Diffusion-based Contrastive Learning for Sequential Recommendation

Ziqiang Cui
City University of Hong Kong
ziqiang.cui@my.cityu.edu.hk

Haolun Wu
McGill University
haolun.wu@mail.mcgill.ca

Bowei He
City University of Hong Kong
boweihe2-c@my.cityu.edu.hk

Ji Cheng
City University of Hong Kong
J.Cheng@my.cityu.edu.hk

Chen Ma*
City University of Hong Kong
chenma@cityu.edu.hk

ABSTRACT

Self-supervised contrastive learning, which directly extracts inherent data correlations from unlabeled data, has been widely utilized to mitigate the data sparsity issue in sequential recommendation. The majority of existing methods create different augmented views of the same user sequence via random augmentation, and subsequently minimize their distance in the embedding space to enhance the quality of user representations. However, random augmentation often disrupts the semantic information and interest evolution pattern inherent in the user sequence, leading to the generation of semantically distinct augmented views. Promoting similarity of these semantically diverse augmented sequences can render the learned user representations insensitive to variations in user preferences and interest evolution, contradicting the core learning objectives of sequential recommendation. To address this issue, we leverage the inherent characteristics of sequential recommendation and propose the use of context information to generate more reasonable augmented positive samples. Specifically, we introduce a context-aware diffusion-based contrastive learning method for sequential recommendation. Given a user sequence, our method selects certain positions and employs a context-aware diffusion model to generate alternative items for these positions with the guidance of context information. These generated items then replace the corresponding original items, creating a semantically consistent augmented view of the original sequence. Additionally, to maintain representation cohesion, item embeddings are shared between the diffusion model and the recommendation model, and the entire framework is trained in an end-to-end manner. Extensive experiments on five benchmark datasets demonstrate the superiority of our proposed method.

1 INTRODUCTION

Sequential recommendation (SR) systems predict the next item for users based on their historical interactions, which have demonstrated significant value on various online platforms like YouTube and Amazon. One of the major challenges in sequential recommendation is data sparsity [22, 43]. The limited and noisy user interaction records impede the training of complex SR models, thereby constraining their performance. Recently, contrastive learning has been employed to alleviate this issue, leading to significant advancements [3, 23, 36].

Contrastive learning extracts inherent data correlations directly from unlabeled data to enhance user representation learning, thereby improving SR models. Existing methods typically use data augmentation to create augmented views of original user sequences and maximize the agreement among different views of the same user. In terms of data augmentation levels, existing methods can be categorized into three types: 1) *Data Level*. This involves generating augmented views of user sequences by applying random augmentations [36] such as masking, substituting, reordering, and cropping. More informative methods based on item correlation are also used [21]. 2) *Model Level*. To reduce the disturbance to original sequences, some methods propose model-level operations [20, 23]. These involve performing a forward pass of neural networks on a user sequence twice, each time with a different dropout mask. However, the dropout operation still introduces a considerable amount of uncertainty due to its randomness. 3) *Mixed Level*. These approaches, exemplified by Qin et al. [22] and Zhou et al. [43], integrate both data-level and model-level augmentations. The aim is to extract more expressive features and establish distinct contrastive objectives for varying levels of augmentation.

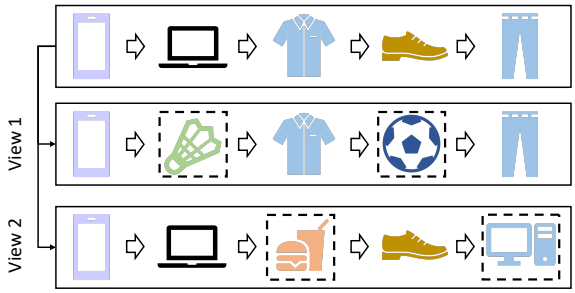


Figure 1: An example of augmented sequences with semantic discrepancies, where view 1 and view 2 are two augmented views of the original user sequence by random substitution

While the above studies have demonstrated efficacy in improving SR models, they neglect the rationality of the augmented positive samples. Most existing methods [3, 20, 22, 23, 36, 43] employ random augmentation either at the data or model level to generate augmented views, and regard two augmented views of the same user as a pair of positive samples. However, these methods introduces a large amount of uncertainty, which may cause unreasonable positive pairs. For example, Figure 1 shows an original user sequence and two augmented views obtained by applying the random substitution operation twice. It is evident that these two augmented sequences exhibit significant semantic discrepancies, primarily reflected in the following points: 1) The preference of

*Corresponding author.

View 1 focuses on sports and clothing, while View 2 mainly concentrates on electronic products. 2) The evolution of interests in the two views exhibits distinctly different patterns. Maximizing the representation agreement between such different views can cause learned representations to overlook significant semantic differences among user sequences, resulting in suboptimal solutions or even representation space collapse [23]. Moreover, this can lead to learned user representations being insensitive to different patterns of interest evolution, which contradicts the ultimate goal of sequential recommendation. Therefore, we argue that more effective contrastive learning should consider the rationality of data augmentation.

How can we generate more reasonable augmented views in sequential recommendation? Intuitively, reasonable data augmentation should take into account the characteristics of sequential recommendation. Sequential recommendation differs from other applications, such as computer vision, in two main aspects: 1) interaction records are often sparse, making the user sequence highly sensitive to modifications, and 2) there is a strong sequential interdependence between items in the sequence. Therefore, when modifying some items in a sequence, failing to consider the preceding and subsequent items and their sequential dependencies (i.e., *context information*) can lead to a complete change in the sequence's semantics, resulting in unreasonable augmented sequences. In light of this, we make the first attempt in this paper to introduce context information to improve the rationality of augmented views.

Our basic idea is to use context information as guidance to generate alternative items that align with the context information for specific positions within a sequence. These generated items then replace the corresponding original items, creating a positively augmented view of the original sequence. Two augmentations of the same user sequence serve as a pair of positive samples for contrastive learning. From a high-level perspective, we learn the conditional distribution of each position within a sequence based on its context information, and generate replacements according to this conditional distribution, thereby producing augmented sequences.

To achieve this goal, we propose the **C**ontext-**a**ware **Di**ffusion-based Contrastive Learning for Sequential **Rec**ommendation, named ***CaDiRec***. CaDiRec generates more reasonable augmented samples through conditional generation. Specifically, we employ a diffusion model as the generator due to its remarkable capabilities in learning underlying data distributions and robust conditional generation [12, 17]. A bidirectional Transformer [4] serves as the encoder of the diffusion model to capture complex sequential dependencies of the context information. The learned context representation guides the diffusion model to gradually refine the generated items, enabling it to accurately learn the conditional distribution of items within a sequence. During contrastive learning, CaDiRec generates alternative items by sampling from the learned conditional distribution. This ensures that the generated items are coherent with the context and sequential dependencies, thereby producing more reasonable augmented sequences. Moreover, to align the embedding space of the diffusion model with that of the SR model, we train both models jointly with shared item embeddings in an end-to-end manner. By integrating these designs, CaDiRec effectively enhances the quality of data augmentation for contrastive learning, leading to better user modeling and improved recommendation performance.

