Context-Aware Diffusion-based Sequential Recommendation

Di You
Worcester Polytechnic Institute
Worcester, USA
dyou@wpi.edu

Kyumin Lee
Worcester Polytechnic Institute
Worcester, USA
kmlee@wpi.edu

***Abstract*—Sequential recommendation aims to recommend the next item that matches a user's interest, based on the sequence of items he/she interacted with before. Although effective, existing work suffers from the following limitations: (1) Existing diffusion-based recommendation methods have undertaken tailored refinements to the diffusion process without considering the difference between recommendation and other tasks, leading to the ignorance of the user's personalized preferences; (2) Self-supervised contrastive learning, widely used to mitigate the data sparsity issue in sequential recommendation, typically employs random augmentation to create multiple views of user sequences. However, random augmentation can disrupt the semantic integrity and interest patterns within the sequence, resulting in semantically divergent augmented views that may misrepresent user preferences. To address these challenges, we propose the Context-Aware Diffusion-based Sequential Recommendation (*CADSR*) model, which leverages context information to generate more semantically consistent positive samples during contrastive learning. This ensures that the model captures both user preferences and their evolution more accurately. Extensive experiments on four public benchmark datasets show that CADSR outperforms 11 state-of-the-art baselines, achieving an average improvement of 10.94% in Recall@10 and 10.54% in NDCG@10 over the best baseline. Source code is available at https://github.com/queenjocey/CADSR.**

***Index Terms*—Diffusion Model, Recommender System, Sequential Recommendation, Contrastive Learning**

I. Introduction

Recommender systems play a pivotal role in enhancing user experience by providing personalized content and suggestions. Among the various types of recommender systems, sequential recommendation systems have gained prominence due to their ability to capture the temporal dynamics of user interactions. However, sequential recommendation is inherently time-sensitive, requiring models that can efficiently handle temporal data while providing personalized and accurate recommendations. In this context, diffusion models, which have shown great success in various domains such as natural language processing (NLP) [1] and computer vision (CV) [2]–[4], offer a promising approach due to their capability in dealing with temporal data and generating diverse representations.

Diffusion models, originally designed as probabilistic generative models, have achieved remarkable results in tasks such as image synthesis, audio generation, and, more recently, recommender systems. These models operate through a denoising process that iteratively removes noise from an initial noisy input to generate high-quality data samples. This iterative refinement introduces a level of uncertainty that aligns well with the inherently diverse and dynamic nature of user interests. The ability of diffusion models to simulate the spread of information and preferences provides a nuanced method for predicting user behavior, making them well-suited for the task of sequential recommendation.

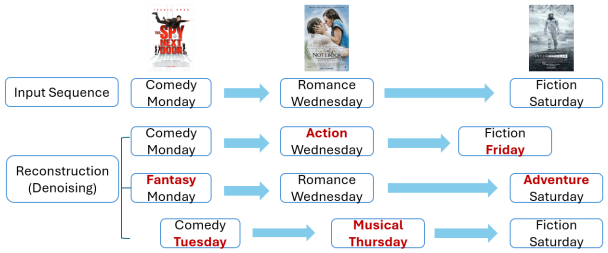


Fig. 1: An example of augmented views (i.e., second, third and fourth sequences) generated during reconstruction/reverse process, serving as positive pairs for contrastive learning.

We use a toy-example in Fig. 1 illustrating how a diffusion model reconstructs Alice's movie preferences. The top/first sequence shows Alice's original movie preferences labeled as the input sequence. Although not shown in this example, in the forward diffusion process, noise is introduced. In the reverse diffusion process, the diffusion model refines these preferences, introducing variations or recovering familiar genres (e.g., changing Romance to Action, Fiction to Adventure, or Monday to Tuesday), as shown in the second, third and fourth sequences in the figure. Simultaneously, the reverse diffusion process generates high-quality augmented views, which enhance contrastive learning by distinguishing similar and dissimilar movie recommendations.

Despite their potential, applying diffusion models to recommender systems presents significant challenges. First, the high degree of personalization required in recommendation tasks necessitates guidance during the diffusion generation process to ensure that the generated recommendations align closely with individual user preferences. This need for guidance contrasts with their application in NLP or CV domains, where the focus is often on generating generalizable outputs. Second, the computational cost associated with diffusion models is substantial. The extensive diffusion steps required for high-quality

outcome generation lead to significant time and resource consumption, presenting a big challenge for recommender systems that must operate in real-time and deliver timely responses to user queries.

To address these challenges, we propose a novel approach that leverages the strengths of diffusion models while incorporating mechanisms to enhance their efficiency and personalization for sequential recommendation tasks. Our motivation for proposing this work is multi-fold: (1) We harness the ability of diffusion models to handle complex sequence data and their strength in generating diverse and uncertain representations. This approach enables the model to effectively capture the dynamic and varied nature of user preferences; (2) We incorporate context-aware guidance into the diffusion generation process. By utilizing intermediate steps in the reverse diffusion process, we generate augmented views for contrastive learning. This ensures that the generated recommendations align closely with the user's contextual preferences, thereby enhancing their personalization and relevance.

Our experimental results demonstrate that the proposed Context-Aware Diffusion-based Sequential Recommendation (CADSR) model significantly improves the performance of sequential recommendation tasks. The model outperforms existing state-of-the-art methods in terms of both accuracy and diversity of recommendations, showcasing its ability to generate highly personalized suggestions that cater to the dynamic interests of users. In summary, this work makes several important contributions:

- We demonstrate the effective application of diffusion models to sequential recommendation tasks, leveraging their strengths in handling temporal data and generating diverse representations.
- By introducing context-aware guidance during the diffusion process, we enhance the personalization and relevance of the recommendations
- Experimental results show that the proposed CADSR model outperforms 11 state-of-the-art baselines across 4 real-world datasets.

These contributions collectively advance the state-of-the-art in sequential recommendation by offering a robust and efficient solution that leverages the unique capabilities of diffusion models to provide highly personalized recommendations.

II. Related Works

A. 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 [5], [6]. 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 [7]–[9], leading to remarkable achievements. Subsequently, the introduction of the attention mechanism has significantly enhanced recommendation performance. SASRec [10], for instance, is the pioneering work that employs the self-attention mechanism to model the evolution of user preference. Following that, BERT4Rec [11] is proposed to use a bidirectional self-attention encoder to capture context information of the user sequence. Recently, many self-attention-based and graph-based methods have made improvements to existing approaches, achieving notable progress [12]–[16].

B. 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 CV [2]–[4] and NLP [1].

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 [17]–[22]. When it comes to sequential recommendation, researchers design informative contrastive learning objectives for learning better user representations from historical interactions. S^3-Rec [23] introduces a method that incorporates auxiliary self-supervised objectives to learn the correlations among items, attributes, and segments. CL4SRec [24] designs three data-level augmentation operators, namely crop, mask, and reorder. CoSeRec [25] suggests substituting a specific item in the sequence with a similar item. Later, DuoRec [26] 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 augmentation at the model level, lacking the ability to maintain semantic consistency. In addition, ICLRec [27] attempts to extract user intent from sequential information and subsequently performs contrastive learning between user representations and intent representations. ECL-SR [28] designs different contrastive learning objectives for augmented views at different levels. MCLRec [29] further combines data-level and model-level augmentation strategies. It applies random data augmentation, as proposed by CL4SRec, to the input sequence and then feeds the augmented data into MLP layers for model-level augmentation.

