''\right)/\tau} + \sum_{g \in U \cap u \neq g} e^{\mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)/\tau}\right)}{\partial \mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)} \right) \\ &\quad \cdot \frac{\partial \mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)}{\partial \vec{t}_g''} \\ &= \frac{\frac{\partial \sum_{g \in U \cap u \neq g} e^{\mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)/\tau}}{\partial \mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)}}{e^{\mathrm{sim}\left(\vec{h}_u', \vec{h}_u''\right)/\tau} + \sum_{g \in U \cap u \neq g} e^{\mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)/\tau}} \cdot \frac{\partial \mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)}{\partial \vec{t}_g''} \\ &= \frac{1}{\tau} \cdot \frac{e^{\mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)/\tau}}{e^{\mathrm{sim}\left(\vec{h}_u', \vec{h}_u''\right)/\tau} + \sum_{g \in U \cap u \neq g} e^{\mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)/\tau}} \cdot \frac{\partial \mathrm{sim}\left(\vec{h}_u', \vec{t}_g''\right)}{\partial \vec{t}_g''}. \end{aligned} \tag{8}$$

When τ is assigned with a small value, the contribution of the similarities between $\vec{h}_u'$ and $\vec{t}_g''$ to $(\partial L_{cl}/\partial \vec{t}_g'')$ is exponential, and the gradient of $\vec{t}_g''$ is led by the similarities between $\vec{h}_u'$ and $\vec{t}_g''$. Essentially, a small value of τ is able to amplify the impact of similarity between $\vec{h}_u'$ and $\vec{t}_g''$ on gradients and make the model focused more on sight differences among entities. Thus, the contrastive learning-based sequential recommendation models with a small τ have the capability of instance discrimination. Existing contrastive learning methods with InfoNCE as the objective function usually adopt a smaller value of τ. Hence, they generally are equipped with the capability of instance discrimination. In contrast, when τ is set with a large value, the differences between $\vec{h}_u'$ and $\vec{t}_g''$ will be weakened and the similar pairs are more likely to be clustered. If the similar pairs are tightly clustered, the differences among the clusters of entities will be highlighted, which would endow the sequential recommendation model with the capability of group-wise discrimination. However, existing contrastive learning-based sequential recommendation models usually assign a small value to parameter τ, resulting in insufficient capability of group-wise discrimination. In our proposed dynamic temperature strategy, the recommendation models are able to dynamically tune the values of τ based on the similarity $\mathrm{sim}(x, y)$ between x and y, formalized as

$$\tau_{xy} = \tau_- + \frac{(\tau_+ - \tau_-)}{2} \times \left(1 + \cos\left(\frac{\pi}{2}(1 + \mathrm{sim}(x, y))\right)\right) \tag{9}$$

where τ_{xy} represents the corresponding value of τ for pair (x, y). τ_+ and τ_- indicate the upper and lower bounds of τ, respectively. Assuming that (x, y) is a negative pair, the similarity $\mathrm{sim}(x, y)$ tends to -1, i.e., τ_{xy} is toward to τ_+, which indicates that the recommendation model should focus on the differences among their clusters (i.e., group-wise discrimination) for negative pairs. By contrast, if (x, y) is a positive pair, the similarity $\mathrm{sim}(x, y)$ is close to 1 and τ_{xy} is approaching τ_-, indicating that the recommendation model pays more attention to the differences between entities (i.e., instance discrimination). In our proposed CT4Rec, we employ the dynamic temperature strategy to simultaneously capture the differences among individual users/items and among the clusters of users/items, which is able to balance the abilities of group-wise discrimination and instance discrimination.

D. Objective Function

We employ sequential Bayesian personalized ranking (S-BPR) as the loss function of recommendation task. Formally, the objective function of recommendation task is defined as follows:

$$\begin{aligned} L_{\mathrm{rec}} &= -\ln \prod_{u \in U} \prod_{j \in S^u} \prod_{j' \in V \setminus j} P\left(j >_{u,i} j' | \Theta\right) P(\Theta) \\ &= \sum_{u \in U} \sum_{j \in S^u} \sum_{j' \in V \setminus j} -\ln \sigma\left(\widehat{p}_{u,i,j} - \widehat{p}_{u,i,j'}\right) + \lambda_\Theta \|\Theta\|_F^2 \end{aligned} \tag{10}$$

where $P(j >_{u,i} j' | \Theta)$ denotes the probability that user u prefers item j over item j' after he/she has visited item i, and Θ is the set of model parameters. $\widehat{p}_{u,i,j} = P(j|u, i)$ indicates the probability of user u successively visiting two items i and j. $\sigma(x) = (1/[1 + e^{-x}])$ is the sigmoid function, λ_Θ is the regularization coefficient.

Similar to works [18], [20], [21], we jointly optimize classical recommendation task L_{rec} and contrastive learning task L_{cl} via a multitask training strategy. Thus, the objective function L of CT4Rec is formalized as

$$L = L_{\mathrm{rec}} + \lambda_{cl} L_{cl} \tag{11}$$

where λ_{cl} weights the contrastive loss in the objective function of CT4Rec.

E. Complexity Analysis of CT4Rec

In this section, we analysis the space and time complexity of CT4Rec. First, we compute the space complexity of CT4Rec. As shown in (10), the model parameters of CT4Rec are $\Theta = \{\vec{t}_u, \vec{\gamma}_i, \beta_i, \vec{t}\}$. Hence, the space complexity of CT4Rec is $O(K \cdot (|U| + |V|) + |V| + K) \approx O(K \cdot (|U| + |V|))$, where K is the embedding size. In fact, the space complexity of CT4Rec grows linearly with $|U|$, $|V|$ and K.

