

Unleashing the Potential of Two-Tower Models: Diffusion-Based Cross-Interaction for Large-Scale Matching

Yihan Wang

wangyihan05@kuaishou.com
Kuaishou Technology
Beijing, China

Qi Song

songqi@kuaishou.com
Kuaishou Technology
Beijing, China

Fei Xiong

xiong_info@163.com
Unaffiliated
Beijing, China

Kaiqiao Zhan

zhankaiqiao@kuaishou.com
Kuaishou Technology
Beijing, China

Zhexin Han

hanzhexin03@kuaishou.com
Kuaishou Technology
Beijing, China

Ben Wang*

wangben@kuaishou.com
Kuaishou Technology
Beijing, China

Abstract

Two-tower models are widely adopted in the industrial-scale matching stage across a broad range of application domains, such as content recommendations, advertisement systems, and search engines. This model efficiently handles large-scale candidate item screening by separating user and item representations. However, the decoupling network also leads to a neglect of potential information interaction between the user and item representations. Current state-of-the-art (SOTA) approaches include adding a shallow fully connected layer(i.e., COLD), which is limited by performance and can only be used in the ranking stage. For performance considerations, another approach attempts to capture historical positive interaction information from the other tower by regarding them as the input features(i.e., DAT). Later research showed that the gains achieved by this method are still limited because of lacking the guidance on the next user intent. To address the aforementioned challenges, we propose a "cross-interaction decoupling architecture" within our matching paradigm. This user-tower architecture leverages a diffusion module to reconstruct the next positive intention representation and employs a mixed-attention module to facilitate comprehensive cross-interaction. During the next positive intention generation, we further enhance the accuracy of its reconstruction by explicitly extracting the temporal drift within user behavior sequences. Experiments on two real-world datasets and one industrial dataset demonstrate that our method outperforms the SOTA two-tower models significantly, and our diffusion approach outperforms other generative models in reconstructing item representations.

*Corresponding Author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WWW '25, Sydney, NSW, Australia

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1274-6/25/04
<https://doi.org/10.1145/3696410.3714829>

CCS Concepts

- Information systems → Recommender systems; Information retrieval.

Keywords

Candidate Matching, Diffusion Models, Embedding-based Retrieval

ACM Reference Format:

Yihan Wang, Fei Xiong, Zhexin Han, Qi Song, Kaiqiao Zhan, and Ben Wang. 2025. Unleashing the Potential of Two-Tower Models: Diffusion-Based Cross-Interaction for Large-Scale Matching. In *Proceedings of the ACM Web Conference 2025 (WWW '25), April 28-May 2, 2025, Sydney, NSW, Australia*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3696410.3714829>

1 Introduction

Recommender systems aim to enhance user experience and business value by suggesting items of interest and driving user engagement and satisfaction. In the industry scenario, a two-stage recommender system, as shown in Figure 1(a), is extensively used for providing users with personalized content with strict latency. The first stage is called the matching stage, which narrows down the potential set of candidates from a large corpus. The second stage, known as the ranking stage [1, 11], selects the final results that the user might be interested in.

The matching stage is a critical phase of recommender systems where filters out the irrelevant candidates from billions of corpus quickly. Due to the high accuracy and low latency requirements of the matching models, two-tower models [13, 23, 33, 35] become a primary paradigm for candidate matching and support for efficient top-k retrieval [26]. The Two-tower model consists of two separate towers, one tower processes all the information about the query (user, context), while the other tower processes information about the candidates. The outputs of two towers are low-dimensional embeddings, which are then multiplied for scoring candidate items.

Since the two-tower models trained independently, they cannot leverage cross-features or interactions between user and item features until the very end, which is referred to as "Late Interaction" [17]. Recent research on fetching the interactive signals can primarily be categorized into two approaches. One method transforms the two-tower architecture into the single-tower structure by adding a shallow fully connected layer (i.e., COLD [32] and FSCD [22]), but the efficiency is still constrained and can only be used in the ranking phase. The other method attempts to augment

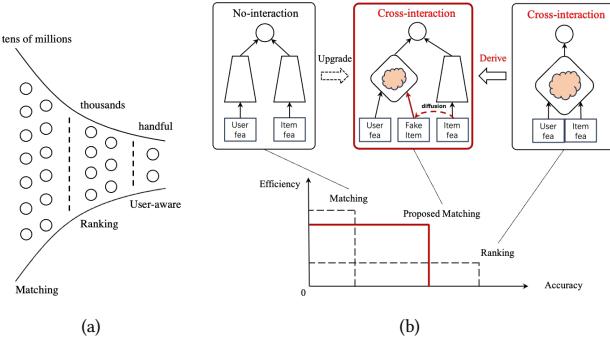


Figure 1: Real-world two-stage recommender system. (a) The two-stage architecture involves matching, which scores a large number of items, and ranking, which further refines the scoring for a smaller subset. **(b)** Intuitive view for accuracy and efficiency of matching and ranking method, where the proposed matching method is derived from ranking and optimized to a cross-interaction architecture.

the embedding input of each tower with a vector that captures historical positive interaction information from the other tower(i.e., DAT [35]), recent research shows that the gains are still limited [18] because of lacking the guidance on the next user positive intent. Current SOTA approaches are difficult to balance model effectiveness and inference efficiency. Figure 1(b) describes the aforementioned models from the perspective of inference efficiency and prediction accuracy.

To tackle the trade-off between efficiency and accuracy, we propose a generative cross-interaction decoupling architecture of the matching paradigm, named Unleashing the Potential of Two-Tower Models: Diffusion-Based Cross-Interaction for Large-Scale Matching (**T2Diff**). T2Diff has exceeded the limits of the two-tower architecture by extracting user-item cross features with the guidance of target item restored by the diffusion module. Considering the performance issues caused by the large-scale corpus in the matching phase, instead of the single-tower structure, we employ a generative method that reconstructs the user's positive interactions contained in the item tower through a diffusion model in the user tower. To model the interactions between user and item features sufficiently, a mixed-attention module is introduced to enhance the user's positive interaction from the other tower. This mixed-attention module extracts user representation more accurately by interacting with the item information and the user's historical behavior sequence. The main contributions of this paper are as follows:

- We propose a new matching paradigm named T2Diff which is a generative cross-interaction decoupling architecture that emphasizes information interactions and unleashes the potential of two-tower model with high accuracy and low latency.
- T2Diff introduces two key innovations: 1) a generative module to reconstruct user's next positive intention by applying diffusion-based model, and 2) a mixed-attention mechanism [29, 38] to address the challenge of the "Late Interaction" by facilitating more complex and enriched user-item feature interactions at the foundational level of the model architecture.