However, the 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 overlook context information, which is crucial for preserving the semantic consistency.

C. Diffusion Models

Diffusion Models have gained significant prominence as a dominant approach in diverse generative tasks, such as image synthesis [30]–[32] and text generation [33], [34]. They demonstrate superior generative capabilities compared to alternative models such as GANs [35] and VAEs [36], 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 [37]–[41] 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 [42], [43] 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.

III. Preliminaries

A. Task Formulation

Let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item set, respectively. In this paper, we aim to estimate the users' preferences on next-item through a neural recommender (e.g., a diffusion model), which can recommend the top-k items for a target user u.

Specifically, we represent historical interaction sequence of user u as $v^u = [v_1^u, v_2^u, \ldots, v_{n-1}^u]$, and denote the subsequent consumed item of the sequence as v_n^u. Let D stand for all the sequences within the training data, and D_t denotes test sequences(i.e. in our context, refers to v_n^u). Each item v_i^u is represented as an embedding vector x_i^u and the interaction sequence is mapped to a matrix $\mathbf{x} \in \mathbb{R}^{(n-1)\times d}$, where d is the embedding dimension. The matrix serves as the input to the diffusion process.

The goal this work is to predict the next item at time step n according to $n-1$, which can be formulated as:

$$\arg\max_{v_i^u \in \mathcal{I}} P(v_n^u = v_i^u | \mathbf{x}^u), \tag{1}$$

where the probability P represents the likelihood of item v_i^u being the next item, conditioned on user's historical interaction matrix $\mathbf{x}^u$.

B. Diffusion Model

A vanilla diffusion model involves two major components: the forward process and reverse process, which uses latent variable modelling to enable the progressive generation of refined representations [44].

Forward Process: In the forward process of the diffusion model, a data point initially sampled from a real-world distribution, $\mathrm{x}_0 \sim q(\mathbf{x})$, is progressively corrupted through a Markov chain into a standard Gaussian noise, represented as $\mathbf{x}_T \sim \mathcal{N}(0, \mathrm{I})$. For each forward step $t \in [1, 2, \ldots, T]$, this corruption transforms the original representation $\mathbf{x}_0$ into a noisy representation $\mathbf{x}_t$ at each step t:

$$\begin{aligned} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) &= \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right), \\ &= \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathrm{I}) \end{aligned} \tag{2}$$

where $\mathcal{N}(x; \mu, \sigma^2)$ is a Gaussian distribution with a mean $\mu = \sqrt{1-\beta_t}$ and variance $\sigma^2 = \beta_t$, $\mathbf{x}_t$ is sampled from this Gaussian distribution, $\beta_t \in (0, 1)$ controls the level of the added noise at the t-th diffusion step and $\mathbf{I}$ is an identity matrix.

This approach shows the flexibility of the direct sampling of $\mathbf{x}_t$ conditioned on the input $\mathbf{x}_{t-1}$ at an arbitrary diffusion step t by sampling from a random Gaussian noise ϵ.

Reverse Process: The primary purpose of the diffusion models is to learn a denoising model capable of removing the noise added to the data and to gradually recover the initial distribution. Indeed, once a noisy embedding $\mathbf{x}_t$ is obtained, the reverse process aims to denoise this $\mathbf{x}_t$, with a trajectory towards the direction of $\mathbf{x}_0$ and to gradually recover the initial representation $\mathbf{x}_0$. The transition $(\mathbf{x}_t \to \mathbf{x}_{t-1} \to \ , \ldots, \to \mathbf{x}_0)$ is defined as follows:

$$\begin{aligned} p(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) &= \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_{\theta t}(\mathbf{x}_t, \mathbf{x}_0), \beta_t \mathbf{I}\right) \\ \mu_{\theta t}(\mathbf{x}_t, \mathbf{x}_0) &= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \\ \beta_t &= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \end{aligned} \tag{3}$$

where μ and σ^2 are the mean and variance of the user/item embeddings, respectively; $\alpha_t = 1 - \beta_t$ while $\bar{\alpha}_t = \prod_{t'=1}^{t} \alpha_t$. The learning of the mean μ_θ is based on a neural network f_θ parameterized by θ. As such, to effectively recover $\mathbf{x}_0$, neural models, such as a Transformer [45] and a U-Net [32], are commonly used in practice [30].

IV. Methodology

In this section, we present the comprehensive methodology of our proposed Context-Aware Diffusion-based Sequential Recommender (*CADSR*). We provide an overview of the proposed framework in Fig. 2. Diffusion model was originally proposed to model continuous data, such as images and audios. To adapt diffusion model for sequential recommendation, we start the pipeline with a sequence encoder that transits the initial input into latent space. In the forward pass, we gradually add Gaussian noise to the encoded embeddings, creating noisy representations, and denoise these embeddings step-by-step under the contextual guidance of user history and the diffusion steps. The final reconstructed embeddings are rounded to predict the most relevant items for the target user.

A. Input Layer

Given users' interaction sequence of v in one-hot encoding, we first map it into hidden representation e, and then conduct the following learning procedure. Unlike existing work [39], [40] which mostly follow the common practice that directly

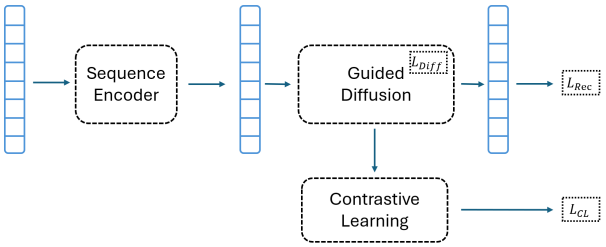


Fig. 2: Overview of proposed *CADSR* framework.

corrupts the initial state, we reweight the sequence with its time interval and process it with a Transformer-based sequence encoder before the diffusion process.

Time-aware reweigting As we formulated our task in Sec.III-A, a big difference that distinguish sequential recommendation from general recommendation is that the order of interactions and its time interval play a crucial role in predicting users' next item preference. Thus, we reweight each input with its timestamp $TS_u = ts_1^u, ts_2^u, \ldots, ts_{n-1}^u$. Each item in the interaction sequence is assigned a weight that is proportional to its temporal distance from the most recent interaction. Formally, we define the weight for an item v_j^u at timestamp ts_j^u as $w_j^u = w_{min} + \frac{ts_j^u - ts_1^u}{ts_{n-1}^u - ts_1^u}(w_{max} - w_{min})$, where w_{min} and w_{max} are predefined lower bound and upper bound of user interaction sequence, and ts_1^u and ts_{n-1}^u are the timestamp of the first and last interactions in the sequence.