Second, we analyse the time complexity of CT4Rec. Specifically, CT4Rec contains two important modules, i.e., the personalized translation operation module and the contrastive translation module. For the personalized translation operation module, it computes the Euclidean distance between two items, thus the time complexity of the personalized translation operation module with the training batch size B is $O(B \cdot K)$. For the contrastive translation module, there are three important operations, the noise injection operation, the operation of contrastive translation and the calculation of InfoNCE loss. For the operation of noise injection, we need to generate

TABLE I
STATISTICS OF FIVE DATASETS

Dataset	$\#users$	$\#items$	$\#interactions$	$sparsity$
Automotive	1,951	10,860	27,592	99.8698%
Instruments	2,273	22,294	39,804	99.9215%
Cell_Phones	9,534	53,479	139,141	99.9727%
Tools	10,076	66,710	169,245	99.9748%
Clothing	41,878	355,571	678,329	99.9954%

the user-specify noise tensors and item-specify noise tensors, whose sizes are consistent with the embeddings of users and items, respectively. Then, we add the noise tensors into the corresponding embedding matrices, and the time complexity of noise injection operation is $O(B\cdot|U|\cdot K+B\cdot|V|\cdot K)$. For the operation of contrastive translation, CT4Rec computes 4 times personalized translation operations, thus the time complexity of contrastive translation operation is $O(4\cdot B\cdot K)$. In addition, the time complexity of the calculation of InfoNCE loss is $O(B\cdot|U|\cdot K+B\cdot|V|\cdot K)$. Hence, the total time complexity of contrastive translation module is $O(B\cdot|U|\cdot K+B\cdot|V|\cdot K+4\cdot B\cdot K+B\cdot|U|\cdot K+B\cdot|V|\cdot K)\approx O(2\cdot B\cdot K\cdot(|U|+|V|))$. It is clear that the time complexity of contrastive translation module depends on U and V. Then, we employ the dynamic temperature strategy to adaptively adjust temperature τ. According to (9), the time complexity of the dynamic temperature strategy is $O(B\cdot(|U|+|V|))$. As a result, the total time complexity of CT4Rec is $O(2\cdot B\cdot K\cdot(|U|+|V|)+B\cdot K+B\cdot(|U|+|V|))\approx O(B\cdot K\cdot(|U|+|V|))$. Since the embedding size K is limited with $K << \min(|U|,|V|)$, our proposed method is able to scale to large-scale datasets.

IV. EXPERIMENT

In this section, extensive experiments are conducted on five benchmark datasets to assess the effectiveness of CT4Rec.

A. Datasets

In this section, we choose five datasets from Amazon[1] (i.e., Clothing, Tools, Cell_Phones, Instruments and Automotive) to evaluate the effectiveness of CT4Rec. For all datasets, we filter out the users and items that have less than 10 interactions. Table I summarizes the statistics of datasets.

B. Evaluation Metrics and Experimental Setting

We choose two widely used rank-oriented metrics, i.e., Recall@N and NDCG@N, to evaluate the performance of all compared methods, because sequential recommendation essentially is a ranking problem. For both metrics, we set the length of recommendation list $N = 10, 50$.

We choose the following sequential recommendation algorithms for comparison.

1) *BPRMF [2]:* BPRMF only focuses on the long-term preferences of users, ignoring the sequential behaviors patterns.
2) *TransRec [7]:* TransRec employs personalized translation operations to model third-order relationships.
3) *Caser [44]:* Caser utilizes horizontal and vertical convolution operations to captures sequential behaviors patterns.
4) *SASRec [13]:* SASRec is a unidirectional self-attention model, which captures user's dynamic interests via a self-attention module.
5) *CL4SRec [21]:* CL4SRec generates augmented views by utilizing crop, mask and reorder operations. However, those augmentation schemes destroy the underlying semantics of item sequences, resulting in suboptimal performance of recommendation.
6) *CoSeRec [16]:* CoSeRec utilizes substitute and insert operations to generate robust augmented sequences, which alleviates the problem of skewness in the length distributions of item sequences.
7) *ICLRec [17]:* ICLRec captures the user potential intentions via the clustering method and injects the user intentions into contrastive learning-based sequential recommendation.
8) *CT4Rec/DT:* CT4Rec/DT is the simplified version of our proposed method, which removes the dynamic temperature strategy from CT4Rec.
9) *CT4Rec:* CT4Rec integrates a dynamic temperature strategy into CT4Rec/DT, endowing CT4Rec with the capabilities of group-wise discrimination and instance discrimination.

The parameters of each model are set based on our experiments or their original configurations. For all models, the learning rate η is chosen among $\{0.001, 0.005, 0.01, 0.05, 0.1\}$ and the embedding size K is set to 64. In addition, λ_Θ is tuned within $\{0, 0.00001, 0.0001, 0.001, 0.01\}$. For Transformer-based sequential recommendations, we fix both the numbers of layers and heads at 2. For Caser, the hyperparameters are set as the same configurations of its original studies.

For CL4SRec, CoSeRec, ICLRec and CT4Rec, λ_{cl} varies in $\{0.001, 0.01, 0.1, 1, 10\}$. In addition, the historical behaviors sequence $S^u = \{S_1^u, S_2^u, \ldots, S_{|S^u|}^u\}$ of each user u is divided into two parts:

1) The last interacted item $S_{|S^u|}^u$, which is used as the test set;
2) The remaining items, which are used as the training set.