- T2Diff not only outperforms the baselines on both two real-world datasets and one industrial dataset, but also demonstrates great inferences efficiency.

2 Related Works

Embedding-based Retrieval (EBR): A technique that uses embeddings to represent users and items, converting the retrieval problem into a nearest neighbor (NN) search problem in the embedding space [5, 15]. EBR models are widely applied in the matching stage [12], which selects a list of candidates from a large corpus based on the user's historical behavior. Typically, EBR models consist of two parallel deep neural networks for learning the encoding of the users and items, which are trained separately and also known as two-tower model [13, 33, 34]. This architecture has the advantages of high throughput and low latency, while the ability to capture the interactive signals between user and item representations is limited. To mitigate the problem, DAT [35] introduces a adaptive-mimic mechanism which customizes an augmented vector for each user and item, compensating for the lack of interactive signals. However, later research [18] shows that the gain of only introducing an augmented vector as the input features is limited. Therefore, T2Diff leverages the mixed-attention module to extract high-order feature interactions and user historical behaviors with the target representations generated by diffusion module.

Session-based Recommendation and Interests Drift. Feng *et al.* [3] have observed that user behaviors within each session exhibit a high degree of homogeneity, yet they tend to drift across different sessions. Zhou *et al.* [37] have discovered that the accuracy of predicting the Click-Through Rate (CTR) is significantly enhanced when the predictions are aligned with the trend of interests drift.

The Application of Generative Model in Sequential Recommendation. Although traditional sequential models, such as SAS-Rec [16], Mamba4Rec [20] have demonstrated satisfactory performance, the emergence of generative models has revealed a new and promising direction. VAEs [2, 8, 31] have been utilized to learn a latent space representation of items and users, from which new sequences can be generated. However, these kind of generative models might oversimplify the data distribution, leading to a loss of information and potentially less accurate representations. Diffusion models have made remarkable success in many fields, including recommender systems [10, 19, 30, 39], natural language processing [8, 14, 21], and computer vision [9, 24, 25]. DiffuRec [19] made the first attempt to apply diffusion modeling to SR and adopted a single embedding to fetch a user's multiple interests due to its ability of distribution generation and diversity representation. While VAEs and diffusion models applied in computer vision [8, 14, 21] typically rely on a Kullback-Leibler divergence loss[KL-loss] to measure the difference between the learned latent distribution and a prior distribution (often a Gaussian), DiffuRec opts for a cross-entropy loss during the process of reconstructing the target item. In order to restore item representation stably and accurately, T2Diff adopts a diffusion module with Kullback-Leibler divergence loss[KL-loss]. This module can accurately reconstruct the target item with low latency, providing a solid foundation for capturing cross-information within the two-tower structure.

3 Preliminary

In this section, we briefly introduce the diffusion models as preliminary knowledge.

3.1 Diffusion Models

Diffusion models can be divided into two stages, diffusion process and reverse process. Fundamentally, Diffusion Models work by destroying training data through the successive addition of Gaussian noise in diffusion process, and then learning to recover the data by reversing this noising process in reverse process.

In the diffusion process, the diffusion models add the Gaussian noise successively to the original representations x_0 via a Markov Chain (i.e., $x_0 \rightarrow x_1 \rightarrow \dots \rightarrow x_T$) as follows:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t I) \quad (1)$$

where $\mathcal{N}(x; \mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2 . β_t represents the amplitude of added Gaussian noise, with higher values of β_t indicating a higher level of introduced noise. I is the identity matrix.

We can go in a closed form from the input data x_0 to x_T in a tractable way and the posterior probability can be defined as:

$$q(x_{1:T}|x_0) = \prod_{t=1}^T q(x_t|x_{t-1}) \quad (2)$$

According to DDPM [9], with the help of reparameterization trick, we can find that the posterior $q(x_r|x_0)$ obey a Gaussian distribution. Let $\alpha_r = 1 - \beta_r$ and $\bar{\alpha}_r = \prod_{i=1}^r \alpha_i$, then the Equation 2 can be rewritten as

$$q(x_r|x_0) = \mathcal{N}(x_r; \sqrt{\bar{\alpha}_r}x_0, (1 - \alpha_r)I) \quad (3)$$

In the reverse process, we gradually denoise from the standard Gaussian representation x_T and approximate the real representation x_0 (i.e. $x_T \rightarrow x_{T-1} \rightarrow \dots \rightarrow x_0$) in an iterative way. Specially, given the current restored representation x_t and the original representation x_0 , the next representation x_{t-1} can be calculated as follows:

$$p(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I) \quad (4)$$

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}x_t \quad (5)$$

$$\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}\beta_t \quad (6)$$

However, the original representation x_0 is always unknown in the reverse process, thus requiring a deep neural network to estimate x_0 . The reverse process is optimized by minimizing the following variational lower bound (VLB).

$$\begin{aligned} L_{VLB} &= E_{q(x_1|x_0)} [\log p_\theta(x_0|x_1)] - D_{KL}(q(x_T|x_0)||p_\theta(x_T)) \\ &\quad - \sum_{t=2}^T E_{q(x_t|x_0)} [D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))] \\ &= L_0 - L_T - \sum_{t=2}^T L_{t-1} \end{aligned} \quad (7)$$

where $p_\theta(x_t) = \mathcal{N}(x_t; 0, I)$ and $D_{KL}(\cdot)$ is the KL divergence.