Our main contributions are summarized as follows:

- We propose a novel model, CaDiRec, that generates reasonable augmented views for contrastive learning through conditional generation, thereby improving sequential recommendation.
- To the best of our knowledge, this is the first work to explore the use of context information (i.e., both preceding and succeeding items) for contrastive learning in sequential recommendation.
- We conduct extensive experiments on five public benchmark datasets, and the results demonstrate the superiority of our method.

2 RELATED WORK

In this section, we summarize the related works from the following three fields: *(i)* sequential recommendation, *(ii)* self-supervised contrastive learning, and *(iii)* diffusion models.

2.1 Sequential Recommendation

Sequential recommendation aims to model a user's preference based on their historical interactions. In the initial phase, researchers treated the evolution of user interests as a Markov process and employed Markov chains to predict the next item for each user [9, 25]. With the rapid advancements in deep learning, various techniques such as convolutional neural networks (CNN) and recurrent neural networks (RNN) have been utilized in sequential recommendation [10, 11, 28], leading to remarkable achievements. Subsequently, the introduction of the attention mechanism has significantly enhanced recommendation performance. SASRec [14], for instance, is the pioneering work that employs the self-attention mechanism to model the evolution of user preference. Following that, BERT4Rec [27] is proposed to use a bidirectional self-attention encoder to capture context information of the user sequence. Recently, many self-attention-based methods have made improvements to existing approaches, achieving notable progress.

2.2 Self-Supervised Contrastive Learning

Self-supervised learning is widely used to tackle challenges associated with data sparsity and noise. It improves representation learning by constructing informative supervisory signals from the unlabeled data itself. Self-supervised learning has been extensively applied in various domains, such as computer vision (CV) [1, 2, 16] and natural language processing (NLP) [37].

Due to the inherent issues of user behavior sparsity and noisy interaction records in recommendation scenarios, self-supervised contrastive learning has played a crucial role in multiple recommendation tasks [13, 33, 35, 38, 40, 41]. When it comes to sequential recommendation, researchers design informative contrastive learning objectives for learning better user representations from historical interactions. S^3-Rec [42] introduces a method that incorporates auxiliary self-supervised objectives to learn the correlations among items, attributes, and segments. CL4SRec [36] designs three data-level augmentation operators, namely crop, mask, and reorder, which are employed to generate positive pairs and promote the invariance of their representations. However, introducing random perturbations to the already sparse interaction records of a user can alter her original preference, and maximizing the agreements among semantically inconsistent sequences can lead the model

to obtain suboptimal solutions. To solve this issue, CoSeRec [21] suggests substituting a specific item in the sequence with a similar item. However, the item similarity is measured by simple co-occurrence counts or item embedding distance, neglecting the context information of user behaviors. Later, DuoRec [23] proposes a model-level augment strategy, which generates positive augment pairs by forward-passing an input sequence twice with different Dropout masks. However, this approach is also a kind of random augment at the model level, lacking the ability to maintain semantic consistency. In addition, ICLRec [3] attempts to extract user intent from sequential information and subsequently performs contrastive learning between user representations and intent representations. ECL-SR [43] designs different contrastive learning objectives for augmented views at different levels. MCLRec [22] further combines data-level and model-level augmentation strategies, which applies random data augmentation proposed by CL4SRec to the input sequence and then feed the augmented data into MLP layers for the model-level augment.

However, the design intentions of these methods do not reflect the constraints on semantic consistency in the augmented views, which can potentially lead to the generation of incorrect positive samples. In addition, they do not take into account context information, which is important for preserving the semantic consistency.

2.3 Diffusion Models

Diffusion Models have gained significant prominence as a dominant approach in diverse generative tasks, such as image synthesis [5, 12, 26] and text generation [7, 17]. They demonstrate superior generative capabilities compared to alternative models such as GANs [8] and VAEs [15], which can be attributed to their precise approximation of the underlying data generation distribution and provision of enhanced training stability.

Recently, diffusion models have been employed in the field of sequential recommendation. Some methods [6, 18, 31, 32, 39] directly utilize diffusion models as the fundamental architecture for sequential recommendation. Specifically, these methods employ a left-to-right unidirectional Transformer to extract guidance signals for the generation of the next item. In contrast, other approaches [19, 34] adopt a two-stage paradigm for data augmentation. Initially, they train a diffusion model to generate pseudo user interactions aimed at expanding the original user sequences. These augmented datasets are then used to train downstream recommendation models. It should be noted that they solely rely on the unidirectional information of user behavior sequences as the diffusion guidance.

Different from these existing methods, our approach leverages the diffusion model for contrastive learning. Specifically, we employ the diffusion model to generate semantic-consistent augmented views of the original sequences and maximize the agreement among different views from the same user. To the best of our knowledge, this is the first instance of employing diffusion models for contrastive learning in the field of sequential recommendation.

3 PRELIMINARY

In this section, we first define our problem statement, followed by introducing basic knowledge of diffusion models.

3.1 Problem Statement

The primary objective of sequential recommendation is to provide personalized recommendations for the next item to users, leveraging their historical interactions. We denote the user and item sets as $\mathcal{U}$ and $\mathcal{V}$, respectively. Each user $u \in \mathcal{U}$ has a chronological sequence of interacted items $\mathbf{s}^u = [v_1^u, v_2^u ..., v_{|\mathbf{s}^u|}^u]$, where v_t^u indicates the item that u interacted with at step t, and $|\mathbf{s}^u|$ is the number of interacted items of user u. The goal is to predict the next item at time step $|\mathbf{s}^u| + 1$ according to $\mathbf{s}^u$, which can be formulated as:

$$\arg\max_{v_i \in \mathcal{V}} P(v_{|\mathbf{s}^u|+1} = v_i | \mathbf{s}^u), \tag{1}$$

where the probability P represents the likelihood of item v_i being the next item, conditioned on $\mathbf{s}_u$.

3.2 Diffusion Models

We provide an introduction to the fundamental principles of diffusion models based on DDPM [12]. Typically, a diffusion model consists of forward and reverse processes. Given a data point sampled from a real-world data distribution $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, the forward process gradually corrupts $\mathbf{x}_0$ into a standard Gaussian noise $\mathbf{x}_T \sim N(0; \mathbf{I})$, which is formulated as:

$$\begin{aligned} q(\mathbf{x}_{1:T}|\mathbf{x}_0) &= \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \\ q(\mathbf{x}_t|\mathbf{x}_{t-1}) &= \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}), \end{aligned} \tag{2}$$

where $\beta_t \in (0, 1)$ is the variance scale at time step t.