Moreover, we choose Adam as the optimizer.

C. Performance Analysis

Performance comparisons are presented in Table II, we have the following main observations.

1) BPRMF has the worst performance. This is because BPRMF only learns the user long-term preferences when generating recommendation items, ignoring the sequential behavior patterns.
2) Compared to the nonsequential model (i.e., BPRMF), sequential recommendation models (i.e., TransRec, Caser and SASRec) achieve better performance. This indicates that capturing sequential patterns is crucial for improving the performance of sequential recommendations.

[1]https://nijianmo.github.io/amazon/index.html

TABLE II
PERFORMANCE COMPARISON

Dataset	Metric	BPRMF	TransRec	Caser	SASRec	CL4SRec	CoSeRec	ICLRec	CT4Rec/DT	CT4Rec
Automotive	$Recall@10$	0.0200	0.0338	0.0267	0.0266	0.0246	0.0276	0.0308	0.0400	**0.0410**
	$NDCG@10$	0.0102	0.0157	0.0132	0.0131	0.0148	0.0132	0.0165	0.0197	**0.0217**
	$Recall@50$	0.0395	0.0855	0.0579	0.0615	0.0648	0.0502	0.0794	0.1031	**0.1051**
	$NDCG@50$	0.0123	0.0266	0.0190	0.0207	0.0210	0.0170	0.0271	0.0339	**0.0347**
Instruments	$Recall@10$	0.0142	0.0189	0.0167	0.0154	0.0180	0.0189	0.0199	0.0255	**0.0264**
	$NDCG@10$	0.0065	0.0086	0.0081	0.0088	0.0097	0.0104	0.0114	0.0120	**0.0127**
	$Recall@50$	0.0370	0.0537	0.0405	0.0466	0.0502	0.0534	0.0589	0.0717	**0.0766**
	$NDCG@50$	0.0114	0.0158	0.0129	0.0158	0.0169	0.0179	0.0196	0.0216	**0.0232**
Cell_Phones	$Recall@10$	0.0149	0.0197	0.0187	0.0264	0.0220	0.0281	0.0269	0.0415	**0.0444**
	$NDCG@10$	0.0077	0.0098	0.0094	0.0119	0.0107	0.0142	0.0122	0.0201	**0.0222**
	$Recall@50$	0.0393	0.0583	0.0510	0.0645	0.0700	0.0758	0.0686	0.1063	**0.1084**
	$NDCG@50$	0.0127	0.0180	0.0156	0.0201	0.0211	0.0242	0.0218	0.0341	**0.0358**
Tools	$Recall@10$	0.0088	0.0125	0.0095	0.0133	0.0127	0.0130	0.0134	0.0153	**0.0178**
	$NDCG@10$	0.0041	0.0062	0.0056	0.0063	0.0066	0.0066	0.0070	0.0081	**0.0087**
	$Recall@50$	0.0230	0.0369	0.0226	0.0371	0.0374	0.0346	0.0379	0.0470	**0.0501**
	$NDCG@50$	0.0071	0.0113	0.0081	0.0114	0.0118	0.0113	0.0120	0.0147	**0.0154**
Clothing	$Recall@10$	0.0030	0.0038	0.0032	0.0034	0.0035	0.0040	0.0042	0.0049	**0.0070**
	$NDCG@10$	0.0015	0.0019	0.0016	0.0017	0.0017	0.0018	0.0018	0.0022	**0.0028**
	$Recall@50$	0.0089	0.0112	0.0090	0.0092	0.0095	0.0097	0.0121	0.0157	**0.0190**
	$NDCG@50$	0.0027	0.0035	0.0027	0.0029	0.0030	0.0032	0.0038	0.0042	**0.0054**

3) SASRec is superior to Caser. This may be because the self-attention mechanism of SASRec is more effective than the CNN used by Caser in capturing sequential behaviors patterns.
4) CL4SRec generally outperforms SASRec. One possible reason is that CL4SRec utilizes self-supervised signals extracted from raw data via contrastive learning, which effectively alleviates the problem of data sparsity.
5) Compared to CL4SRec, CoSeRec achieves better performance. This observation confirms that the substitute and insert operations used by CoSeRec are more robust than augmentation operations employed by CL4SRec.
6) The performance of ICLRec is better than those of CoSeRec. One of the main reasons is that ICLRec models the potential intentions of users and injects the user intentions into contrastive learning-based sequential recommendation, improving the quality of the latent representations of entities.
7) On all datasets, our proposed contrastive translation-based sequential recommendation method consistently performs better than other methods. For instance, in terms of *Recall*@10 and *NDCG*@10, CT4Rec/DT outperforms ICLRec by 16.67% and 22.22% on the large-scale dataset, i.e., Clothing, respectively. On the medium-scale dataset, i.e., Tools, the respective improvements over ICLRec are 17.54% and 22.58%, respectively. Moreover, on the small-scale dataset, i.e., Automotive, the performance enhancements of CT4Rec/DT are 29.80% and 19.51%, respectively. This observation confirms that the contrastive translation operation is beneficial to traditional translation-based recommendation models.
8) Compared against CT4Rec/DT, CT4Rec adopts the dynamic temperature strategy and further improves the performance. This demonstrates that using the dynamic temperature strategy to adaptively adjust temperature parameters is significant, which endows CT4Rec with the capabilities of group-wise discrimination and instance discrimination.