Each KL divergence term in L_{VLB} , with the exception of L_0 , involves the comparison of two Gaussian distributions. As such, these

terms can be analytically computed in closed form. The term L_T is constant during the training process, making it inconsequential for optimization. This is because the distribution q lacks trainable parameters and x_T is simply Gaussian noise. For modeling L_0 , Ho *et al.* utilize a separate discrete decoder derived from N . Following [9], L_{VLB} can be simplified as a Gaussian noise learning process, which can be denoted as

$$L_{simple} = E_{t \in [1, T], x_0, \epsilon_t} [||\epsilon_t - \epsilon_\theta(x_t, t)||^2] \quad (8)$$

where $\epsilon \sim \mathcal{N}(0, I)$ is sampled from a standard Gaussian distribution, and $\epsilon_\theta(\cdot)$ represents an **Estimator** that can be learned by a deep neural network.

4 Method

In this section, we first introduce the notation and background related to T2Diff. We then detail the framework of our model, which consists of a diffusion module and a mixed-attention module, as shown in the Figure 2(a).

4.1 Notations and Problem Formulation

Suppose that we have a set of users \mathcal{U} and a set of items \mathcal{M} . We collect the behavior sequence of each user and denote it as $X_{sequence} \in \mathcal{M}$. We tell each behavior of user $u \in \mathcal{U}$ as x_j^u , where j represents the j -th item of the behavior sequence. For each user, suppose that we have n historical behaviors, then index $j \in \{1, 2, \dots, n+1\}$ and $X_{sequence} = [x_1, x_2, \dots, x_n]$. Building upon the concept presented in [3], we aim to achieve a more refined modeling of user behavior sequences by dividing them into two distinct parts based on the time intervals that separate each action. Specifically, we divide the ordered behavior sequence into the current session with recent k interacted behaviors denoted by $X_{session} = [x_{n-k+1}, \dots, x_n]$ and historical behaviors denoted as $X_{history} = [x_1, x_2, \dots, x_{n-k}]$. We believe that the user's behaviors within the most recent session are temporally continuous, and reflect the user's nearest intentions. Finally, the most important point, we unleash the two-tower model's potential by introducing the predicted next positive behavior \hat{x}_{n+1} from the true one x_{n+1} .

Embedding-based retrieval (EBR) methods encode user and item features into embeddings by two independent deep neural networks. The relevance of item \mathcal{M} to user \mathcal{U} is determined based on the distance (most commonly, inner product) between user embedding e_u and item embedding e_i .

Our proposed T2Diff has two main parts: 1) A diffusion module designed to identify drift in interests between adjacent behaviors during the training phase, and to reintroduce next behavior during the inference stage. 2) A session-based mixed-attention module that extracts current interests from the latest session and the predicted next behavior by applying a self-attention module and fetching historical interests with a target-attention mechanism. The combination of these two components enables a full cross-interaction between the user's behavior sequence and the next behavior.

4.2 Diffusion Module

Referring to the bottom of Figure 2(a), the input of the diffusion module is the complete user's behaviors $X_{sequence}$ and the next positive behavior x_{n+1} , which fed into the diffusion process and

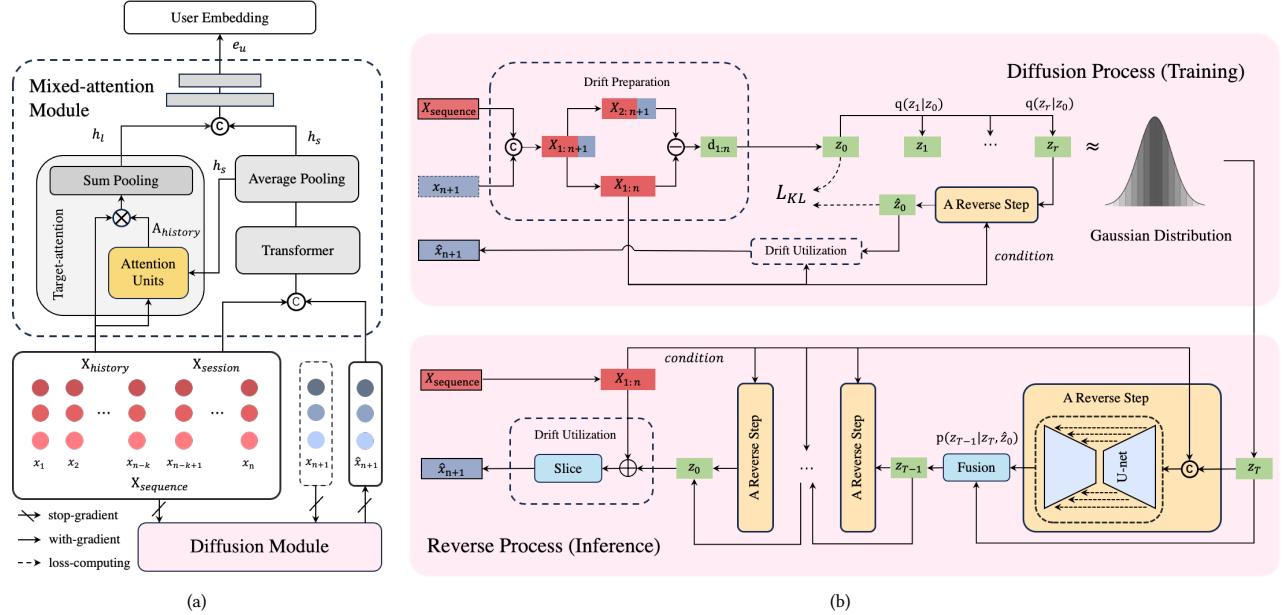


Figure 2: (a) The main architecture of T2Diff includes a mixed-attention module and a diffusion module. The forward slash character used to connect the diffusion module and embedding layer indicates stop-gradient. **(b)**: The details of the diffusion module, which adopts different processes for training and inference. For each reverse step in the diffusion module, we utilize a Unet approximator.

transformed into the standard Gaussian distribution. The output is the predicted next positive behavior \hat{x}_{n+1} restored from a sample of standard Gaussian distribution, which is expected to be the same as x_{n+1} in the reverse process.

$$\hat{x}_{n+1} = \text{Diffusion}(X_{session}, x_{n+1}) \quad (9)$$

Regarding the drift preparation step illustrated on the upper left of Figure 2(b), assuming there are n user historical behaviors and 1 next behavior, concatenating these behaviors in time series we can get $X_{1:n+1}$. One necessary step is to obtain the drift between adjacent behaviors. We employ sliding windows to derive $X_{1:n}$ and $X_{2:n+1}$ separately, from those we calculate the element-wise subtraction of $X_{2:n+1}$ from $X_{1:n}$, resulting in $d_{1:n}$ or z_0 , to which we add noise.

$$X_{1:n+1} = \text{concat}([X_{sequence}, x_{n+1}]) \quad (10)$$

$$z_0 = d_{1:n} = X_{2:n+1} - X_{1:n} \quad (11)$$

We believe that diffusion and reverse from drift between adjacent behaviors is much easier than from the original user behavior sequence $X_{1:n}$. The effectiveness of this approach will be demonstrated by experiments in Section 5.1.4.