After the completion of the forward process, the reverse denoising process aims to gradually reconstruct the original data $\mathbf{x}_0$. This is achieved by sampling from $\mathbf{x}_T$ using a learned diffusion model, which can be formulated as:

$$\begin{aligned} p_\theta(\mathbf{x}_{0:T}) &= p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \\ p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) &= \mathcal{N}\big(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)\big). \end{aligned} \tag{3}$$

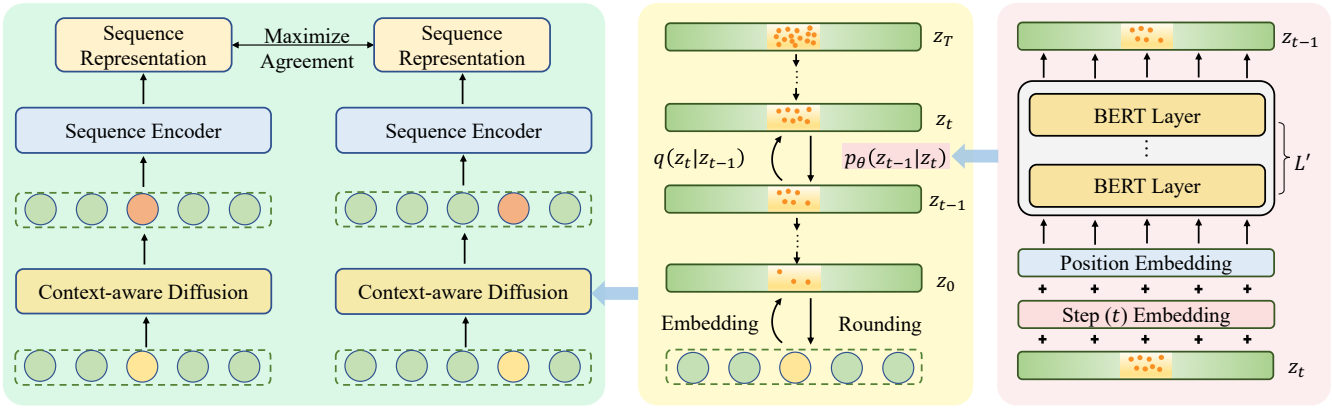
Training can be performed by optimizing the variational lower bound of $\log p_\theta(\mathbf{x}_0)$:

$$\begin{aligned} \mathcal{L}_{\text{vlb}}(\mathbf{x}_0) = \mathop{\mathbb{E}}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} [&\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} \\ &+ \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]. \end{aligned} \tag{4}$$

Ho et al. [12] further propose to utilize the KL divergence for more efficient training, which directly compares $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ against forward process posteriors, resulting in a mean-squared error loss:

$$\mathcal{L}_{\text{simple}}(\mathbf{x}_0) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{q(\mathbf{x}_t|\mathbf{x}_0)} ||\mu_\theta(\mathbf{x}_t, t) - \hat{\mu}(\mathbf{x}_t, \mathbf{x}_0)||^2, \tag{5}$$

where $\mu_\theta(\mathbf{x}_t, t)$ is the predicted mean of $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ computed by a neural network, and $\hat{\mu}(\mathbf{x}_t, \mathbf{x}_0)$ is the mean of the posterior $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$, which is tractable when conditioned on $\mathbf{x}_0$.



(a) The framework of our proposed context-aware diffusion-based contrastive learning for sequential recommendation (CaDiRec)

(b) The diffusion process of our context-aware diffusion model

(c) The bidirectional encoder of our diffusion model

Figure 2: Overview of our proposed CaDiRec. (a) illustrates the framework of CaDiRec. For some items in the sequence, CaDiRec generates alternative items based on context information to replace them, thereby producing augmented views of the original sequence. Different augmented views of the same user are considered as a pair of positive samples. (b) shows the diffusion process of our context-aware diffusion model. In the forward process, noise is gradually added to only some items in the sequence. In the reverse process, context is used to guide the restoration of the conditional distribution at the corresponding positions step by step. (c) depicts the encoder structure of the diffusion model. CaDiRec employs a bidirectional transformer to capture contextual dependencies, providing effective guidance for the denoising process.

4 METHOD

This section begins with an introduction to the sequential recommendation model. Next, we present the details of our proposed context-aware diffusion-based contrastive learning method, as shown in Figure 2. Finally, we introduce the end-to-end training objective of the whole framework.

4.1 Sequential Recommendation Model

Similar to many previous studies [3, 22, 23], our framework uses a Transformer-based architecture for sequential recommendation task, which comprises the embedding layer, the transformer layer, and the prediction layer.

4.1.1 Embedding Layer. We create an item embedding matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}|\times d}$ for the item set, where d represents the latent dimensionality. Given a user sequence $\mathbf{s} = [v_1, v_2, ..., v_n]$ where n is the max sequence length, we can obtain the input embedding vectors $\mathbf{e} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_n] \in \mathbb{R}^{n\times d}$ with respect to $\mathbf{s}$. In addition, we also construct a position embedding matrix $\mathbf{P} \in \mathbb{R}^{n\times d}$. For w-th item of the sequence, we add the item embedding $\mathbf{e}_w$ and the corresponding position embedding $\mathbf{p}_w$, resulting the final input vector at step w $\mathbf{h}_w^0 = \mathbf{e}_w + \mathbf{p}_w$, and $\mathbf{h}^0 = [\mathbf{h}_1^0, \mathbf{h}_2^0, ..., \mathbf{h}_n^0]$ denotes the representation of input sequence $\mathbf{s}$.

4.1.2 User Sequence Encoder. Following the embedding layer, the input vector $\mathbf{h}^0$ is passed through L Transformer blocks to learn the user sequence representations. Each Transformer (Trm) block consists of a self-attention layer and a feed-forward network layer, which can be formulated as:

$$\mathbf{h}^L = \mathbf{Trm}(\mathbf{h}^0), \tag{6}$$

where $\mathbf{h}^L \in \mathbb{R}^{n\times d}$ denotes the hidden states of the last layer, and the vector of the last position $\mathbf{h}_n^L \in \mathbb{R}^d$ is used to represent the whole user sequence.

4.1.3 Prediction Layer. The goal of sequential recommendation is to predict the next item. In the prediction layer, we first calculate the similarities between the user sequence representation vector $\mathbf{h}_n^L$ and item embedding vectors through an inner-product as:

$$\mathbf{r} = \mathbf{h}_n^L \mathbf{M}^{\mathrm{T}}, \tag{7}$$

where $\mathbf{r} \in \mathbb{R}^{|\mathcal{V}|}$, and r_i is the likelihood of v_i being the next item. The items are then ranked based on $\mathbf{r}$ to generate the top-k recommendation list.

During training, we adopt the Binary Cross-Entropy (BCE) loss with negative sampling to train the SR model, following many previous methods [14, 36, 42].

$$\mathcal{L}_{\text{rec}} = -\sum_{u\in\mathcal{U}}\sum_{t=1}^{n} \log\left(\sigma(\mathbf{h}_t^L \cdot \mathbf{e}_{v_{t+1}})\right) + \log\left(1-\sigma(\mathbf{h}_t^L \cdot \mathbf{e}_{v_j^-})\right), \tag{8}$$

where we pair each ground-truth item v_{t+1} with one negative item v_j^- that is randomly sampled from the item set.

4.2 The Framework of CaDiRec

In this section, we introduce the overall framework of our proposed context-aware diffusion-based contrastive learning method. The details of the context-aware diffusion model will be introduced in next section.

Existing methods neglect the context information of user sequences, thereby potentially generating unreasonable augmented views for contrastive learning. In contrast to these methods, we propose to utilize context information as a guidance to generate

more reasonable augmented views through conditional generation. Different augmented views of the same user are considered as a pair of positive samples for contrastive learning, thereby improving the SR model. The framework is shown in Figure 2 (a).