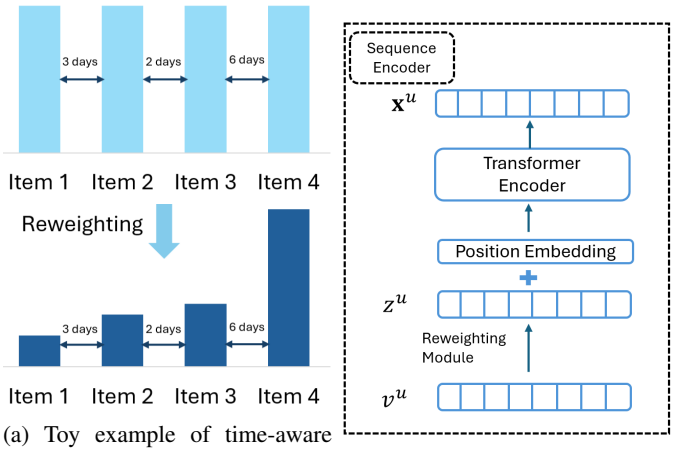
We presents the effect of time-aware reweighting strategy in Fig. 3a. Before reweighting (i.e. the upper-left subfigure), all items in the user interaction sequence are treated with equal importance, represented by bars of uniform height (i.e., uniform weight). While after reweighting based on time intervals(i.e. the lower-left subfigure), the importance of each item is adjusted according to the temporal proximity to the most recent interaction. The most recent items is assigned with highest weights and items closer in time were assigned with similar weights, thus resulting in varying bar heights. This reweighting mechanism prioritizes more recent interactions, thereby capturing the evolving nature of user preferences for more accurate next-item prediction. The resulting reweighted sequence $\mathbf{z^u} = [\mathbf{z}_1^u, \ldots, \mathbf{z}_{n-1}^u]$ is computed as $\mathbf{z}_j^u = v_j^u \cdot w_j^u$, where v_j^u is the one-hot encoded representation of the item.

Sequence Encoder As shown in Fig. 3b, we augmented time-reweighted sequence with sinusoidal position embedding [45]. The position embedding is adjusted to have the same units as the input, then add these parts element-wisely.

The output vector is then fed into a Transformer block to learn the user sequence representations. We keep the consistent architecture and configuration with the standard *Transformer* block, consisting of a multi-head self-attention layer and a feed-forward layer, defined as:

$$\mathbf{x} = Transformer(\mathbf{z}) \tag{4}$$

, where $\mathbf{x} \in \mathbb{R}^{(n-1)\times d}$ denotes the hidden states of user sequence. And we use the last layer outputs as the contextualized representation of user interaction sequence.



(a) Toy example of time-aware reweighting for input sequence.

(b) Sequence Encoder

Fig. 3: Fig. (3a) illustrate proposed time-aware reweighting mechainism and Fig. (3b) describe the architecture of input sequence encoder.

B. Diffusion-based Sequential Recommendation

We depict the forward and reverse process along with diffusion-augmented contrastive learning in Fig. 4.

1) Forward Process: Given pre-trained user embedding $\mathbf{x}^u$, we begin the forward process and continuously incorporate Gaussian noises into with adjustable scales and steps. In the forward diffusion process, noisy embeddings $\mathbf{x}_t^u$ are generated at each step t.

The value of β_t is generated from a pre-defined noise schedule β which arranges how much noise is injected at which step. The common noise schedule includes square-root [33], cosine [30], and linear [46]. According to [30], at an arbitrary diffusion step t, we can derive $\mathbf{x}_t$ conditioned on the input $\mathbf{x}_0$ in a straightforward way following the Markov chain process. Then Eq. 2 can be rewritten as follows:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\overline{\alpha}_t}\mathbf{x}_0, (1-\overline{\alpha}_t)\mathbf{I}\right) \tag{5}$$

$$\overline{\alpha}_t = \prod_{s=1}^{t} \alpha_s, \quad \alpha_s = 1 - \beta_s \tag{6}$$

A proper noise schedule is important in forward process to control the noise injection. In this work, we follow [39] and adopt a *linear variance* noise schedule as:

$$1 - \bar{\alpha}_t = s \cdot \left[\alpha_{\min} + \frac{t-1}{T-1}(\alpha_{\max} - \alpha_{\min})\right], \tag{7}$$

where α_{min} and α_{max} are the minimum and maximum of the noise correspondingly, t is the current forward step, T is the total forward step, and $s \in (0, 1)$ controls the noise scale.

Denoting the maximum diffusion step as t, in the training process, we adopt an importance sampling strategy [46] to emphasize critical steps. For each item, we sample a diffusion step s according to a predefined probability distribution $p(s)$, where $p(s)$ is proportional to the magnitude of the loss at step

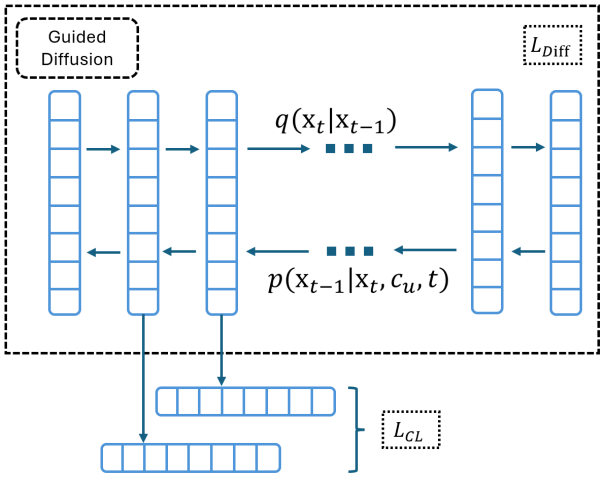


Fig. 4: Diffusion processes and diffusion augmented contrastive learning in *CADSR*.

s, where $s = \lfloor s' \rfloor, s' \sim U(0, t)$. The sampling strategy ensures that steps with higher loss, which is more difficult to learn, are sampled more frequently. Following Eq. 5, we can generate $\mathbf{x}_s$ in a range of $1 \leq s \leq T$. Following reparameterization trick [30], [47], we could generate $\mathbf{x}_s$ as follows:

$$\mathbf{x}_s = \sqrt{\overline{\alpha}_s}\mathbf{x}_0 + \sqrt{1 - \overline{\alpha}_s}\epsilon \tag{8}$$

This approach enables more efficient and robutst training. We present detailed evaluation on the influences of different noise schedules in Section V-E1.

2) Reverse Process: In the reverse diffusion process, the noisy embedding $\mathbf{x}_t^u$ are iteratively denoised, resulting in reconstructed embeddings $\hat{\mathbf{x}}_t^u$. The process is natuarally intractable as $\mathbf{x}_0$ is required at each reverse step, therefore we need a well-designed approximator for estimation. The approximator serves as the backbone model to iteratively recover user representations during the reverse process. While U-net [48] is widely used in image domain, we choose Transformer [45] as our approximator, which has proven to be effective in sequence data modeling.