4.2.1 Diffusion Process. During the training process, we randomly add r steps of Gaussian noise to a batch of data, which can be achieved by 1 step from $q(\cdot)$, according to Equation 3. The step-index r is randomly selected from Uniform distribution $[1, T]$, where T is the upper limit of the diffusion step. We have devised an exponential noise schedule β to introduce noise incrementally, showcasing its novelty in Section 5.1.5. The procedure of a single

diffusion step can be represented by Equation 12-14:

$$r \sim \text{Uniform}(\{1, \dots, T\}) \quad (12)$$

$$\beta_r = a \cdot e^{br} \quad (13)$$

$$z_r = \sqrt{\bar{\alpha}_r} z_0 + \sqrt{1 - \bar{\alpha}_r} \epsilon \quad (14)$$

where a, b, T are hyper-parameters, $\epsilon \sim \mathcal{N}(0, I)$. Next, we choose U-Net as the backbone of our approximator to recover its unbiased estimation \hat{z}_0 . The traditional U-Net architecture comprises an encoder, a decoder, and skip connections, facilitating the generation of an output that keeps dimensional congruence with the input. One of the salient advantages of utilizing U-Net lies in its convolutional kernels, which can capture user interest drift over time. This feature significantly augments the model's capacity for discerning intricate user interest patterns within the original input sequence and effectively reconstructing them from a noised input, denoted as z_r .

Notably, conditional diffusion models incorporate additional information as input, such as the class label c . In our context, the user's original behavior sequence $X_{1:n}$ is rich in information regarding interest drift and is accessible during both the training and inference phases. Consequently, $X_{1:n}$ is employed as a conditional factor to direct the reverse direction of the approximator.

$$\hat{z}_0 = \text{U-Net}(\text{concat}([z_r, X_{1:n}])) \quad (15)$$

4.2.2 Reverse Process. Following the application of Diffusion Models in Computer Vision [9, 24, 25], even if we start reversing next behavior from a sample of standard Gaussian noise $z_T \sim$

$N(0, I)$, it is possible to recover the drift between the last behavior x_n and the next behavior x_{n+1} with the guidance of original user behavior $X_{1:n}$. In each reverse step, as shown in Equation 15 and Equation 16, we obtain \hat{z}_0 from the U-Net approximator with learnable parameters, denoted as f_θ . Then we combine \hat{z}_0 with z_t to get z_{t-1} via $p(\cdot)$. After applying the reparameterization trick, a single reverse step presents as follows:

$$\hat{z}_0 = f_\theta(z_0 | z_t, X_{1:n}) \quad (16)$$

$$z_{t-1} = \text{Fusion}(z_t, \hat{z}_0)$$

$$\begin{aligned} &= \tilde{\mu}_t(z_t, \hat{z}_0) + \tilde{\beta}_t \epsilon' \\ &\sim p(z_{t-1} | z_t, \hat{z}_0) \end{aligned} \quad (17)$$

where $\epsilon' \sim N(0, I)$. After T reverse steps, the output z_0 is still an intermediate one. Therefore, in the drift utilization step, we add it to $X_{1:n}$, and obtain the last behavior as predicted next behavior \hat{x}_{n+1} , as shown on the lower left of Figure 2(b).

$$\hat{x}_{n+1} = \text{Slice}(z_0 + X_{1:n}) \quad (18)$$

where $\text{Slice}(x) = x[-1]$. The reverse process of the diffusion module is illustrated in Algorithm 2.

Algorithm 1 Diffusion Process(Training)

- 1: **Inputs:**
 - 2: User Historical Sequence: X_{sequence} or $X_{1:n}$
 - 3: Next Behavior: x_{n+1}
 - 4: $X_{1:n+1} = \text{concat}([X_{\text{sequence}}, x_{n+1}])$
 - 5: $z_0 = X_{2:n+1} - X_{1:n}$ // drift preparation
 - 6: $r \sim \text{Uniform}(\{1, \dots, T\})$
 - 7: $z_r \sim q(z_r | z_0)$
 - 8: $\hat{z}_0 = \text{U-Net}(\text{concat}([z_r, X_{1:n}]))$
 - 9: parameter update : $L_{KL}(\hat{z}_0, z_0)$
 - 10: $\hat{x}_{n+1} = \text{Slice}(\hat{z}_0 + X_{1:n})$ // drift utilization
 - 11: **return** \hat{x}_{n+1}
-

Algorithm 2 Reverse Process (Inference)

- 1: **Inputs:**
 - 2: User Historical Sequence: X_{sequence} or $X_{1:n}$
 - 3: Gaussian Sampling: $z_T \sim N(0, I)$
 - 4: **for** $t = T, \dots, 1$ **do**
 - 5: $\hat{z}_0 = \text{U-Net}(\text{concat}([z_t, X_{1:n}]))$
 - 6: $z_{t-1} = \text{Fusion}(z_t, \hat{z}_0)$
 - 7: **end for**
 - 8: $\hat{x}_{n+1} = \text{Slice}(z_0 + X_{1:n})$ // drift utilization
 - 9: **return** \hat{x}_{n+1}
-