Specifically, given a sequence $\mathbf{s}^u$ of user u, we select a subset of items within $\mathbf{s}^u$ with a pre-defined ratio ρ, and the position indices of the selected items within the sequence are recorded as $\mathbf{a}_1^u$. Next, we employ the context-aware diffusion model to generate items that align with the context information. The generated items are then used to replace the original items at $\mathbf{a}_1^u$ positions of sequence $\mathbf{s}^u$, resulting the augmented sequence $\mathbf{s}_1^u$. That is, the sole distinction between the original sequence $\mathbf{s}^u$ and the augmented sequence $\mathbf{s}_1^u$ is the replacement of selected items from $\mathbf{s}^u$ with context-aligned items generated by the diffusion model. The details of the proposed diffusion model will be introduced in Sec. 4.3.2. By repeating a similar operation, we can obtain another augmented view $\mathbf{s}_2^u$ with respect to another set of selected position indices $\mathbf{a}_2^u$. Note that our method takes into account contextual information and sequential dependencies. Therefore, the generated augmented views do not disrupt the user's interest preferences and interest evolution. Consequently, two augmented views $\mathbf{s}_1^u$ and $\mathbf{s}_2^u$ of user u can be considered as a pair of rational positive samples, and their representations should be brought closer.

We adopt the standard contrastive loss function to maximize the representation agreement between two different augmented views of the same user sequence and minimize the agreement between the augmented sequences derived from different users. Specifically, for $\mathbf{s}_1^u$ and $\mathbf{s}_2^u$ of user u, we first obtain their embeddings and then input them to the user sequence encoder defined in Sec. 4.1.2 to generate their representation $\tilde{\mathbf{h}}_1^u$ and $\tilde{\mathbf{h}}_2^u$ according to Eq. (6). In this way, we can obtain the representations corresponding to the two augmented views for all users. For user u, $\tilde{\mathbf{h}}_1^u$ and $\tilde{\mathbf{h}}_2^u$ are regarded as the positive pair, while the remaining $2(N-1)$ augmented representations within the same batch are treated as negative samples $\mathbf{H}^-$, where N is the batch size. Then, we employ the inner product to assess the representation similarity. Finally, we define the loss function $\mathcal{L}_{\mathbf{cl}}$ in a similar manner to the widely used cross-entropy loss as follows:

$$\mathcal{L}_{\mathrm{cl}}^u = -\log \frac{\exp\left(\mathrm{sim}(\tilde{\mathbf{h}}_1^u, \tilde{\mathbf{h}}_2^u)\right)}{\exp\left(\mathrm{sim}(\tilde{\mathbf{h}}_1^u, \tilde{\mathbf{h}}_2^u)\right) + \sum_{\tilde{\mathbf{h}}^- \in \mathbf{H}^-} \exp\left(\mathrm{sim}(\tilde{\mathbf{h}}_1^u, \tilde{\mathbf{h}}^-)\right)}, \quad (9)$$

where $\mathrm{sim}(\cdot)$ denotes the inner product of vectors.

4.3 Context-aware Diffusion Model

In this section, we present our proposed context-aware diffusion model, which is shown in Figure 2 (b). Our diffusion model generates items for replacing the original ones by utilizing the context information of the selected positions, thereby achieving context-aligned data augmentation. The diffusion process consists of the forward process with partial position noising and the reverse process with context-conditional denoising.

4.3.1 Forward Process with Partial Position Noising. In the forward process, we gradually add noise to the selected items of the user sequence. Specifically, at the start of the forward process, we incorporate a Markov transition from discrete input items to a continuous space using the embedding map, following Diffusion-LM [17]. This transition is parametrized by $q_\phi(\mathbf{z}_0|\mathbf{s}) = \mathcal{N}(\mathbf{e}, \beta_0\mathbf{I})$, where $\mathbf{e}$ represents the embedding vectors corresponding to the sequence $\mathbf{s}$ as defined in Section 4.1.1. This transformation allows us to integrate the discrete sequence into the standard forward process. At each forward step $q(\mathbf{z}_t|\mathbf{z}_{t-1})$, we incrementally add Gaussian noise into the hidden states of the previous time step $\mathbf{z}_{t-1}$, to obtain $\mathbf{z}_t$.

Unlike other diffusion models, we selectively apply noise to items at randomly chosen positions with a certain ratio ρ instead of the entire sequence, while retaining the items at the remaining positions (i.e., context information). This approach allows the hidden vectors at the remaining positions and their relative positions to act as the conditional guidance during the reverse phase, enabling our model to utilize context information for controlling item generation.

4.3.2 Reverse Process with Context-Conditional Denoising. In the reverse process, context is used to guide the restoration of the conditional distribution at the corresponding positions step by step. Specifically, the objective of the denoising process is to gradually remove noise starting from $\mathbf{z}_T$ and ultimately recover the original data distribution, which is formulated as:

$$p_\theta(\mathbf{z}_{0:T}) = p(\mathbf{z}_T) \prod_{t=1}^{T} p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t). \quad (10)$$

We use a learnable model $f_\theta(\mathbf{z}_t, t)$ to model the reverse process at each step:

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) = \mathcal{N}\big(\mathbf{z}_{t-1}; \mu_\theta(\mathbf{z}_t, t), \Sigma_\theta(\mathbf{z}_t, t)\big). \quad (11)$$

Following Diffusion-LM [17], we incorporate a trainable rounding step $p_\theta(\mathbf{s}|\mathbf{z}_0) = \prod_{i=1}^{n} p_\theta(v_i|z_i)$ in the reverse process to map the hidden states back to the embedding space, where $p_\theta(v_i|z_i)$ is a softmax distribution. More details about the rounding step can be found in [17]. In addition, we set $\Sigma_\theta(\mathbf{z}_t, t)$ to untrained time dependent constants following previous methods [7, 12, 17].

Note that only the hidden vectors corresponding to items selected in the forward process are subjected to the addition of noise. Therefore, during the reverse process, the hidden vectors of items at the remaining positions (i.e., context information) as well as their position encoding can serve as a condition to guide the generation.

Here, we require a model architecture that can effectively encode the context information for learning the conditional distribution, thereby guiding the item generation. However, there are complex sequential dependencies between items in sequential recommendation. If we fail to capture these patterns, we cannot effectively utilize the context information. Fortunately, the bidirectional Transformer (BERT) [4, 30] offers an exciting alternative for achieving this goal. Due to the equipment of bidirectional self-attention mechanism and the position encoding, the bidirectional Transformer can capture a comprehensive understanding of the context from both left and right items. Therefore, we employ a bidirectional Transformer to model $f_\theta(\mathbf{z}_t, t)$. The architecture of the our encoder is shown in Fig. 2 (c), which is constructed by stacking L' BERT layers together. Each BERT layer consists of a multi-head self-attention layer and a position-wise feed-forward network. At diffusion step t, the encoder receives $\mathbf{z}_t$ along with the positional encoding of the sequence and the diffusion step encoding, and then outputs $\mathbf{z}_{t-1}$.