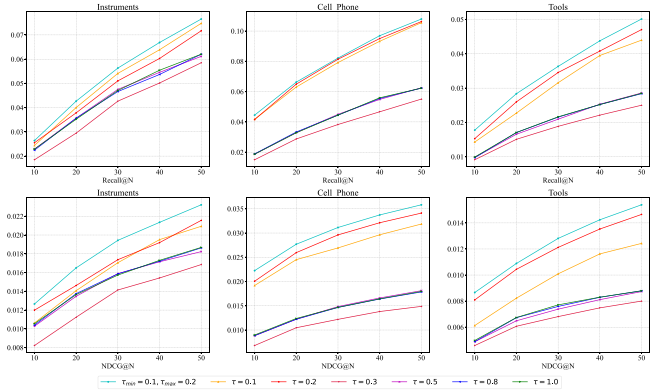


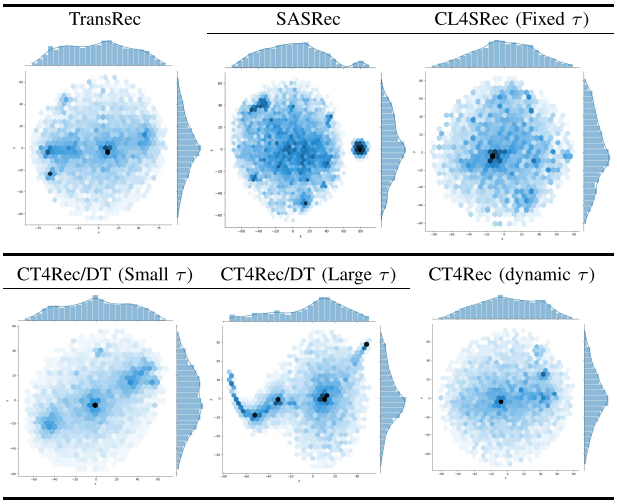
Fig. 4. Impact of different τ.

D. Impact of τ

In CT4Rec, a smaller τ enables the sequential recommendation models learn the differences among individual users/items, i.e., achieving instance discrimination. Moreover, a larger value of τ makes the sequential recommendation models capture the differences among the clusters of users/items, i.e., achieving group-wise discrimination. In this experiment, we investigate how the parameter τ influences the recommendation performance of CT4Rec.

As Fig. 4 shows, CT4Rec/DT is weaker than CT4Rec when $\tau = 0.2$. In addition, as τ gradually increases, the performance of CT4Rec/DT begins to decline. One possible reason is that increasing the value of τ causes the recommendation model to focus on the differences among user/item clusters, which ignores the individual differences among users/items. CT4Rec achieves the best performance when adopting the dynamic temperature strategy. For instance, on the Instruments, the Recall@10 and NDCG@10 of CT4Rec are 0.0264 and 0.0127, which are 3.45% and 5.50% higher than those of CT4Rec/DT

TABLE III
DISTRIBUTION OF ITEM REPRESENTATIONS



with $\tau = 0.2$, respectively. On the Tools, compared to CT4Rec/DT with $\tau = 0.2$, the Recall@10 and NDCG@10 of CT4Rec improve 16.23% and 7.05%, respectively. This once again confirms that adopting the dynamic temperature strategy is able to endow CT4Rec with the capability of group-wise discrimination and instance discrimination.

In addition, we conduct another group of experiments to visualize the differences among the models utilizing the dynamic temperature strategy and the static temperature setting. Specifically, we choose TransRec, SASRec, CL4SRec, CT4Rec/DT and CT4Rec for further analysis, where CL4SRec and CT4Rec/DT adopt the static temperature setting, and CT4Rec utilizes the dynamic temperature strategy. First, we randomly sample 5,000 items from Automotive and project their high-dimensional embeddings into 2-D coordinates by utilizing T-SNE. Then, we draw the heatmaps and the corresponding kernel density map according to the 2-D coordinates. In the heatmaps, the darker the color is, the more points fall in that area. Experimental results are presented in Table III.

As shown in Table III, when CT4Rec/DT adopts the static temperature setting and utilizes a small τ, the distribution of item representations is relatively uniform. One possible reason is that a small τ makes the CT4Rec/DT focus on the differences among individual users/items and leads to the distribution of features relatively uniform. In addition, when τ is large, the item representations are aggregated into different clusters, indicating that the sequential recommendation model pays more attention to the differences among different users/items clusters. Moreover, compared to CT4Rec/DT with a small τ, the distribution of item representations learned by CT4Rec is more uniform. Meanwhile, the representations of items obviously are grouped into several clusters. The main reason is that CT4Rec utilizes the dynamic temperature strategy to adaptively achieve group-wise discrimination and instance discrimination.

E. Impact of Noise Combination

In CT4Rec, we generate augmented views of item sequences by injecting noises into embeddings of users and items.