4.3 Mixed-attention Module

To overcome the issue of "Late Interaction" in the two-tower model, we propose a mixed-attention mechanism that facilitates intricate feature interactions by engaging multi-layer user representations with the reconstructed user's recent positive item representation obtained by the diffusion module in Section 4.2. In the realm of short-video recommendation, user consumption behaviors demonstrate temporal continuity. We consider that the last session contains

the user's recent positive intention, and to enhance the cross-interactions between historical sequences and the next positive item representation, we concatenate X_{session} and \hat{x}_{n+1} along the temporal dimension. In our approach, we deploy the encoder component of the transformer architecture [29] and average pooling to generate current interests embedding h_s for "Early Interaction".

$$h_s = \text{avg}(\text{Transformer}(\text{concat}([X_{\text{session}}, \hat{x}_{n+1}]))) \quad (19)$$

To further exploit the benefit of cross-interaction, following [38], we use h_s as guidance to extract similar information from the user's historical behaviors X_{history} . In the activation units, the historical behavior embeddings X_{history} , the current interests embedding h_s , and their outer product are provided as the inputs for generating attention weights A_{history} , as illustrated in Figure 3. Finally, h_t and h_s will collectively determine the user embedding e_u .

$$a_j = \frac{\text{FFN}(\text{concat}([x_j, x_j - h_s, x_j * h_s, h_s]))}{\sum_{i=1}^{n-k} \text{FFN}(\text{concat}([x_i, x_i - h_s, x_i * h_s, h_s]))} \quad (20)$$

$$h_l = f(h_s, [x_1, x_2, \dots, x_{n-k}]) = \sum_{j=1}^{n-k} a_j x_j \quad (21)$$

$$e_u = \text{FFN}(\text{concat}([h_l, h_s])) \quad (22)$$

where a_j is the j -th element of A_{history} . Considering the temporal dependencies within a session and the correlations of behavioral patterns across sessions, we introduce the time lag between the target behavior and historical behaviors as a critical feature.

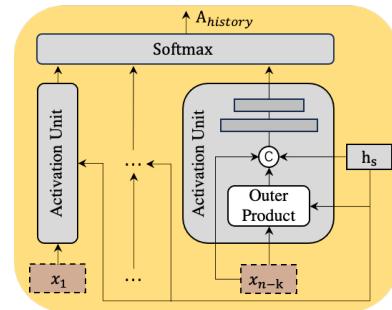


Figure 3: The detailed implementation of activation units used for target-attention.

4.4 Model Optimization

In each diffusion step, we derive \hat{z}_0 directly from z_r , where \hat{z}_0 and z_0 both represent the mean of the distribution by reparameterization. Therefore, the simplified version of L_{VLB} from Equation 7 can be rewritten as follows, denoted as L_{KL} ,

$$L_{KL} = E_{r \in [1, T], x_0, \mu_r} [||\mu_r - \mu_\theta(z_r, r)||^2] \quad (23)$$

where μ_r and z_r represent the noise added during the diffusion process at step r and the results after adding noise, respectively, μ_θ denotes the estimator with parameters θ .

With the help of L_{KL} , we can reduce the difference between z_0 and \hat{z}_0 and renew the parameters in the approximator by gradient descent. The diffusion process of the diffusion module is illustrated in Algorithm 1.

Following the general principles of loss function in the recommender systems, a softmax loss L_{TOWER} is utilized to bring the user embedding e_u close to the target item embedding e_i while far away from the remaining irrelevant item embeddings $e_{m \in M}$, which is denoted as

$$L_{TOWER} = -\log \frac{\exp(e_u \cdot e_i)}{\sum_{m \in M} \exp(e_u \cdot e_m)} \quad (24)$$

Enabled by the loss function L_{TOWER} , the sparse embedding table undergoes thorough training, thereby establishing a robust foundation for the diffusion process training. The total loss can be denoted as

$$L_{TOTAL} = L_{TOWER} + \lambda L_{KL}. \quad (25)$$

where λ is a hyper-parameter, usually set to 1 or 10. Because the optimization direction of the approximator within the diffusion module is inconsistent with that of traditional recommender systems, which will easily lead to a situation where gradients counteract each other, so we employ a stop-gradient mechanism to isolate the gradient updates of the diffusion module, effectively enhancing the optimization efficiency of both the approximator and the tower parameters, as shown on the bottom of Figure 2(a).

5 Experiments

In this section, we conduct experiments on two benchmark datasets frequently utilized in the research community and a large-scale industrial dataset consisting of one million users from an online short-video platform. In detail, we try to answer the following questions:

- How effective is the proposed method compared to other SOTA two-tower sequential recommendation models? **Q1**
- What are the distinct contributions of the individual modules within our proposed model, including the diffusion module for the reconstruction of the next behavior and the mixed-attention module that facilitate efficient cross-interaction? **Q2**
- How should the hyper-parameters of the diffusion module be selected to strike a balance between accuracy and efficiency, particularly with respect to the noise generation schedule β and the upper limit of diffusion step T ? **Q3**

5.1 Offline Evaluation

5.1.1 Datasets. We conduct extensive offline experiments over two public datasets, KuaiRand and ML-1M. The number of users, items and interactions are displayed in the Table 1.

Table 1: Statistics of the datasets

Dataset	#Users	#Items	#Interactions
KuaiRand	25,828	108,025	6,492,153
ML-1M	6,040	3,648	643,979

KuaiRand [4] is a publicly available dataset collected from the logs of the recommender system in Kuaishou. Following [36], we extract user-items interactions from main recommendation scenario where the "tab" field equals one and treat clicked items as relevant to user.

ML-1M [6] consists of over one million anonymous ratings of about 3648 movies made by 6040 MovieLens users, which is widely used in other sequential-aware methods, such as Caser [28] and SASRec [16]. All the movies watched by a user are considered relevant.

For both datasets, we focus on positive samples for training purposes, specifically clicked items in KuaiRand and watched items in ML-1M. Additionally, we exclude users with interaction frequencies below five to ensure a more robust training dataset.

5.1.2 Evaluation Metrics and Baselines. To emulate real-world scenarios, for each method, a candidate set of \mathcal{K} most relevant items is generated. Then, we utilize $Recall@K$ and Mean Reciprocal Rank ($MRR@K$) to evaluate the offline effectiveness, which are widely applied in recommender systems. K is set to 2 and 20 for ML-1M dataset and 10 and 100 for KuaiRand dataset.