Context-aware guiding The original formulation of reverse process is to recover the representation from pure Gaussian noise. However, in this context, we expect the engagement of users' historical interaction to generate high-quality embedding. In particular, we introduce two key components as guidance: 1) users' historical interaction sequence c_u, 2) diffusion step t. The noisy embedding x_t^u is denoised to reconstructed $\hat{\mathbf{x}}_t^u$ at each step:

$$\hat{\mathbf{x}}_0^u = Transformer(\hat{\mathbf{x}}_t^u, c_u, t) \tag{9}$$

$$\hat{\mathbf{x}}_{t-1} = \tilde{\mu}_t(\hat{\mathbf{x}}_t, \mathbf{x}_0) + \tilde{\beta}_t\epsilon' \tag{10}$$

where $\tilde{\mu}_t(\hat{\mathbf{x}}_t^u, \mathbf{x}_0) = \frac{\sqrt{\overline{\alpha}_{t-1}}\beta_t}{1-\overline{\alpha}_t}\mathbf{x}_0^u + \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}\hat{\mathbf{x}}_t$ and $\tilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$, and $\epsilon' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $\hat{\mathbf{x}}_{t-1}^u$ generation. Repeat the above process until we arrive at $\hat{\mathbf{x}}_0$. The reverse phase can be illustrated in Algorithm 1.

Unlike tasks in other domains that general output with high probability are usually expected, in recommendation tasks, we want to keep personalized preference of each user, therefore, we anticipate that these contextual information can serve as guiding signals to help the model better preserve user preferences and temporal dynamics of their behavior, allowing for more informed and personalized predictions.

In corresponding to the transition at encoding layer, we use a rounding function to map the learned embedding back into embedding space, defined as:

$$\textbf{Rounding}(\hat{\mathbf{x}}_0^u, v_i) = \hat{\mathbf{x}}_0^u \cdot v_i, \quad i \in \mathcal{I} \tag{11}$$

where $\hat{\mathbf{x}}_0^u$ denotes the reconstructed user embedding conditioned on initial input $\mathbf{x}_0^u$ and current step embedding. We rank the similarity score and select top-k candidate for recommendation.

3) Training and Optimization: Our total loss consist of three parts:

$$\mathcal{L}_{Total} = \mathcal{L}_{Diff} + \alpha \cdot \mathcal{L}_{Rec} + \beta \cdot \mathcal{L}_{Cl} \tag{12}$$

Diffusion Optimization To ensure conditional generation, it is essential to minimize the variational lower bound of the predicted user and item embeddings. Following previous methods [39], the reconstruction loss can be formulate as:

$$\mathcal{L}'_{vlb} = \mathop{\mathbb{E}}_{q_\phi(\mathbf{x}_0|\mathbf{v})} \left[\mathcal{L}_{vlb}(\mathbf{x}_0) + \log q_\phi(\mathbf{x}_0|\mathbf{v}) - \log p_\theta(\mathbf{v}|\mathbf{x}_0)\right]. \tag{13}$$

The above equation can be further simplified as:

$$\begin{aligned}\mathcal{L}_{Diff} = \mathbb{E}_{q(\mathbf{x}_T|v)} \Bigg[\left(\sum_{n=2}^{N} \left\| \mathbf{x}_T^0 - f_\theta(\mathbf{x}_T^n, n) \right\|^2 \right) \\ + \left\| \mathbb{E}(v) - f_\theta(\mathbf{x}_T, 1) \right\|^2 \Bigg]\end{aligned} \tag{14}$$

where the first term and second term calculate the MSE between the predicted hidden representation and the original representation, $f_\theta(\cdot)$ denotes the approximator which is Transformer in this context.

Recommendation Loss Solely relies on the diffusion generation may lead to trivial representations because that the $\mathcal{L}_{Diff}$ tends to maximize the similarity between the predicted mean and corrupted target item embedding. However, as a ranking task, we anticipate to avoid high score for all items. Consequently, we introduce L_{Rec} to guide the prediction of recommendation task that also leverage dissimilar pairs.

$$\mathcal{L}_{Rec} = - \sum_{(i,j^+,j^-)} \log \sigma \left(\mathbf{x}_{ij^+} - \mathbf{x}_{ij^-} \right) \tag{15}$$

Given recommendation data with the triplet (u, j^+, j^-), item j^+ and item j^- are the positive item and negative item of user u, respectively. Note that we only conduct denoising for user and positive interacted item since denoising negative item is relatively insignificant due to the random negative sampling mechanism.

Algorithm 1 CADSR Training

Require: interaction data $\bar{\boldsymbol{D}}$, pre-trained user embedding $\boldsymbol{x_0^u}$, diffusion step T, user reconstruction Transformer $\boldsymbol{f_\theta}$, noise schedule β_s

repeat
  Sample a batch of interactions $\boldsymbol{D} \subset \bar{\boldsymbol{D}}$.
  for all $(u, i, j) \in \boldsymbol{D}$ **do**
    Sample diffusion step $t \sim \mathcal{U}(1, T)$
    Sample random Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
    Compute $\boldsymbol{x_t^u}$ given $\boldsymbol{x_0^u}$ and t via $q(\boldsymbol{x_t^u}|\boldsymbol{x_0^u})$ in Eq. (5);
    Reconstruct $\tilde{\boldsymbol{x}}_{\boldsymbol{0}}^{\boldsymbol{u}}$ and through $\boldsymbol{f_\theta}$;
    Calculate $\mathcal{L}_{\text{final}}$ by Eq. (12);
    Take gradient descent step on $\nabla_\theta(\mathcal{L}_{\text{final}})$ to optimize θ;
  end for
until converged

Ensure: optimized θ.

Contrastive Learning To avoid drift of prediction because of noise perturbation, we also designed a contrastive learning module where the diffusion model serves as augmented view generator. The reverse process is naturally a denoising process, and generates multiple semantically consistent augmented views of the user sequence at intermediate steps. We consider these views as positive samples, and the constrastive learning module should maximize agreement between similar representations while distancing from dissimilar representations. We use the InfoNCE loss and define as follows:

$$\mathcal{L}_{cl}^{t} = \frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\hat{\mathbf{x}}_t^{i\top}\mathbf{x}_t^i/\tau)}{\sum_j \exp(\hat{\mathbf{x}}_t^{i\top}\mathbf{x}_t^j/\tau) + \sum_j \mathbb{1}_{[j\neq i]}\exp(\hat{\mathbf{x}}_t^{i\top}\hat{\mathbf{x}}_t^j/\tau)}, \tag{16}$$

,where $\hat{\mathbf{x}}_{\mathbf{t}}^{\mathbf{i}}$ is the output of conditional denoising decoder of position i at denoising step t, $\mathbf{x}_{\mathbf{t}}^{\mathbf{i}}$ is the output of diffuser at diffusion step t at position i of the sequence. The diffusion-augmented contrastive learning mitigates the impact of noise.