To train the diffusion model, we compute the variational lower bound following previous methods [7, 12, 17]. As we have incorporated the embedding step and rounding step, the variational lower bound loss $\mathcal{L}_{vlb}$ introduced in Eq. (4) now becomes as follows:

$$\mathcal{L}'_{vlb} = \mathop{\mathbb{E}}_{q_\phi(\mathbf{z}_0|\mathbf{s})} \left[\mathcal{L}_{vlb}(\mathbf{z}_0) + \log q_\phi(\mathbf{z}_0|\mathbf{s}) - \log p_\theta(\mathbf{s}|\mathbf{z}_0) \right]. \quad (12)$$

Following previous methods [7, 17], this training objective can be further simplified as:

$$\begin{aligned} \mathcal{L}_d &= \sum_{t=2}^{T} ||\mathbf{z}_0 - f_\theta(\mathbf{z}_t, t)||^2 + ||\mathbf{e} - f_\theta(\mathbf{z}_1, 1)||^2 - \log p_\theta(\mathbf{s}|\mathbf{z}_0) \\ &\rightarrow \sum_{t=2}^{T} ||\tilde{\mathbf{z}}_0 - \tilde{f}_\theta(\mathbf{z}_t, t)||^2 + ||\tilde{\mathbf{e}} - \tilde{f}_\theta(\mathbf{z}_1, 1)||^2 - \log p_\theta(\mathbf{s}|\mathbf{z}_0), \end{aligned} \quad (13)$$

where $\tilde{\mathbf{z}}_0$, $\tilde{f}_\theta$, and $\tilde{\mathbf{e}}$ denote the part of $\mathbf{z}_0$, f_θ, and $\mathbf{e}$ corresponding to selected positions, respectively. Note that while we only calculate the loss with respect to the selected positions in the first term, the reconstruction of the selected items $\tilde{\mathbf{z}}_0$ also takes into account the remaining items (i.e., context information) of the sequence due to the bidirectional self-attention mechanism.

4.3.3 Generating Augmented Views. During contrastive learning, the diffusion model acts as a data generator to generate reasonable augmented views, thereby improving the contrastive learning. Given the user sequence $\mathbf{s}$, we target to generate context-aligned items for arbitrary position indices τ. We first randomly sample $\tilde{\mathbf{z}}_T \sim N(0; \mathbf{I})$ to replace the item embeddings $\mathbf{e}$ with respect to selected position indices τ to obtain $\mathbf{z}_T$. Then, we can iterate the reverse procedure until we reach the initial state $\mathbf{z}_0$. Following DiffuSeq [7], for each step, we adopt the following operations: 1) performing the rounding step (defined in Sec. 4.3.2) on $\mathbf{z}_t$ to map it back to item embedding space; 2) replacing the part of recovered $\mathbf{z}_{t-1}$ that does not belong to selected positions τ with the original item embeddings, thereby preserving context information. Note that due to the different initial random noise, the generated items with the same context information will exhibit a certain level of diversity, which is also important for contrastive learning. Finally, through the substitution of generated items into the corresponding positions of the original sequence, an augmented sequence is obtained. Performing the same operation twice with different selected positions τ for the same user results in a pair of positive samples.

4.4 End-to-End Training

As both of the diffusion model and SR model rely on item embeddings, employing separate sets of item embeddings would result in a misalignment between the representation spaces of two models. To overcome this challenge, we propose to share item embeddings between the diffusion model and SR model, and train the full framework in an end-to-end manner. Therefore, the objective function is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \alpha \mathcal{L}_{cl} + \beta \mathcal{L}_d, \quad (14)$$

where α and β are hyperparameters that determine the weightings. $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{cl}$ represent the loss for the sequential recommendation (SR) task and the contrastive learning task, respectively. $\mathcal{L}_d$ is the loss for the diffusion model. The training of the diffusion model aims to learn better conditional distributions, thereby generating more reasonable augmented samples for contrastive learning.

Note that our model, due to the introduction of diffusion model-based data augmentation, has a longer training time compared to random augmentation-based SR models like CL4SRec [36]. However, the training time is comparable to that of diffusion-based recommendation methods like DreamRec [39]. Additionally, our data augmentation and contrastive learning are only performed during training. During model inference, the recommendation results are directly produced by the SR model, without involving the diffusion model. Consequently, our method has a much faster inference time compared to other diffusion-based SR models [39] and is comparable to random augmentation-based SR methods [36].

Table 1: Dataset description.

Datasets	#Users	#Items	#Actions	Avg. Length	Density
ML-1m	6,040	3,953	1,000,209	165.6	4.19%
Beauty	22,363	12,101	198,502	8.8	0.07%
Sports	35,598	18,357	296,337	8.3	0.05%
Toys	19,412	11,924	167,597	8.6	0.07%
Yelp	30,431	20,033	316,354	10.4	0.05%

5 EXPERIMENTS

5.1 Experimental Settings

5.1.1 Datasets. We conduct experiments on five real-world public datasets, including MovieLens, Beauty, Sports, Toys, and Yelp. The statistics of these datasets are shown in Table 1. These datasets encompass a wide range of application scenarios. The MovieLens[1] dataset is a stable benchmark dataset which collects movie ratings provided by users. Beauty, Sports, and Toys datasets are obtained from Amazon[2], one of the largest e-commerce platforms globally. Yelp is a renowned dataset primarily used for business recommendation. We adopt the same preprocessing method as employed in numerous previous studies [21, 36], filtering items and users with fewer than five interaction records.

5.1.2 Evaluation Metrics. To evaluate the performance of our model and baseline models, we employ widely recognized evaluation metrics: Hit Rate (HR) and Normalized Discounted Cumulative Gain (NDCG), and report values of HR@k and NDCG@k for k=5 and 10. We use the standard leave-one-out strategy, utilizing the last and second-to-last interactions for testing and validation, respectively, while the remaining interactions serve as training data. To ensure unbiased evaluation, we rank all items in the item set and compute the metrics based on the rankings across the entire item set.

5.1.3 Baseline Methods. To ensure a comprehensive assessment, we compare our method with eleven baseline methods, which can be divided into three categories: classical methods (BPR-MF, Caser, SASRec, BERT4Rec), contrastive learning based methods (S^3-Rec, CL4SRec, CoSeRec, DuoRec, MCLRec), and diffusion based methods (DiffuASR, DreamRec).

[1] https://grouplens.org/datasets/movielens/
[2] http://jmcauley.ucsd.edu/data/amazon/

Table 2: Performance comparison of different methods on five datasets. CaDiRec achieves state-of-the-art results among all baseline models, as confirmed by a paired t-test with a significance level of 0.01.