We compare it with the following SOTA baselines:

- **SASRec** [16] introduces self-attention mechanism to quickly generate user embedding.
- **Caser** [28] uses convolutional layers and max-pooling operation to capture local and global patterns in the sequence.
- **GRU4Rec** [7] employs multiple GRU units to effectively capture the complex relationships in a user historical behavior sequence.
- **Bert4Rec** [27] introduces the Cloze objective and bidirectional transformer structure to accomplish target prediction.
- **ContrastVAE** [31] employs a two-branched VAE framework guided by ContrastELBO to address the challenge for sparsity of user-item interactions.
- **STOSA** [2] devises a novel Wasserstein Self-Attention module to characterize item-item position-wise similarity in the sequence.
- **DiffuRec** [19] employs diffusion models to accomplish item representation construction and uncertainty injection for sequence recommendation.
- **DAT+** [35] integrates a Adaptive-Mimic Mechanism (AMM) to mitigate the lack of information interaction. In the experiment setting, we employ AMM base on regular two-tower architecture.
- **Mamba4Rec** [20] firstly leverages the power of selective SSMS for efficient sequential recommendation to address the dilemma of recommendation performance and inference efficiency.

In summary, SASRec, Caser, GRU4Rec, Bert4Rec and Mamba4Rec are representative examples of traditional two-tower models. DAT is distinguished by its item-augmented approach within the same architectural framework. Furthermore, ContrastVAE, STOSA, and DiffuRec are recognized as generative models that innovate upon the two-tower paradigm.

5.1.3 Comparisons with SOTA. (Q1). Due to the sheer volume of data in the KuaiRand dataset, it is challenging for the model to rank the target item within a smaller candidate set, which consequently results in relatively poorer overall performance when compared to the ML-1M dataset. To simplify the presentation, the results on the KuaiRand dataset will be multiplied by 10. The detailed quantitative comparison results is shown in Table 2.

Among all traditional two-tower recommendation baselines, DAT+ outperforms others on some metrics, suggesting that it is beneficial to introduce additional interactions for two-tower model. However, DAT ignores the temporal relationship within the user

Table 2: Performance comparison on ML-1M and KuaiRand with other SOTA methods. The best results of all methods are highlighted in bold font and the best results of the baselines are underlined. 'Improvement' is the relative improvement against the best baseline performance. All the performance gains are statistically significant at $p < 0.05$. 'Params' denotes parameters. 'Infer Time' is the inference time consumption per sample, test on 5 Tesla T4 GPUs.

Dataset	ML_1M								Params (MB)	Infer Time (ms)	
	Recall		MRR		Recall		MRR				
Method	2	20	2	20	10	100	10	100			
SASRec	0.05779	0.15564	0.04515	0.05717	0.03796	0.21192	0.01631	0.01991	0.126	0.56	
Caser	0.05808	0.15709	0.04259	0.05769	0.04076	0.19544	0.01653	0.01981	0.182	0.64	
GRU4Rec	0.06070	0.16564	0.04503	0.05753	0.03909	0.21304	0.01492	0.01880	0.119	0.60	
Bert4Rec	0.06272	0.20559	0.04589	0.06532	0.04041	0.19902	0.01465	0.01898	0.126	0.57	
DAT+	0.05988	<u>0.25785</u>	0.04506	<u>0.07140</u>	0.04334	0.20760	0.01320	0.01624	0.127	0.54	
Mamba4Rec	0.06919	0.20581	0.05262	0.06888	0.05219	0.22800	0.02031	0.02378	0.131	1.11	
ContrastVAE	0.06023	0.15203	0.04436	0.05764	0.04195	0.21155	0.01605	0.02024	0.126	0.63	
STOSA	0.05905	0.16008	0.04454	0.05705	0.03745	0.21365	0.01678	0.01991	0.244	0.56	
DiffuRec	0.06076	0.14971	0.04288	0.05469	0.04349	0.22614	0.01708	0.02028	0.146	0.65	
Mixed-attention	0.06308	0.26017	0.04826	0.07452	0.04502	0.23990	0.01735	0.02310	0.126	0.59	
W/O DP	0.07233	0.25198	0.05683	0.07996	0.05014	0.24200	0.01694	0.02142	0.187	0.68	
T2Diff(Ours)	0.07738	0.27727	0.06076	0.08730	0.05505	0.25294	0.02106	0.02475	0.187	0.68	
Improvement	+11.84%	+7.53%	+15.47%	+22.27%	+5.48%	+10.94%	+3.69%	+4.08%	---	---	

historical behaviors, which limits the extent of performance improvement. In contrast, our proposed T2Diff effectively reconstructs target item representation by accounting for the temporal drift within user sequences, thereby achieving superior performance.

ContrastVAE and STOSA both utilize the Variational AutoEncoder (VAE) framework to model user behavior sequences. However, these models rely on two distinct embedding representations to capture the mean and variance, complicating the optimization process. DiffuRec utilizes diffusion model to introduce target information, but often struggles to achieve good performance due to ignoring the temporal relationship between target items and the users' historical sequence of actions.

In comparison to the best baseline, our proposed T2Diff demonstrates a significant enhancement in recall and MRR, specifically by 11.84% and 22.27%, respectively, on the ML-1M dataset, and by 10.94% and 4.08%, respectively, on the KuaiRand dataset. It demonstrates that reconstructing target item is an effective strategy for addressing the "Late Interaction" problem, which significantly enhances the model's performance and elevates the target item to a desirable rank within the candidate set.

Additionally, we compare the computational complexity of our proposed T2Diff with other SOTA methods. As demonstrated in Table 2, T2Diff achieves superior performance without a substantial increase in parameters. Furthermore, diffusion models are often associated with longer inference times, which can limit their applicability in real-world recommendation scenarios. To address this, we also calculate the inference time of various models, as presented in Table 2. Compared to other SOTA methods, T2Diff maintains stable performance while keeping competitive time complexity.