TABLE IV
PERFORMANCE COMPARISON AMONG DIFFERENT CT4REC VARIANTS

Method	Instruments		Cell_Phones		Tools	
	$Rec@10$	$NDCG@10$	$Rec@10$	$NDCG@10$	$Rec@10$	$NDCG@10$
TransRec	0.0189	0.0086	0.0197	0.0098	0.0125	0.0062
ICLRec	0.0199	0.0114	0.0269	0.0122	0.0134	0.0070
CT4Rec$_{uu}$	0.0264	0.0127	0.0444	0.0222	0.0178	0.0087
CT4Rec$_{ug}$	0.0219	0.0105	0.0420	0.0202	0.0176	0.0082
CT4Rec$_{gg}$	0.0203	0.0098	0.0406	0.0198	0.0168	0.0078

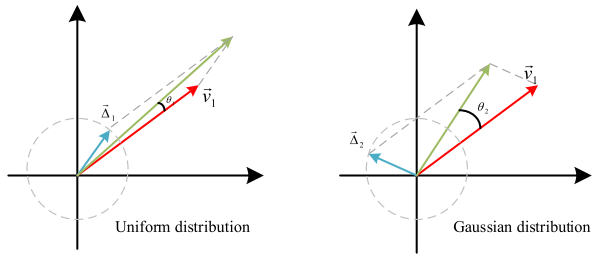


Fig. 5. Brief illustration of the injection process of different noises.

The widely used types of noise include uniform noise and Gaussian noise. In this experiment, we assess the effects of different noise combinations on generating augmented views of item sequences and their impact on the performance of CT4Rec. Specifically, CT4Rec$_{uu}$ or CT4Rec$_{gg}$ indicates that two augmentation operations use the same uniform noise or Gaussian noise to generate two augmented item sequences. In addition, CT4Rec$_{ug}$ represents that one augmented view of item sequence is generated by injecting uniform noise and the other is created by adding Gaussian noise. We conduct experiments on the Instruments, Cell-Phone and Tools.

As shown in Table IV, the performance of CT4Rec$_{uu}$ is superior to that of CT4Rec$_{ug}$, and CT4Rec$_{ug}$ outperforms CT4Rec$_{gg}$. In other words, compared to CT4Rec$_{gg}$, utilizing uniform noise to generate augmented views is more beneficial to CT4Rec. One possible reason is as follows. Since the noises randomly sampled from Gaussian distribution $N(0, 1)$ have negative numbers, the direction of the augmented vector of users or items may be significantly changed, rather than a small angle rotation. The brief illustration of the injection process of different noises is shown in Fig. 5.

From Fig. 5, $\vec{v}_1$ represents the original vector. $\vec{\Delta}_1$ and $\vec{\Delta}_2$ are the noise vectors randomly sampled from uniform distribution and Gaussian distribution, respectively. It should be noted that the direction of $\vec{\Delta}_1$ is basically the same as the original vector $\vec{v}_1$, because all elements of uniform noise are all positive numbers. Different from uniform noise $\vec{\Delta}_1$, the gap between the directions of $\vec{\Delta}_2$ and $\vec{v}_1$ is relatively large. That is because the noises randomly sampled from Gaussian distribution have negative numbers. Therefore, angle θ_2 is much larger than angle θ_1, which indicates that the Gaussian noise is more likely to distort the original semantic relationships compared to the uniform noise. Hence, the noises sampled from uniform distribution are more beneficial to our proposed model.

TABLE V
RESULTS OF ABLATION STUDY

Settings	Automotive		Instruments		Cell_Phones		Tools		Clothing	
	$Rec@10$	$NDCG@10$	$Rec@10$	$NDCG@10$	$Rec@10$	$NDCG@10$	$Rec@10$	$NDCG@10$	$Rec@10$	$NDCG@10$
CT4Rec	0.0410	0.0217	0.0264	0.0127	0.0444	0.0222	0.0178	0.0087	0.0070	0.0028
w/o DT	0.0400	0.0197	0.0255	0.0120	0.0415	0.0201	0.0153	0.0081	0.0049	0.0022
w/o CL	0.0338	0.0157	0.0189	0.0086	0.0197	0.0098	0.0125	0.0062	0.0038	0.0019
w/o T	0.0179	0.0102	0.0158	0.0078	0.0118	0.0057	0.0047	0.0023	0.0029	0.0015
w/o T-DT	0.0108	0.0052	0.0132	0.0068	0.0101	0.0043	0.0032	0.0015	0.0020	0.0011

F. Ablation Analysis

We derive four variants to further analyze the design of our proposed CT4Rec, where each one removes a specific key component of CT4Rec. Details are presented as follows.

1) *w/o DT* is the simplified version of our proposed method, which removes the dynamic temperature strategy from CT4Rec.
2) *w/o CL* removes the contrastive translation module and CT4Rec degenerates into TransRec.
3) *w/o T* drops the personalized translation operation module, which makes the model only retain the contrastive learning module and the dynamic temperature strategy.
4) *w/o T-DT* removes the dynamic temperature strategy from *w/o T*.

We conduct the ablation analysis to investigate the effectiveness of each components. The performance comparison among four variants is presented in Table V. As shown in Table V, we can draw the conclusions as follows.

1) Each component contributes to the performance of our proposed sequential recommendation model, demonstrating the effectiveness of each component.
2) Comparing the performance of CT4Rec to *w/o DT*, as well as *w/o T* to *w/o T-DT*, we can observe that adopting the dynamic temperature strategy is able to further improve the performance of sequential recommendation models. This once again verifies the necessity of simultaneously capturing both the differences between individual instances and the differences among various clusters of entities.
3) The performances of *w/o T* and *w/o T-DT* are worse than that of *w/o CL*. The main reason is that the sequential recommendation models are unable to learn the original user preferences and sequential patterns when removing the personalized translation module. In contrastive learning-based sequential recommendation, the main task is to learn the informative representations of entities and sequential patterns, and we treat the contrastive learning task as an auxiliary task to promote the learning process of the sequential recommendation task.