Dataset	Metric	BPR-MF	Caser	SASRec	BERT4Rec	S^3-Rec	CL4SRec	CoSeRec	DuoRec	MCLRec	DiffuASR	DreamRec	**CaDiRec**	Improv.
ML-1m	HR@5	0.0164	0.0836	0.1112	0.0925	0.1082	0.1147	0.1162	0.1216	0.1298	0.1105	0.1205	**0.1504**	15.9%
	ND@5	0.0097	0.0451	0.0645	0.0522	0.0624	0.0672	0.0684	0.0702	0.0824	0.0637	0.0813	**0.1001**	21.5%
	HR@10	0.0354	0.1579	0.1902	0.1804	0.1961	0.1978	0.1952	0.1996	0.2047	0.1892	0.2006	**0.2282**	11.5%
	ND@10	0.0158	0.0624	0.0906	0.0831	0.0922	0.0932	0.0974	0.1003	0.1055	0.0903	0.1077	**0.1251**	16.2%
Beauty	HR@5	0.0122	0.0256	0.0384	0.0360	0.0387	0.0401	0.0404	0.0422	0.0437	0.0388	0.0440	**0.0495**	12.5%
	ND@5	0.0071	0.0147	0.0249	0.0216	0.0244	0.0258	0.0265	0.0264	0.0278	0.0251	0.0274	**0.0314**	12.9%
	HR@10	0.0298	0.0342	0.0628	0.0601	0.0646	0.0651	0.0648	0.0669	0.0689	0.0633	0.0687	**0.0718**	4.2%
	ND@10	0.0132	0.0236	0.0321	0.0308	0.0327	0.0322	0.0334	0.0336	0.0357	0.0316	0.0352	**0.0386**	8.1%
Sports	HR@5	0.0095	0.0154	0.0225	0.0217	0.0173	0.0221	0.0245	0.0232	0.0249	0.0217	0.0248	**0.0276**	10.8%
	ND@5	0.0062	0.0124	0.0142	0.0143	0.0112	0.0129	0.0159	0.0154	0.0161	0.0138	0.0151	**0.0183**	13.7%
	HR@10	0.0193	0.0261	0.0339	0.0359	0.0311	0.0383	0.0372	0.0362	0.0382	0.0322	0.0374	**0.0426**	11.2%
	ND@10	0.0091	0.0138	0.0174	0.0181	0.0147	0.0173	0.0205	0.0189	0.0197	0.0166	0.0191	**0.0233**	13.7%
Toys	HR@5	0.0102	0.0169	0.0453	0.0461	0.0443	0.0468	0.0474	0.0459	0.0491	0.0448	0.0497	**0.0522**	5.0%
	ND@5	0.0061	0.0106	0.0306	0.0311	0.0294	0.0317	0.0323	0.0322	0.0327	0.0312	0.0316	**0.0356**	8.9%
	HR@10	0.0135	0.0271	0.0675	0.0665	0.0693	0.0684	0.0695	0.0681	0.0702	0.0667	0.0643	**0.0785**	11.8%
	ND@10	0.0094	0.0140	0.0374	0.0368	0.0375	0.0388	0.0401	0.0385	0.0412	0.0382	0.0402	**0.0441**	7.0%
Yelp	HR@5	0.0127	0.0151	0.0161	0.0186	0.0199	0.0201	0.0198	0.0199	0.0209	0.0157	0.0174	**0.0238**	13.9%
	ND@5	0.0074	0.0096	0.0100	0.0118	0.0118	0.0124	0.0120	0.0123	0.0129	0.0102	0.0116	**0.0149**	15.5%
	HR@10	0.0273	0.0253	0.0274	0.0338	0.0291	0.0349	0.0323	0.0342	0.0354	0.0268	0.0245	**0.0387**	9.3%
	ND@10	0.0121	0.0129	0.0136	0.0171	0.0168	0.0181	0.0179	0.0189	0.0177	0.0133	0.0152	**0.0197**	4.2%

- **BPR-MF** [24]. It employs matrix factorization to model users and items, and uses the pairwise Bayesian Personalized Ranking (BPR) loss to optimize the model.
- **SASRec** [14]. It is the first work to utilize the self-attention mechanism for sequential recommendation.
- **Caser** [28]. It utilizes a CNN-based approach to model high-order relationships in the context of sequential recommendation.
- **BERT4Rec** [27]. It employs the BERT [4] framework to capture the context information of user behaviors.
- **S^3-Rec** [42]. It leverages self-supervised learning to uncover the inherent correlations within the data. However, its primary emphasis lies in integrating the user behavior sequence and corresponding attribute information.
- **CL4SRec** [36]. It proposes three random augmentation operators to generate positive samples for contrastive learning.
- **CoSeRec** [21]. It introduces two informative augmentation operators leveraging item correlations based on CL4SRec. We compare with these informative augmentations.
- **DuoRec** [23]. It combines a model-level augmentation and a novel sampling strategy for choosing hard positive samples.
- **MCLRec** [22]. It integrates both data-level and model-level augmentation strategies, utilizing CL4SRec's random data augmentation for the input sequence and employing MLP layers for model-level augmentation.
- **DiffuASR** [19]. It leverages the diffusion model to generate pseudo items and concatenates them at the beginning of raw sequences. Then, the extended sequences are fed into a downstream recommendation model for next item prediction.
- **DreamRec** [39]. It directly utilizes the diffusion model to generate the next item based on the historical interactions.

5.1.4 Implementation Details. We implement all baseline methods according to their released code. The embedding size for all methods is set to 64. Our method utilizes a Transformer architecture for the SR model, comprising 2 layers and 2 attention heads each layer. Meanwhile, our diffusion model employs a bidirectional Transformer with 1 layer and 2 attention heads. The total number of diffusion steps is set to a fixed value of 1000. We tune the coefficients of the two critical terms in the loss function, α and β within the range of [0.1, 0.2, 0.4, 0.6, 0.8, 1.0]. Additionally, we explore the substitution ratio ρ within the range of [0, 0.1, 0.2, 0.4, 0.6, 0.8]. The Dropout rate is chosen from the set {0.1, 0.2, 0.3, 0.4, 0.5} for both the embedding layer and the hidden layers. We set the training batch size to 256 and employ the Adam optimizer with a learning rate of 0.001. Following most previous works [14], we set the max sequence length to 50 for three Amazon datasets and Yelp, and to 200 for the MovieLens dataset. For sequences with fewer interactions than the maximum sequence length, we will pad them with a padding token to match the max sequence length.

It is noteworthy that for the recommendation task, the majority of baseline models employ the negative sampling strategy during the training process. Specifically, for each positive sample, one negative sample is randomly selected, and optimization is performed using the Binary Cross-Entropy (BCE) loss. However, some methods, such as DuoRec, do not utilize negative sampling. Instead, they calculate the probability of each item across the entire item set using the softmax function. This approach, however, becomes impractical when dealing with considerably large item sets. In our initial experiments, we observed that the two distinct training strategies significantly impact the outcomes of the recommendation task, complicating our ability to accurately evaluate the effectiveness of

Table 3: Ablation study on five datasets.

	Metric	w/o CG	w/o B-Enc	w/o $\mathcal{L}_d$	w/o $\mathcal{L}_{cl}$	CaDiRec
ML-1m	HR@10	0.1762	0.2203	0.1757	0.1932	0.2282
	ND@10	0.0861	0.1212	0.0855	0.0982	0.1251
Beauty	HR@10	0.0647	0.0695	0.0644	0.0673	0.0718
	ND@10	0.0355	0.0365	0.0353	0.0359	0.0386
Sports	HR@10	0.0361	0.0399	0.0363	0.0391	0.0426
	ND@10	0.0202	0.0211	0.0199	0.0208	0.0233
Toys	HR@10	0.0698	0.0738	0.0695	0.0721	0.0785
	ND@10	0.0398	0.0419	0.0396	0.0412	0.0441
Yelp	HR@10	0.0304	0.0351	0.0300	0.0312	0.0387
	ND@10	0.0149	0.0172	0.0148	0.0153	0.0197

the contrastive learning approach. To facilitate a fair comparison focused solely on assessing the impact of contrastive learning, it is crucial to standardize the training strategy across all methods. Specifically, we employ the BCE loss with the negative sampling strategy (defined in Equation (8)) for all methods.