5.1.4 Ablation experiments. (Q2). In this section, we will delineate the specific contributions of each module. Table 2 shows the ablation results on both datasets. Our model surpasses all baselines with the mere inclusion of the mixed-attention module in several metrics.

Furthermore, by incorporating diffusion module, our model achieves significant improvements on both datasets, with recall rates increasing by 22.67% on ML-1M, and by 25.90% on KuaiRand, respectively.

Furthermore, the importance of fetching user interest drift can be evaluated by removing the drift preparation (DP) step. As shown in Table 2, our experimental results demonstrate that the inclusion of the DP step significantly improved our model's performance on both datasets. These findings provide further evidence to support the notion that modeling user interest drift is vital for accurate prediction of user's preferred items.

To further substantiate the corrective impact of the diffusion module during the diffusion and subsequent reverse processes, we track the cosine similarity between z_0 and \hat{z}_0 throughout the reverse process. As depicted in Figure 4, the diffusion module demonstrated satisfactory performance in reversing z_r back to z_0 after adequate amount of iterations.

5.1.5 Verification of Hyper-parameters Selection for the Diffusion Module. (Q3). To enhance the effectiveness of the Diffusion module,

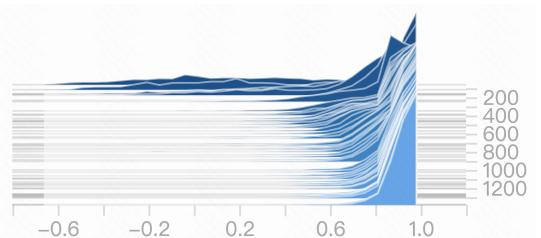


Figure 4: This figure delineates the distribution of similarities between z_0 and \hat{z}_0 across the diffusion process. The horizontal axis represents cosine similarity, while the vertical axis corresponds to the number of iterations.

we undertake an investigation of the diffusion process by replacing various noise injection methods and optimizing the number of steps. The detailed description of our experiments is provided below.

Validation of β design. We conduct a comprehensive analysis of the impact of various noise generation methods. Specifically, we carry out a comparative study among three different approaches, called linear schedule, logarithmic schedule, and exponential schedule. As shown in Table 3, our finding reveals that the exponential schedule, which is applied in our model, outperforms the popular linear schedule utilized in previous DiffuRec [19] studies. This approach can better satisfy the needs of the diffusion model for uniformly perturbing the inputs at each step, leading to improved performance and reliability of our results.

Table 3: Ablation studies of different methods to improve the noise level in diffusion process on ML-1M dataset.

β schedule	recall@2	recall@20	mrr@2	mrr@20
linear	0.07686	0.27519	0.05950	0.08677
log	0.07256	0.25151	0.05767	0.08137
exp(ours)	0.07738	0.27727	0.06076	0.08730

Validation of the maximum step in diffusion module. We conduct an ablation study to ascertain the optimal diffusion step T for our model. Experiments were conducted with T set to 10, 50, and 200. As shown in Table 4, an increased diffusion step yielded progressively better performance for our proposed T2Diff model. Notably, while a diffusion step of 200 yielded the optimal MRR, further increments in T dose not proportionally enhance performance over the $T = 50$. Furthermore, elevating the diffusion step to 200 from 50 significantly amplified the inference time per sample by 238%, which could impede the practical utility of T2Diff in industrial settings. Consequently, we have elected to set the diffusion step count at 50 for both industrial applications and all subsequent experiments detailed in this paper.

Table 4: Ablation studies of maximum steps in diffusion module on ML-1M dataset.

steps	recall@2	recall@20	mrr@2	mrr@20	Infer Time (ms)
10	0.0769	0.2724	0.0592	0.0852	0.22
50(ours)	0.0774	0.2773	0.0608	0.0873	0.68
200	0.0771	0.2715	0.0623	0.0885	2.30

5.2 Live A/B Experiment

To validate the effectiveness of the proposed matching framework named T2Diff, we conducted a week-long online A/B test (from March 27 to April 3, 2024) on a prominent large-scale short-video platform. The test involved over three million users, who were part of the experimental cohort. Within the experiment group, T2Diff is utilized as one of the potential candidate sources during the matching stage. We perform a comparison between the engagement rates of items selected from our source and those of other two-tower matching methods within the experiment. As evidenced in Table

5, the proposed approach exhibits superior performance over the other candidate sources. Furthermore, Figure 5 demonstrates the live experiment results. On the x-axis is the date, and on the y-axis is the relative difference of a metric in percentage between the experiment and control. Relative to the control, the experiment group with T2Diff improves the average App usage duration by **+0.143%** with a 95% confidence interval of **(+0.02%, +0.26%)**.

Table 5: A Comparison of Online Engagement Rates for Recommended Items between a regular two-tower matching method, DiffuRec and Our Approach: Analysis of Effective View Rate (EVR), Follow Rate (FTR), Average Played Duration (Play)

Metrics	EVR(%)	FTR(%)	Play(s)
Regular Two-tower	17.2%	0.45%	11.4s
DiffuRec	21.9%	0.60%	15.26s
Ours	24.6%	0.67%	20.96s
Improvement	+10.98%	+11.67%	+37.42%

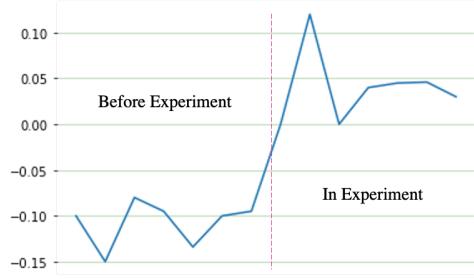


Figure 5: Live experiment results. On the x-axis is the date; on the y-axis is the relative difference in percentage between the experiment and control.

6 Conclusion

In this paper, we propose a novel matching paradigm, T2Diff, which represents a significant advancement in generative cross-interaction decoupling architectures. This paradigm unleashes the potential of two-tower model by fetching the cross information between user and item representations which surmounts the challenge of "Late Interactions". T2Diff incorporates a generative module for precise reconstruction of the user's impending positive intention and introduces a mixed-attention mechanism to capture interactive signals based on the positive intention generated by the diffusion module. Moreover, the application of the diffusion model in matching stage to restore the target information offers expanded possibilities for generative retrieval methods. Extensive offline and online experiments demonstrate that T2Diff outperforms the SOTA two-tower retrieval models significantly, while numerous ablation studies validate the accuracy of our model design.