G. Comparison of Training Efficiency and Memory Usage

In this section, we conduct two groups of experiments to evaluate the efficiency of our proposed method. We choose translation-based approaches (i.e., TransRec, CT4Rec/DT, CT4Rec) and Transformer-based approaches (i.e., SASRec, CL4SRec) for further analysis. We report the training time per epoch and memory usage of all compared methods in Tables VI and VII, respectively.

TABLE VI
COMPARISON OF TRAINING EFFICIENCY (SECONDS)

Dataset	TransRec	SASRec	CL4SRec	CT4Rec/DT	CT4Rec
Automotive	0.08	2.51	3.92	0.28	0.29
Instruments	0.12	2.79	4.42	0.63	0.65
Cell_Phones	0.43	6.50	11.95	4.46	4.50
Tools	0.66	7.53	14.84	6.08	6.14
Clothing	6.46	57.13	95.48	53.87	54.52

TABLE VII
COMPARISON OF MEMORY USAGE (MB)

Dataset	TransRec	SASRec	CL4SRec	CT4Rec/DT	CT4Rec
Automotive	649	1,639	1,768	1,166	1,296
Instruments	659	1,863	2,133	1,352	1,582
Cell_Phones	740	2,306	3,013	1,930	2,308
Tools	743	2,902	3,417	2,366	2,773
Clothing	2,450	7,700	11,030	7,878	9,770

As shown in Table VI, we can observe that TransRec has the least training time. This is because that TransRec does not require training of both Transformer and contrastive learning modules. In addition, the training speeds of SASRec and CL4SRec are slower than those of other methods. The main reason is that the optimization of Transformer is time-consuming. Moreover, the time cost of CL4SRec is about twice of the cost of SASRec on all datasets. One reason is that CL4SRec adopts the additional contrastive learning module to generate augmented views and compute contrastive loss. The training speeds of both CT4Rec/DT and CT4Rec are faster than that of SASRec and CL4SRec. According to the above observations, we argue that the optimization of Transformer component may be a bottleneck for Transformer-based sequential recommendations, which limits their applications in real-world practices. In addition, although CT4Rec adopts the dynamic temperature strategy, the computation costs of CT4Rec and CT4Rec/DT for one epoch are comparable, indicating that the dynamic temperature strategy does not involve the large overhead cost.

From Table VII, we can observe that the TransRec uses the least memory, since it mainly includes two parameters, i.e., the embedding of users and the embedding of items. Compared to

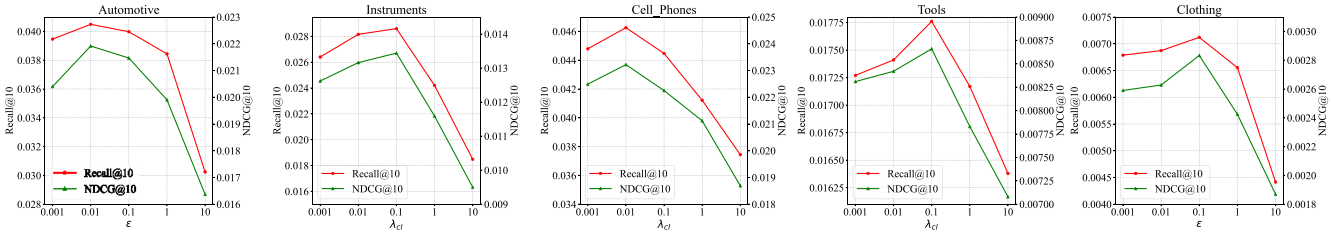


Fig. 6. Impact of different λ_{cl}.

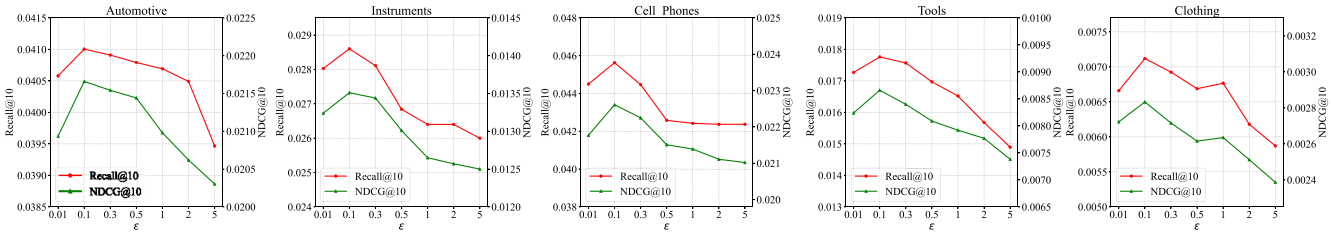


Fig. 7. Impact of different ε.

TransRec, SASRec needs more memory usage. The possible reason is that SASRec adopts a Transformer network as the backbone, which introduces a large number of extra parameters. In terms of memory usage, SASRec is superior to CL4SRec, indicating that the contrastive learning component not only requires additional time consumption but also costs extra memory usage. The memory usage of CT4Rec is almost 4 times than that of TransRec. The main reason is that the noise augmentation operation needs to allocate memory for two types of noise tensors and the dynamic temperature strategy needs to allocate memory for personalized temperature tensors. In summary, our proposed method is superior to Transformer-based sequential recommendation models in terms of the training efficiency and memory usage.