4) Inference: During inference phase, our *CADSR* leverages a modified reverse diffusion process to reconstruct user preference efficiently. We start the reverse process from a midpoint ($T' < T$) rather than pure noise, preserving critical personalized information of user preference. This approach reduces noise while maintaining contextual features, ensuring accurate reconstruction. We also employs a deterministic sampling strategy that skips unnecessary intermediate steps, and accelerate inference without sacrificing performance. We manage to strike a balance between computational efficiency and high-quality recommendations. See details at Algorithm 2

V. Experiments

A. Experimental Settings

1) Datasets: We conduct experiments on four real-world public datasets, including MovieLens, Amazon-Beauty, Amazon-Book, Amazon-Music. The statistics of these datasets

Algorithm 2 CADSR Inference

Require: target item embedding $v_t \sim \mathcal{N}(0, \mathbf{I})$, sequence length n, total reverse step t, noise schedule β_n, hyperparameter λ sampling, reconstruction Transformer $\boldsymbol{f_\theta}$, inference user sequence $v^u = v_1^u, v_2^u, \ldots, v_{n-1}^u, [unk]$

Sample a batch of users $\boldsymbol{U} \subset \bar{\boldsymbol{U}}$.
for all $u \in \boldsymbol{U}$ **do**
  for $t = T, \ldots, 1$ **do**
    Compute $\hat{\boldsymbol{x}}_{\boldsymbol{t}}^{\boldsymbol{0}}$ via Eq. (9);
    Compute $\hat{\boldsymbol{x}}_{\boldsymbol{t}}^{\boldsymbol{N}}$ given $\hat{\boldsymbol{x}}_{\boldsymbol{0}}^{\boldsymbol{i}}$ and T via $q(\hat{\boldsymbol{x}}_{\boldsymbol{T}}^{\boldsymbol{i}}|\hat{\boldsymbol{x}}_{\boldsymbol{0}}^{\boldsymbol{i}})$ in Eq. (3);
    Compute $\hat{\boldsymbol{x}}_{\boldsymbol{t-1}}^{\boldsymbol{i}}$ from $\hat{\boldsymbol{x}}_{\boldsymbol{t}}^{\boldsymbol{i}}$ and $\hat{\boldsymbol{x}}_{\boldsymbol{0}}^{\boldsymbol{i}}$ via Eq. (10);
  end for
  Rounding via Eq. (11) to get the ideal item embedding $\boldsymbol{x}$;
end for

Ensure: the ideal item embedding $\boldsymbol{x}$.

TABLE I: Dataset description.

Datasets	#Users	#Items	#Actions	Avg. Length	Density
ML-1M	6,040	3,953	1,000,209	163.6	4.19%
Amazon-Beauty	22,363	12,101	198,502	8.8	0.07%
Amazon-Music	75,601	64,227	198,502	8.3	0.02%
Amazon-Book	604,592	368,743	8,898,041	10.4	0.004%

are shown in Table I. 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, Book, and Music datasets are obtained from Amazon [49][2], one of the largest e-commerce platforms globally. We adopt the same preprocessing method as employed in numerous previous studies [24], [25], filtering items and users with fewer than five interaction records.

2) Evaluation Metrics: To evaluate the performance of our model and baseline models, we employ widely recognized evaluation metrics: Recall and Normalized Discounted Cumulative Gain (NDCG), and report values of Recall@k and NDCG@k for k=10 and 20. 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.

3) Baseline Methods: To ensure a comprehensive assessment, we compare our method with 11 baseline methods, which can be divided into four categories: conventional recommender (BPR-MF [50], LightGCN [12]), sequential recommender (SASRec [10], Bert4Rec [11], Caser [7]), generative recommender (SVAE [51], ACVAE [52], CODIGEM [53]), and diffusion-based recommender (DiffRec [39], DreamRec [39], Diff-MSR [24]).

4) Implementation Details: We implement all baseline methods according to their released code. The embedding size

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

TABLE II: Performance comparison of different methods on four datasets. The best performance is bolded, and the runner-up is underlined. Our proposed *CADSR* achieves state-of-the-art results among all baseline models, as confirmed by a paired t-test with a significance level of 0.01.

		Conventional Recommender		Sequential Recommender			Generative Recommender			Diffusion-based Recommender				
		mf	LightGCN	SASRec	Bert4Rec	Caser	SVAE	ACVAE	CODIGEM	DiffRec	DreamRec	Diff-MSR	**Ours**	Improvement
Amazon Book	Recall@10	0.0437	0.0534	0.0592	0.0611	0.0621	0.0571	0.0576	0.0581	0.0671	0.0695	0.0627	**0.072**	3.60%
	NDCG@10	0.0264	0.0325	0.0412	0.041	0.0405	0.0398	0.0395	0.0292	0.0451	0.0484	0.0442	**0.0552**	14.05%
	Recall@20	0.0852	0.1007	0.1115	0.1260	0.1260	0.1129	0.1113	0.1104	0.1374	0.1349	0.1302	**0.1385**	8.01%
	NDCG@20	0.0607	0.0632	0.0909	0.0882	0.0904	0.0841	0.0890	0.0600	0.0874	0.0913	0.0812	**0.1063**	16.43%
MovieLens-1M	Recall@10	0.0876	0.0987	0.0992	0.1022	0.1045	0.0943	0.0958	0.0972	0.1058	0.1011	0.1007	**0.1231**	21.76%
	NDCG@10	0.0749	0.0833	0.0874	0.0892	0.0902	0.0866	0.0859	0.0837	0.0921	0.0904	0.0896	**0.1022**	13.05%
	Recall@20	0.1808	0.1889	0.2039	0.2011	0.2168	0.1947	0.1967	0.1971	0.2011	0.2162	0.2057	**0.2205**	17.07%
	NDCG@20	0.1490	0.1629	0.1796	0.1853	0.1751	0.1639	0.1697	0.1592	0.1844	0.1823	0.1862	**0.1891**	2.55%
Amazon Beauty	Recall@10	0.0444	0.0525	0.0671	0.595	0.0601	0.0642	0.0653	0.0671	0.611	0.715	0.603	**0.0764**	6.85%
	NDCG@10	0.0335	0.0531	0.0522	0.0552	0.0615	0.0613	0.0629	0.0667	0.0673	0.0659	0.0644	**0.0695**	3.26%
	Recall@20	0.0642	0.0746	0.0797	0.0751	0.0785	0.0739	0.0721	0.0719	0.0802	0.0786	0.0747	**0.0871**	8.6%
	NDCG@20	0.0567	0.0618	0.0657	0.0627	0.0619	0.0604	0.0648	0.0695	0.0700	0.0697	0.0693	**0.0745**	6.43%
Amazon Music	Recall@10	0.0642	0.0768	0.0727	0.0833	0.0863	0.0725	0.0744	0.0792	0.0874	0.0902	0.0822	**0.1006**	11.53%
	NDCG@10	0.0389	0.0522	0.0505	0.0536	0.054	0.0487	0.0579	0.0585	0.0521	0.0611	0.0523	**0.0683**	11.78%
	Recall@20	0.1370	0.1441	0.1492	0.1625	0.1680	0.1468	0.1517	0.1660	0.1659	0.1801	0.1648	**0.1853**	10.30%
	NDCG@20	0.0756	0.1125	0.1060	0.1006	0.1097	0.1031	0.1067	0.1134	0.0944	0.1018	0.0979	**0.1232**	12.31%

for all methods is set to 64. Our method utilizes a bidirectional Transformer architecture for the sequential recommenders, comprising 1 layer and 2 attention heads each layer. The total number of diffusion steps is set to a fixed value of 1000, and we adopt a linear noise schedule in practical use, which improves convergence (see detailed discussion in V-E1). Additionally, we sample the reverse step to 20 during inference without significant performance degradation and facilitates inference efficiency. 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]. 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 work [10], we set the maximum sequence length to 50 for the three Amazon datasets 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 maximum sequence length. For the recommendation task, we employ the negative sampling strategy (Eq. 15) for all methods during training. Specifically, for each positive sample, a negative sample is randomly selected and optimization is performed using the Bayesian Personalized Ranking (BPR) loss.