5.2 Experimental Results

We run each experiment five times and report the average results. The comparison results across all datasets are presented in Table 2. Based on these results, we make the following observations:

- Our method consistently outperforms all eleven baseline models across all datasets. Additionally, a paired t-test reveals that our method achieves significantly better performance than the second-best result, with a significance level of 0.01.
- Classical methods (BPR-MF, Caser, SASRec, BERT4Rec) that do not employ contrastive learning tend to perform poorly compared to methods that integrate data augmentation and contrastive learning. This suggests that contrastive learning, serving as an auxiliary task, facilitates more comprehensive learning of user sequence representations in the presence of limited data, thereby improving sequential recommendation.
- Our method consistently outperforms contrastive learning-based baselines (S^3-Rec, CL4SRec, CoSeRec, DuoRec, MCLRec) across all metrics on all datasets. CL4SRec introduces three random data augmentation operations for contrastive learning based on SASRec, achieving better performance. CoSeRec takes into account item similarity based on random augmentation, outperforming CL4SRec. DuoRec and MCLRec further improve contrastive learning-based sequential recommendation by incorporating model-level learnable augmentation, resulting in certain improvements. However, all these baseline models neglect context information during augmentation, which may lead to unreasonable positive pairs. Our model, in contrast, leverages context information to guide the generation of augmented views, resulting in superior performance.
- Our model performs significantly better than existing diffusion-based methods (DiffuASR, DreamRec). DiffuASR does not perform well, likely because its augmentation strategy resembles the reverse multi-step sequential recommendation task, which is extremely challenging and prone to introducing noisy data. Furthermore, DiffuASR feeds these extended sequences to the recommendation model, which may lead to error accumulation. DreamRec, on the other hand, directly uses the diffusion model to generate the next item based on historical items, thus performing better than DiffuASR. Unlike these two diffusion-based baselines, our method uses the diffusion model to generate more reasonable augmented user sequences for better contrastive learning. With context guidance, CaDiRec generates alternative items that adhere to the learned context-conditional distribution. The results show that CaDiRec consistently outperforms both diffusion-based baselines across all datasets.

5.3 Ablation Study

In this section, we demonstrate the effectiveness of our model by comparing its performance with five different versions across five datasets. The results are shown in Table 3, where "w/o CG" denotes removing context guidance, "w/o B-Enc" denotes removing the BERT encoder (utilizing an MLP encoder instead), "w/o $\mathcal{L}_d$" means removing the diffusion loss term, and "w/o $\mathcal{L}_{cl}$" means removing the contrastive learning loss term. Specifically, when context information is removed, the model's performance significantly decreases across all datasets, highlighting the substantial contribution of context information. The context guidance allows the model to generate more reasonable augmented views, thus enhancing the quality of contrastive learning. When using an MLP encoder instead of the BERT encoder to model context information, performance also declines, indicating that the BERT encoder is more effective at capturing contextual dependencies, thereby providing better guidance for data augmentation. Furthermore, removing $\mathcal{L}_d$ results in a performance drop because the diffusion model is not involved in the training update, equivalent to random augmentation, which can lead to unreasonable augmented positive sample pairs. Finally, the decline in performance upon removing $\mathcal{L}_{cl}$ underscores the importance of the contrastive learning task, which has been validated in many previous studies [23]. Overall, the results indicate that removing any component reduces the model's performance, thereby validating the effectiveness of each module.

5.4 Hyperparameter Study

In this section, we investigate the impacts of three important hyperparameters (α, β, and ρ) on HR@10 and NDCG@10 across all five datasets. Here, α represents the weight of the contrastive learning loss, β is the weight of the diffusion loss, and ρ is the substitution ratio. The results are shown in Figure 3. We observe that as α increases, HR@10 and NDCG@10 initially rise slightly and then decline across all datasets, with the optimal value at approximately $\alpha = 0.2$. β controls the weight of the diffusion loss in the total loss. As β varies, HR@10 and NDCG@10 values show minimal changes, with an overall trend of initial increase followed by a slight decline. The model achieves optimal performance on all datasets with $\beta \leq 0.4$. As the substitution ratio ρ gradually increases from 0 to 0.8 (note that $\rho = 1$ represents removing context, as shown in the ablation study), the model's performance initially improves and then declines. The optimal performance is observed when ρ is approximately 0.1 to 0.2. This can be explained by the reduction of context

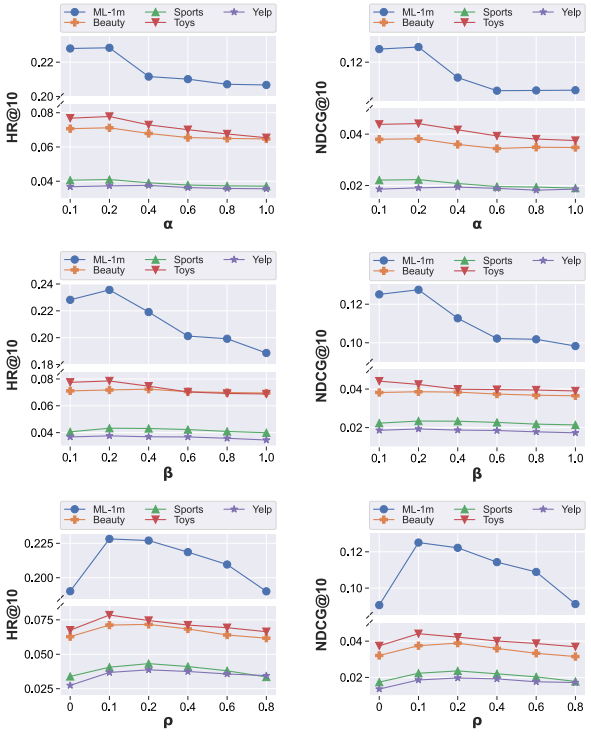


Figure 3: Hyperparameter study of α, β, and ρ on five datasets.

information as ρ increases; without adequate context guidance, the model is unable to generate reasonable positive samples. When $\rho = 0$, no replacements are made, which is equivalent to not using contrastive learning, resulting in poor performance. Therefore, to enhance the effectiveness of contrastive learning, it is advisable to select an appropriate ρ for data augmentation. Additionally, the metrics for the MovieLens dataset vary differently with changes in hyperparameters compared to the other four datasets. This difference is due to the fact that the other four datasets are sparse, while MovieLens is relatively dense.

5.5 Robustness w.r.t. User Sequence Length

To further examine the robustness of our model against varying degrees of data sparsity, particularly its performance with limited interaction records, we categorize user sequences into three groups based on their length and analyze the evaluation results for each group. Figure 4 presents the comparison results on the four sparse datasets (excluding MovieLens, as it is a dense dataset). By comparing our model with representative baseline models, including the strongest baseline MCLRec, we make the following observations: 1) The performance of all models deteriorates as interaction frequency decreases, indicating the influence of data sparsity on model performance. 2) Our model consistently outperforms the baseline models in each user group. Even for the group with the most limited data (sequence length of 5), our model maintains a significant lead, demonstrating the positive impact of our context-aware diffusion-based contrastive learning approach in addressing data sparsity. This finding underscores the robustness of our model across various degrees of data sparsity in user sequences.