H. Parameter Sensitivity Analysis

In the following experiments, the parameter sensitivities of λ_{cl} and ε are analyzed.

1) Impact of λ_{cl}: We choose the Automotive, Instruments, Cell-Phones, Tools and Clothing to investigate how λ_{cl} affects the performance of our proposed CT4Rec. λ_{cl} is tuned within $\{0.001, 0.01, 0.1, 1, 10\}$. The results are presented in Fig. 6. The performance of CT4Rec increases at the beginning, and gradually reaches its peak when $\lambda_{cl} = 0.01$ on Automotive, 0.1 on Instruments, 0.01 on Cell_Phones, 0.1 on Tools, and 0.1 on Clothing. Then, the performance of CT4Rec begins to degrade. We suspect that a too small contrastive coefficient makes CT4Rec extract limited supervised signals from the contrastive learning module, resulting in suboptimal performance. On the contrary, an excessive contrastive coefficient makes CT4Rec only focus on the contrastive translation module, which ignores learning user long-term preferences and sequential patterns from user interactions behaviors. Compared to the contrastive translation module, learning user long-term preferences and sequential patterns from historical user interactions behaviors plays a more important role in sequential recommendation models.

2) Impact of ε: For the sensitivity of parameter ε, we change ε within $\{0.01, 0.1, 0.3, 0.5, 1, 2, 5\}$ and other parameters remain unchanged. From Fig. 7, when $\varepsilon \leq 0.1$, we observe that the performance of CT4Rec continuously improves with the increase of ε. Then, the performance of CT4Rec begins to decline when $\varepsilon > 0.1$. For the highest sparse dataset Clothing, the overall performance trend on Clothing is consistent with those observed from other datasets. It should be noted that, although the performance of CT4Rec slightly improves on clothing when ε is 1, the overall trend of Recall@10 and NDCG@10 is declining when $\varepsilon > 0.1$.

The above phenomenon indicates that a smaller ε is more beneficial for contrastive translation-based sequential recommendation. On the contrary, a larger ε hinders the performance of CT4Rec. This is owing to the fact that the similarities among augmented embeddings of users or items are completely dominated by noises, which makes CT4Rec fail to learn invariance between augmented views. On Instruments, Cell_Phones and Tools, CT4Rec achieves its best performance when ε is 0.1.

V. Conclusion

In this study, we propose a contrastive translation-based sequential recommendation model, namely, CT4Rec. CT4Rec generates augmented views of item sequences by injecting noises into embeddings of users and items, which is able to guarantee that the underlying semantics of augmented views are consistent with those of the original item sequence and alleviate the problem of false negative samples caused by inappropriate augmentation strategies. Moreover, we integrate the dynamic temperature module into CT4Rec, which enables CT4Rec not only capture individual differences among users/items, i.e., achieving instance discrimination, but also learn the differences among the clusters of users/items, i.e., achieving group-wise discrimination. Our validation on five benchmark datasets shows that CT4Rec out- performs SOTA sequential recommendation methods.

REFERENCES

[1] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[2] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. UAI*, 2009, pp. 452–461.

[3] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, "Collaborative metric learning," in *Proc. 26th WWW*, 2017, pp. 193–201.

[4] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. 19th WWW*, 2010, pp. 811–820.

[5] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new POI recommendation," in *Proc. IJCAI*, 2015, pp. 2069–2075.

[6] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2015, pp. 403–412.

[7] R. He, W.-C. Kang, and J. McAuley, "Translation-based recommendation," in *Proc. RecSys*, 2017, pp. 161–169.

[8] B. Wu, X. He, Z. Sun, L. Chen, and Y. Ye, "ATM: An attentive translation model for next-item recommendation," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1448–1459, Mar. 2020.

[9] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[10] A. Vaswani et al., "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.

[11] J. Chang et al., "Sequential recommendation with graph neural networks," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2021, pp. 378–387.

[12] L. Xue, D. Yang, and Y. Xiao, "Factorial user modeling with hierarchical graph neural network for enhanced sequential recommendation," in *Proc. ICME*, 2022, pp. 1–6.

[13] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *Proc. ICDM*, 2018, pp. 197–206.

[14] J. Li, Y. Wang, and J. McAuley, "Time interval aware self-attention for sequential recommendation," in *Proc. WSDM*, 2020, pp. 322–330.

[15] F. Sun et al., "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proc. CIKM*, 2019, pp. 1441–1450.

[16] Z. Liu, Y. Chen, J. Li, P. S. Yu, J. McAuley, and C. Xiong, "Contrastive self-supervised sequential recommendation with robust augmentation," 2021, *arXiv:2108.06479*.

[17] Y. Chen, Z. Liu, J. Li, J. McAuley, and C. Xiong, "Intent contrastive learning for sequential recommendation," in *Proc. WWW*, 2022, pp. 2172–2182.

[18] J. Wu et al., "Self-supervised graph learning for recommendation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2021, pp. 726–735.

[19] M. Jing, Y. Zhu, T. Zang, and K. Wang, "Contrastive self-supervised learning in recommender systems: A survey," *ACM Trans. Inf. Syst.*, vol. 42, no. 2, pp. 1–39, 2023.

[20] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, "Are graph augmentations necessary? Simple graph contrastive learning for recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2022, pp. 1294–1303.

[21] X. Xie et al., "Contrastive learning for sequential recommendation," in *Proc. ICDE*, 2021, pp. 1259–1273.