B. Overall Performance

In this section, we compare the overall performance of our model and other baseline methods on four benchmark datasets. We report the detailed results in Table II. We have the following observations

- Our proposed CADSR consistently outperforms all baseline methods across multiple scenarios from e-commerce datasets (e.g. Amazon) to movie dataset (e.g. MovieLens). On average, our proposed model improved 10.94% at Recall@10 and 10.54% at NDCG@10 compared with the best baseline. The results confirm the consistent effectiveness of our proposed method against all baselines.
- Conventional methods perform poorly compared to sequential baselines, which achieve better results. This is likely due to the sequential baselines' explicit incorporation of temporal and sequential features from the input data, which are crucial for capturing the inherent sequential patterns in user behavior.
- Generative and diffusion-based methods demonstrate competitive performance, particularly with sparse datasets. This is attributed to the generative models' ability to introduce uncertainty during training, which helps capture the complexity of user behavior. However, generative baselines often struggle to learn high-quality user representations in many scenarios, likely due to unstable training dynamics and limited capacity to effectively handle long-tail user sequences.
- Diffusion-based methods leverage a powerful framework that models item representations as probability distributions, effectively capturing the uncertainty in user behavior. Additionally, these methods learn the input sequence in an autoregressive manner, which allows for better modeling of sequential features.
- We attribute the success of our proposed CADSR to several key factors: 1) The preprocessing steps of temporal reweighting and sequential encoding effectively preserve dependencies within the sequence, 2) The guiding signal in the reverse process helps retain user preferences, and 3) The high-quality augmented views generated through the diffusion process, combined with contrastive learning, enhance the model's robustness.

C. Ablation Study

To verify the effectiveness of each component in our proposed CADSR, we design several variants as listed below:

- w/o Contextual Guidance: Contextual guidance c_u and diffusion step t are removed in reverse pass.
- w/o L_{Rec}: The recommendation loss is removed.
- w/o L_{Cl}: The contrastive learning loss is removed

TABLE III: Ablation Study.

	Metrics	Full	w/o Contextual Guidance	w/o L_Rec	w/o L_CI	w/o Time-interval Reweighting	w/o Sequence Encoding
Amazon Book	Recall@10	**0.072**	0.0689	0.0692	0.0712	0.0704	0.0701
	NDCG@10	**0.552**	0.0501	0.521	0.549	0.0533	0.0545
ML-1M	Recall@10	**0.1231**	0.1102	0.1121	0.1159	0.1162	0.1134
	NDCG@10	**0.1022**	0.0844	0.0936	0.0971	0.0964	0.0952
Amazon Beauty	Recall@10	**0.0764**	0.0611	0.0697	0.0711	0.0682	0.0721
	NDCG@10	**0.0695**	0.0627	0.0651	0.0654	0.0663	0.0647
Amazon Music	Recall@10	**0.1006**	0.0922	0.0969	0.0923	0.0955	0.0914
	NDCG@10	**0.0683**	0.061	0.0622	0.0677	0.0652	0.0651

- w/o Time-interval Reweighting: The time-aware reweighting block is removed.
- w/o Sequence Encoding: Sequential encoding layer is removed.

We compare the performances of these variants with the default setting. From the experiment results in Table III, we have following observations:

The performance decreases when contextual guidance is removed across all dataset. In particular, we can observe a noticeable drop in NDCG metric, indicating that the guidance signal plays a critical role in generating personalized recommendation. Removing the reconstruction loss (w/o L_{Rec}) also reduces performance in all datasets. As for contrastive learning, it also contributes positively to entire framework, as seen in the results. Because that our contrastive learning module is built on the high-quality augmented view generated by diffusion model, which enhances recommendation robustness. We observe that the effect of time-aware reweighting is more data-dependent compared to other factors. We also observe that the model benefits from sequence encoding. Removing it reduces performance across all datasets, particularly in Amazon Music and beauty datasets, where there's a notable decrease in NDCG.

To sum up, the full model outperforms all of its ablated versions across all datasets, confirming the importance of each component. Contextual guidance appears to contribute the most significant improvements in performance, as its removal results in the largest drops.

D. Cross-domain Recommendation

One of the benefits of using diffusion model is that by applying the reverse diffusion process, the model recovers useful user preferences from noisy or incomplete data, improving the quality of knowledge transfer in cross-domain scenarios [54]. In this case, we anticipate that the diffusion-based recommender can capture complex patterns and generate comprehensive user preference across both observed and unobserved items, which helps the model to be more generalized, especially when data from the target domain is limited.

Cross-domain recommendation (CDR) aims to transfer knowledge from a source domain to a target domain, where the source domain typically contains richer data and the target domain contains sparser data. Theoretically, the knowledge transfer process can benefit from the diffusion process, which transfers data across distributions. This is analogous to inferring user preferences from sparse interactions, making it ideal for stable predictions even for cold-start users.

TABLE IV: Performance of CADSR on Cross Domain datasets.

Source Domain	Target Domain	Training Set	Methods	Recall@10	NDCG@10
Music	Book	Mixed user sequence	DiffRec	0.0682	0.049
			CADSR	**0.0714**	**0.056**
		Overlapped user sequence	DiffRec	0.0573	0.036
			CADSR	0.0631	0.042
Book	Music	Mixed user sequence	DiffRec	0.0921	0.0611
			CADSR	**0.1021**	**0.0652**
		Overlapped user sequence	DiffRec	0.0769	0.0588
			CADSR	0.0825	0.0604

We conduct the experiments following classic CDR task settings [55], [56]. Specifically, we perform the CDR task on the Amazon Music and Amazon Books datasets, where there are more than 17,000 overlapping users. We design two variant settings to generate cross-domain behavior sequence: 1) by merging the behavior sequences of each user from two single domains in chronological order (called mixed user sequence), and 2) by using the behavior sequences of *only overlapping users* from both single domains, sorted in chronological order (called overlapped user sequence), to predict the next-item recommendation in the target domain.