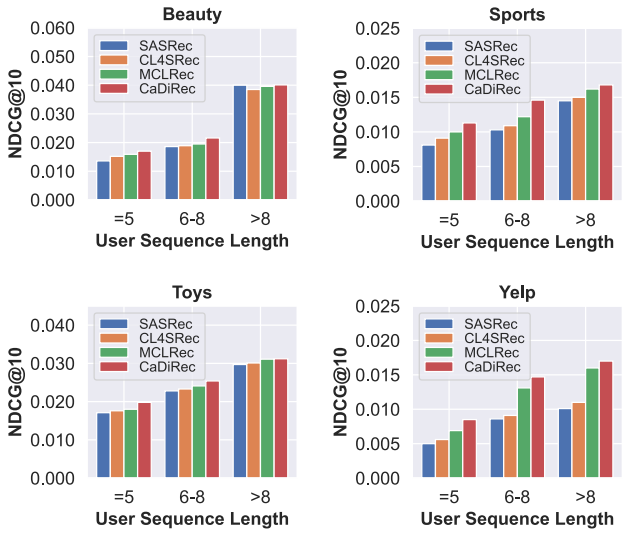


Figure 4: Performance comparison on different user groups.

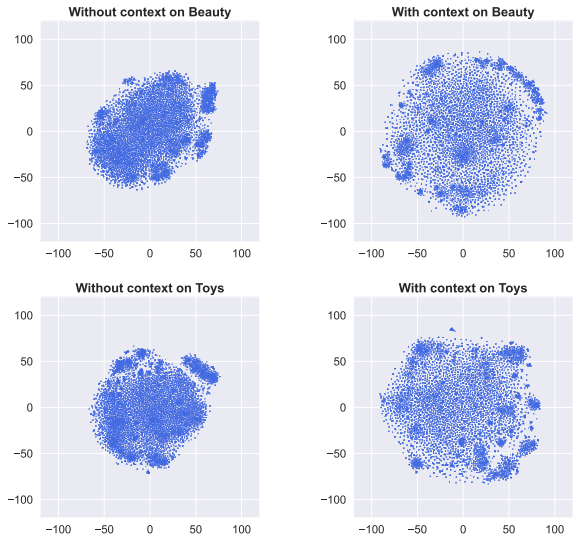


Figure 5: Visualization of learned sequence representations.

5.6 Sequence Representation Visualization

To further analyze the impact of context information on representation learning, we visualize the user sequence representations learned by our model with and without context guidance. For both versions of the model, we train it for 300 epochs in an end-to-end manner and utilize t-SNE [29] to reduce the learned user representations to two-dimensional space. Due to space limitations, the results for Beauty and Toys are presented in Figure 5. Intuitively, the embeddings with and without context exhibit different levels of dispersion in the visualizations. The embeddings without context appear overly compact, while the embeddings with context are comparatively more dispersed, suggesting richer and more informative representations. This pattern is consistent across both

datasets. This may be because augmentations without context resemble random augmentations, which easily generate unreasonable positive sample pairs. In such cases, the contrastive learning objective forcibly brings together dissimilar user sequences, leading to overly compact user representations in the embedding space and even a tendency for representation collapse. Conversely, using context information to guide the generation of augmented views results in more reasonable augmentations, effectively addressing this issue and preventing representation collapse.

6 CONCLUSION

In this paper, we propose a context-aware diffusion-based contrastive learning method for sequential recommendation. We employ a diffusion model to generate more reasonable augmented sequences through conditional generation, thereby improving contrastive learning. We conduct extensive experiments and analyses on five public benchmark datasets. The results demonstrate the advantages of our proposed method over existing baselines.

REFERENCES

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[2] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 15750–15758.

[3] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*. 2172–2182.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.

[6] Hanwen Du, Huanhuan Yuan, Zhen Huang, Pengpeng Zhao, and Xiaofang Zhou. 2023. Sequential Recommendation with Diffusion Models. *arXiv preprint arXiv:2304.04541* (2023).

[7] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2022. DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models. In *The Eleventh International Conference on Learning Representations*.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[9] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.

[10] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 843–852.

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

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[13] Yangqin Jiang, Yuhao Yang, Lianghao Xia, and Chao Huang. 2023. DiffKG: Knowledge Graph Diffusion Model for Recommendation. *arXiv preprint arXiv:2312.16890* (2023).

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

[15] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[16] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. 2020. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966* (2020).

[17] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems* 35 (2022), 4328–4343.

[18] Zihao Li, Aixin Sun, and Chenliang Li. 2023. DiffuRec: A Diffusion Model for Sequential Recommendation. *arXiv preprint arXiv:2304.00686* (2023).

[19] Qidong Liu, Fan Yan, Xiangyu Zhao, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Feng Tian. 2023. Diffusion Augmentation for Sequential Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1576–1586.

[20] Zhiwei Liu, Yongjun Chen, Jia Li, Man Luo, Philip S Yu, and Caiming Xiong. 2022. Improving contrastive learning with model augmentation. *arXiv preprint arXiv:2203.15508* (2022).

[21] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. 2021. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479* (2021).

[22] 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. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 89–98.

[23] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 813–823.

[24] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[25] 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.

[26] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

[27] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[28] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

[29] Laurens Van Der Maaten. 2014. Accelerating t-SNE using tree-based algorithms. *The journal of machine learning research* 15, 1 (2014), 3221–3245.

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

[31] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023. Diffusion Recommender Model. *arXiv preprint arXiv:2304.04971* (2023).

[32] Yu Wang, Zhiwei Liu, Liangwei Yang, and Philip S Yu. 2023. Conditional Denoising Diffusion for Sequential Recommendation. *arXiv preprint arXiv:2304.11433* (2023).

[33] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.

[34] Zihao Wu, Xin Wang, Hong Chen, Kaidong Li, Yi Han, Lifeng Sun, and Wenwu Zhu. 2023. Diff4Rec: Sequential Recommendation with Curriculum-scheduled Diffusion Augmentation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 9329–9335.

[35] Lianghao Xia, Chao Huang, Chunzhen Huang, Kangyi Lin, Tao Yu, and Ben Kao. 2023. Automated Self-Supervised Learning for Recommendation. In *Proceedings of the ACM Web Conference 2023*. 992–1002.

[36] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.

[37] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741* (2021).

[38] Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. 2023. Debiased Contrastive Learning for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2023*. 1063–1073.

[39] Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. 2023. Generate What You Prefer: Reshaping Sequential Recommendation via Guided Diffusion. *arXiv preprint arXiv:2310.20453* (2023).

[40] Yihang Yin, Qingzhong Wang, Siyu Huang, Haoyi Xiong, and Xiang Zhang. 2022. Autogcl: Automated graph contrastive learning via learnable view generators. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36. 8892–8900.

[41] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR*

conference on research and development in information retrieval. 1294–1303.

[42] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1893–1902.

[43] Peilin Zhou, Jingqi Gao, Yueqi Xie, Qichen Ye, Yining Hua, Jaeboum Kim, Shoujin Wang, and Sunghun Kim. 2023. Equivariant contrastive learning for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 129–140.