[22] A. Kukleva, M. Böhle, B. Schiele, H. Kuehne, and C. Rupprecht, "Temperature schedules for self-supervised contrastive methods on long-tail data," in *Proc. ICLR*, 2023, pp. 1–15.

[23] S. Manna, S. Chattopadhyay, R. Dey, S. Bhattacharya, and U. Pal, "DySTreSS: Dynamically scaled temperature in self-supervised contrastive learning," 2023, *arXiv:2308.01140*.

[24] L. Guo, J. Zhang, L. Tang, T. Chen, L. Zhu, and H. Yin, "Time interval-enhanced graph neural network for shared-account cross-domain sequential recommendation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 3, pp. 4002–4016, Mar. 2024.

[25] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proc. AAAI*, 2019, pp. 346–353.

[26] Y. Pang et al., "Heterogeneous global graph neural networks for personalized session-based recommendation," in *Proc. 15th ACM Int. Conf. Web Search Data Min.*, 2022, pp. 775–783.

[27] Z. Yin, K. Han, P. Wang, and X. Zhu, "H3GNN: Hybrid hierarchical hypergraph neural network for personalized session-based recommendation," *ACM Trans. Inf. Syst.*, vol. 42, no. 3, pp. 1–30, 2024.

[28] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. ICML*, 2020, pp. 1597–1607.

[29] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. CVPR*, 2020, pp. 9726–9735.

[30] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proc. EMNLP*, 2021, pp. 6894–6910.

[31] J. Giorgi, O. Nitski, B. Wang, and G. Bader, "DeCLUTR: Deep contrastive learning for unsupervised textual representations," in *Proc. ACL*, 2021, pp. 879–895.

[32] Z. Liu, H. Li, G. Chen, Y. Ouyang, W. Rong, and Z. Xiong, "PopDCL: Popularity-aware debiased contrastive loss for collaborative filtering," in *Proc. 32nd CIKM*, 2023, pp. 1482–1492.

[33] H. Wang, S. Yan, C. Wu, L. Han, and L. Zhou, "Cross-view temporal graph contrastive learning for session-based recommendation," *Know.-Based Syst.*, vol. 264, Mar. 2023, Art. no. 110304.

[34] Z. Liu et al., "Semantic-enhanced contrastive learning for session-based recommendation," *Knowl.-Based Syst.*, vol. 280, Nov. 2023, Art. no. 111001.

[35] Z. Wan et al., "Spatio-temporal contrastive learning-enhanced GNNs for session-based recommendation," *ACM Trans. Inf. Syst.*, vol. 42, no. 2, pp. 1–26, 2023.

[36] Y. Zhang et al., "Enhancing sequential recommendation with graph contrastive learning," in *Proc. IJCAI*, 2022, pp. 2398–2405.

[37] Y. Yang, C. Huang, L. Xia, C. Huang, D. Luo, and K. Lin, "Debiased contrastive learning for sequential recommendation," in *Proc. WWW*, 2023, pp. 1063–1073.

[38] Y. Liu, L. Xia, and C. Huang, "SelfGNN: Self-supervised graph neural networks for sequential recommendation," in *Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2024, pp. 1609–1618.

[39] R. Qiu, Z. Huang, H. Yin, and Z. Wang, "Contrastive learning for representation degeneration problem in sequential recommendation," in *Proc. WSDM*, 2022, pp. 813–823.

[40] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. NIPS*, 2013, pp. 2787–2795.

[41] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI*, 2014, pp. 1112–1119.

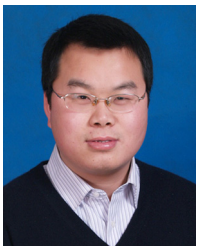
[42] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI*, 2015, pp. 2181–2187.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[44] J. Tang and K. Wang, "Personalized top-N sequential recommendation via convolutional sequence embedding," in *Proc. WSDM*, 2018, pp. 565–573.

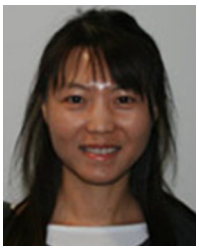
Aoran Zhang received the B.S. degree in computer science from the College of Tongda, Nanjing University of Posts and Telecommunications, Nanjing, China, in 2022. He is currently pursuing the M.S. degree with the Jiangsu University of Science and Technology, Zhenjiang, China.

His current research interests include data mining and recommendation systems.



Yonghong Yu received the Ph.D. degree in computer science from Nanjing University, Nanjing, China, in 2017.

He is a Professor with the Nanjing University of Posts and Telecommunications, Nanjing. His main research interests include machine learning and recommender systems.



Li Zhang (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Birmingham, Birmingham, U.K, in 2004.

She is currently an Professor with the Department of Computer Science, Royal Holloway, University of London, Egham, U.K. She holds expertise in machine learning, deep learning, computer vision, and intelligent robotics.

Rong Gao received the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 2018.

He is currently an Assistant Professor with the School of Computer Science, Hubei University of Technology, Wuhan. His research interests include machine learning and artificial intelligence.

Hongzhi Yin (Senior Member, IEEE) received the Ph.D. degree in computer science from Peking University, Beijing, China, in 2014.

He is a Professor with the University of Queensland, Brisbane, QLD, Australia. His research interests include recommendation system, user profiling, topic models, deep learning, social media mining, and location-based services.

Dr. Yin received the Australia Research Council Discovery Early-Career Researcher Award in 2015.