The results presented in Table IV demonstrate that our proposed *CADSR* model consistently outperforms *DiffRec*, a state-of-the-art competitive baseline, across various settings, indicating its effectiveness in handling other recommendation tasks, such as cross-domain sequential recommendation. We hypothesize that the inferior performance of *DiffRec* may be attributed to the noise introduced during knowledge transfer. Additionally, the experiment utilizing only overlapped user sequences performs worse than the mixed user sequence setting, which is expected given the extreme data sparsity. In the first experimental setting (i.e., mixed user sequence), the interactions from the source domain serve as auxiliary data, effectively increasing the dataset size and enabling the model to learn more complex user behaviors. In summary, diffusion models enhance cross-domain recommendation by leveraging their ability to handle uncertainty and generate comprehensive user preferences, where effective cross-domain knowledge transfer is critical. Furthermore, the alignment of learned representations with the target domain is intuitively important and will be explored in future work.

E. In-depth Analysis

1) Noise Schedule: In this section, we investigate the impact of different schedules to the final results. Different types of noise schedules control how to add noise iteratively to the embeddings in the forward process, therefore influencing the denoising process. Following [46], we plot the value of $\bar{\alpha}_t$ under different schedules as shown in Fig. 5, We consider three kinds of noise schedules:

- **The sqrt noise schedule** [33] quickly rises the noise levels at the first few diffusion steps and gradually slows down the injection of diffusion noises at the latter diffusion steps. It is defined as: $\bar{\alpha}_n = 1 - \sqrt{n/N + 0.0001}$.

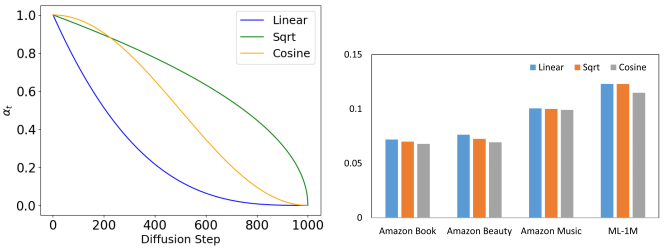


Fig. 5: Influences of different noise schedule. The left figure visualizes the $\bar{\alpha}_t$ under different noise schedules. Right figure compares the performance of Recall@10 under different noise schedule on four datasets.

TABLE V: Performance comparison with different approximator on four datasets.

Backbone Model	Amazon Book		Amazon Beauty		Amazon Music		ML-1M	
	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10
MLP	0.0696	0.0547	0.0729	0.0681	0.0876	0.0664	0.1058	0.0941
GRU	0.0698	0.0522	0.0753	0.0672	0.0892	0.0621	0.1101	0.0941
Transformer	0.072	0.0552	0.0764	0.0695	0.1006	0.0683	0.1231	0.1022

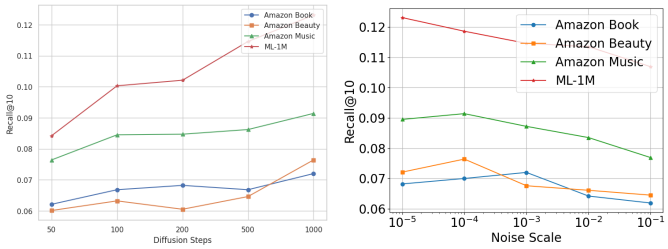
- **The cosine noise schedule** [30] smoothly increases diffusion noises using a cosine function to prevent sudden changes in the noise level. It is defined as: $\bar{\alpha}_n = \frac{g(n)}{g(0)}, g(n) = \cos(\frac{n/N+0.008}{1+0.008} \cdot \frac{\pi}{2})^2$.
- **The linear noise schedule** [46] is our default setting. It is originally proposed for image generation. It increases β_n linearly from $\beta_1 = 10^{-4}$ to $\beta_N = 0.02$.

To compare their differences, we provide a visualization of the noise schedules and and evaluate their performances on four datasets. Since $\bar{\alpha}_n$ and β_n are mutually convertible[3], we plot the value of $\bar{\alpha}_n$ with 1000 diffusion steps (default setting). A lower value of $\bar{\alpha}_n$ means higher noise level. All the results are presented in Fig. 5.

As [57] emphasizes that different schedules will impact the final outcomes to some extent but could not acquire huge margin fluctuations. We observe that the performance does not vary significantly between linear and sqrt noise schedules. The default noise schedule, linear noise schedule, generally performs well, achieving the best performance on Amazon-Book, Amazon-Music and Amazon-Beauty dataset and competitive in ML-1M dataset. The sqrt noise schedule shows similar performance, while the cosine noise schedule shows slightly lower performance. We anticipate that the superiority of linear schedule lies in facilitating a smoother and more gradual noise injection in the embeddings, which improves the stability in the training phase.

2) Effect of Backbone Model: By default, we selected *Transformer* as our approximator for estimation during diffusion process, supposing that it can properly reveal the users' current interest iteratively. We also conduct experiment with other common alternatives, specifically *MLP* and *GRU*, and present the results in Table V. Accordingly, we observe that *Transformer* consistently achieves better performance than the other network structures.

[3] $\bar{\alpha}_n$ is defined as $\bar{\alpha}_n = \prod_{i=1}^{n} 1 - \beta_i$ and we note that $\beta_n = 1 - \frac{\bar{\alpha}_n}{\bar{\alpha}_{n-1}}$



(a) (b)

Fig. 6: Influences of varying diffusion steps and noise scales.

F. Hyper-parameter Analysis

Due to limited space, we only discuss the most important hyper-parameters, diffusion steps N and noise scale s.

Diffusion Steps Generally speaking, diffusion models require a large number of diffusion steps N to achieve satisfactory performance. This is probably because a large N allows for better approximation of the Gaussian diffusion process [58]. To investigate the influences of diffusion steps, we train *CADSR* using different diffusion steps, ranging from 50 to 1000, and report the performance in Fig. 6a. We can see that model performs better with diffusion steps increasing and reaches the best performance with 1000 diffusion steps.

Noise Level Another crucial hyperparameter is maximum noise scale s, which controls the level of noise injection. From the results in Fig. 6b, we observe that the performance generally increases up to a noise scale of 10^{-4}, after which it decreases steadily. This pattern confirms our assumption that injecting noise introduces uncertainty, which helps the model learn more robust representations. However, increasing the noise scale may compromise personalization.

We attribute the pattern observed in Fig. 6 to the combined influence of sequence length and dataset sparsity. Longer sequences (i.e., ML-1M dataset) inherently contain more noisy interactions and thus require less noise injection. In contrast, sparser datasets (e.g., *Amazon Books*) necessitate a higher level of added noise.

VI. Conclusion

In this work, we propose a Context-Aware Diffusion-based Sequential Recommendation (*CADSR*) model, introducing a novel approach to tackle challenges in sequential recommendation. *CADSR* leverages contextual information during the diffusion process to generate more accurate user preferences, and contrastive learning with diffusion-based augmented views enhances the model's robustness. Extensive experiments conducted on four datasets demonstrate that *CADSR* consistently outperforms 11 state-of-the-art baselines. These results validate the effectiveness of our approach in improving recommendation accuracy.

References

[1] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, and W. Xu, "Consert: A contrastive framework for self-supervised sentence representation transfer," *arXiv preprint arXiv:2105.11741*, 2021.

[2] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, "Prototypical contrastive learning of unsupervised representations," *arXiv preprint arXiv:2005.04966*, 2020.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.

[4] X. Chen and K. He, "Exploring simple siamese representation learning," in *CVPR*, 2021.

[5] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*, 2010.

[6] R. He and J. McAuley, "Fusing similarity models with markov chains for sparse sequential recommendation," in *ICDM*, 2016.

[7] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *WSDM*, 2018.

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

[9] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations," in *CIKM*, 2018.

[10] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *ICDM*, 2018.

[11] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *CIKM*, 2019.

[12] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *SIGIR*, 2020.

[13] B. He, X. He, Y. Zhang, R. Tang, and C. Ma, "Dynamically expandable graph convolution for streaming recommendation," in *WWW*, 2023.

[14] M. Zhang, S. Wu, X. Yu, Q. Liu, and L. Wang, "Dynamic graph neural networks for sequential recommendation," *TKDE*, vol. 35, no. 5, 2022.

[15] B. He, X. He, R. Zhang, Y. Zhang, R. Tang, and C. Ma, "Dynamic embedding size search with minimum regret for streaming recommender system," in *CIKM*, 2023.

[16] L. Xia, C. Huang, Y. Xu, and J. Pei, "Multi-behavior sequential recommendation with temporal graph transformer," *TKDE*, 2022.

[17] Y. Jiang, Y. Yang, L. Xia, and C. Huang, "Diffkg: Knowledge graph diffusion model for recommendation," *arXiv preprint arXiv:2312.16890*, 2023.

[18] 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 *SIGIR*, 2022.

[19] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *SIGIR*, 2021.

[20] L. Xia, C. Huang, C. Huang, K. Lin, T. Yu, and B. Kao, "Automated self-supervised learning for recommendation," in *WWW*, 2023.

[21] Y. Yin, Q. Wang, S. Huang, H. Xiong, and X. Zhang, "Autogcl: Automated graph contrastive learning via learnable view generators," in *AAAI*, 2022.

[22] Y. Yang, C. Huang, L. Xia, C. Huang, D. Luo, and K. Lin, "Debiased contrastive learning for sequential recommendation," in *WWW*, 2023.

[23] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, "S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *CIKM*, 2020.

[24] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, J. Zhang, B. Ding, and B. Cui, "Contrastive learning for sequential recommendation," in *ICDE*, 2022.

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

[26] R. Qiu, Z. Huang, H. Yin, and Z. Wang, "Contrastive learning for representation degeneration problem in sequential recommendation," in *WSDM*, 2022.

[27] Y. Chen, Z. Liu, J. Li, J. McAuley, and C. Xiong, "Intent contrastive learning for sequential recommendation," in *WWW*, 2022.

[28] P. Zhou, J. Gao, Y. Xie, Q. Ye, Y. Hua, J. Kim, S. Wang, and S. Kim, "Equivariant contrastive learning for sequential recommendation," in *Recsys*, 2023.

[29] X. Qin, H. Yuan, P. Zhao, J. Fang, F. Zhuang, G. Liu, Y. Liu, and V. Sheng, "Meta-optimized contrastive learning for sequential recommendation," in *SIGIR*, 2023.

[30] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, 2020.

[31] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *ICLR*, 2020.

[32] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *NeurIPS*, 2021.

[33] X. Li, J. Thickstun, I. Gulrajani, P. S. Liang, and T. B. Hashimoto, "Diffusion-lm improves controllable text generation," *NeurIPS*, 2022.

[34] S. Gong, M. Li, J. Feng, Z. Wu, and L. Kong, "Diffuseq: Sequence to sequence text generation with diffusion models," in *ICLR*, 2022.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *NeurIPS*, 2014.

[36] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[37] Z. Yang, J. Wu, Z. Wang, X. Wang, Y. Yuan, and X. He, "Generate what you prefer: Reshaping sequential recommendation via guided diffusion," *arXiv preprint arXiv:2310.20453*, 2023.

[38] Y. Wang, Z. Liu, L. Yang, and P. S. Yu, "Conditional denoising diffusion for sequential recommendation," *arXiv preprint arXiv:2304.11433*, 2023.

[39] W. Wang, Y. Xu, F. Feng, X. Lin, X. He, and T.-S. Chua, "Diffusion recommender model," *SIGIR*, 2023.

[40] Z. Li, A. Sun, and C. Li, "Diffurec: A diffusion model for sequential recommendation," *arXiv preprint arXiv:2304.00686*, 2023.

[41] H. Du, H. Yuan, Z. Huang, P. Zhao, and X. Zhou, "Sequential recommendation with diffusion models," *arXiv preprint arXiv:2304.04541*, 2023.

[42] Q. Liu, F. Yan, X. Zhao, Z. Du, H. Guo, R. Tang, and F. Tian, "Diffusion augmentation for sequential recommendation," in *CIKM*, 2023.

[43] Z. Wu, X. Wang, H. Chen, K. Li, Y. Han, L. Sun, and W. Zhu, "Diff4rec: Sequential recommendation with curriculum-scheduled diffusion augmentation," in *MM*, 2023.

[44] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *ICML*, 2015.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017.

[46] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *ICML*, 2021.

[47] E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré, and M. Welling, "Argmax flows and multinomial diffusion: Learning categorical distributions," *NeurIPS*, 2021.

[48] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention*, 2015.

[49] W. Fu, Z. Peng, S. Wang, Y. Xu, and J. Li, "Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems," in *AAAI*, 2019.

[50] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[51] N. Sachdeva, G. Manco, E. Ritacco, and V. Pudi, "Sequential variational autoencoders for collaborative filtering," in *WSDM*, 2019.

[52] Z. Xie, C. Liu, Y. Zhang, H. Lu, D. Wang, and Y. Ding, "Adversarial and contrastive variational autoencoder for sequential recommendation," in *WWW*, 2021.

[53] J. Walker, T. Zhong, F. Zhang, Q. Gao, and F. Zhou, "Recommendation via collaborative diffusion generative model," in *KSEM*, 2022.

[54] Y. Xuan, "Diffusion cross-domain recommendation," *arXiv preprint arXiv:2402.02182*, 2024.

[55] H. Ma, R. Xie, L. Meng, X. Chen, X. Zhang, L. Lin, and J. Zhou, "Triple sequence learning for cross-domain recommendation," *TOIS*, vol. 42, no. 4, pp. 1–29, 2024.

[56] X. Zheng, J. Su, W. Liu, and C. Chen, "Ddghm: Dual dynamic graph with hybrid metric training for cross-domain sequential recommendation," in *MM*, 2022.

[57] Z. He, T. Sun, K. Wang, X. Huang, and X. Qiu, "Diffusionbert: Improving generative masked language models with diffusion models," *arXiv preprint arXiv:2211.15029*, 2022.

[58] S. De Groot and P. Mazur, *Non-Equilibrium Thermodynamics*, ser. Dover Books on Physics. Dover Publications, 2013.