References

- [1] Fedor Borisyuk, Krishnaram Kenthapadi, David Stein, and Bo Zhao. 2016. CaSMoS: A Framework for Learning Candidate Selection Models over Structured Queries and Documents. 441–450. <https://doi.org/10.1145/2939672.2939718>
- [2] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. 2022. Sequential recommendation via stochastic self-attention. In *Proceedings of the ACM Web Conference 2022*. 2036–2047.
- [3] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep Session Interest Network for Click-Through Rate Prediction. arXiv:1905.06482 [cs.IR] <https://arxiv.org/abs/1905.06482>
- [4] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. Kuairand: an unbiased sequential recommendation dataset with randomly exposed videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3953–3957.
- [5] Ruiqi Guo, Quan Geng, David Simcha, Felix Chern, Sanjiv Kumar, and Xiang Wu. 2019. New Loss Functions for Fast Maximum Inner Product Search. *CoRR* abs/1908.10396 (2019). arXiv:1908.10396 <http://arxiv.org/abs/1908.10396>
- [6] F. Maxwell Harper, Joseph A. Konstan, and Joseph A. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5 (2016), 19:1–19:19. <https://api.semanticscholar.org/CorpusID:16619709>
- [7] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. arXiv:1511.06939 [cs.LG]
- [8] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=Sy2fzU9gl>
- [9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 6840–6851. https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf
- [10] Yu Hou, Jin-Duk Park, and Won-Yong Shin. 2024. Collaborative Filtering Based on Diffusion Models: Unveiling the Potential of High-Order Connectivity. arXiv:2404.14240 [cs.IR] <https://arxiv.org/abs/2404.14240>
- [11] Jiri Hron, Karl Krauth, Michael I. Jordan, and Niki Kilbertus. 2020. Exploration in two-stage recommender systems. *CoRR* abs/2009.08956 (2020). arXiv:2009.08956 <https://arxiv.org/abs/2009.08956>
- [12] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-Based Retrieval in Facebook Search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Virtual Event, CA, USA) (KDD '20)*. Association for Computing Machinery, New York, NY, USA, 2553–2561. <https://doi.org/10.1145/3394486.3403305>
- [13] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management* (San Francisco, California, USA) (CIKM '13). Association for Computing Machinery, New York, NY, USA, 2333–2338. <https://doi.org/10.1145/2505515.2505665>
- [14] Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, Chenye Cui, and Yi Ren. 2022. ProDiff: Progressive Fast Diffusion Model for High-Quality Text-to-Speech. In *Proceedings of the 30th ACM International Conference on Multimedia (Lisboa, Portugal) (MM '22)*. Association for Computing Machinery, New York, NY, USA, 2595–2605. <https://doi.org/10.1145/3503161.3547855>
- [15] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *CoRR* abs/1702.08734 (2017). arXiv:1702.08734 <http://arxiv.org/abs/1702.08734>
- [16] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.
- [17] Omar Khatab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. arXiv:2004.12832 [cs.IR]
- [18] Xiangyang Li, Bo Chen, HuiFeng Guo, Jingjie Li, Chenxi Zhu, Xiang Long, Sujian Li, Yichao Wang, Wei Guo, Longxia Mao, Jinxing Liu, Zhenhua Dong, and Ruiming Tang. 2022. InfTower: the Next Generation of Two-Tower Model for Pre-Ranking System. arXiv:2210.09890 [cs.IR]
- [19] Zihao Li, Aixin Sun, and Chenliang Li. 2023. DiffuRec: A Diffusion Model for Sequential Recommendation. arXiv:2304.00686 [cs.IR]
- [20] Chengkai Liu, Jianghao Lin, Jianling Wang, Hanzhou Liu, and James Caverlee. 2024. Mamba4rec: Towards efficient sequential recommendation with selective state space models. *arXiv preprint arXiv:2403.03900* (2024).
- [21] Calvin Luo. 2022. Understanding Diffusion Models: A Unified Perspective. arXiv:2208.11970 [cs.LG]
- [22] Xu Ma, Pengjie Wang, Hui Zhao, Shaoguo Liu, Chuhan Zhao, Wei Lin, Kuang chih Lee, Jian Xu, and Bo Zheng. 2021. Towards a Better Tradeoff between Effectiveness and Efficiency in Pre-Ranking: A Learnable Feature Selection based Approach. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021). <https://api.semanticscholar.org/CorpusID:234742003>
- [23] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. SimpleX: A Simple and Strong Baseline for Collaborative Filtering. *CoRR* abs/2109.12613 (2021). arXiv:2109.12613 <https://arxiv.org/abs/2109.12613>
- [24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10684–10695.
- [25] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=St1giarCHLP>
- [26] Liangcai Su, Fan Yan, Jieming Zhu, Xi Xiao, Haoyi Duan, Zhou Zhao, Zhenhua Dong, and Ruiming Tang. 2023. Beyond Two-Tower Matching: Learning Sparse Retrievable Cross-Interactions for Recommendation. arXiv:2311.18213 [cs.IR]
- [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] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fb053c1c4a485aa-Paper.pdf
- [30] Wenjie Wang, Yitian Xu, Fulij Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023. Diffusion Recommender Model. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) (SIGIR '23). Association for Computing Machinery, New York, NY, USA, 832–841. <https://doi.org/10.1145/3539618.3591663>
- [31] Yu Wang, Hengrui Zhang, Zhiwei Liu, Liangwei Yang, and Philip S Yu. 2022. Contrastvae: Contrastive variational autoencoder for sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2056–2066.
- [32] Zhe Wang, Lipin Zhao, Biye Jiang, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2020. COLD: Towards the Next Generation of Pre-Ranking System. arXiv:2007.16122 [cs.IR]
- [33] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Wang, Taibai Xu, and Ed H. Chi. 2020. Mixed Negative Sampling for Learning Two-tower Neural Networks in Recommendations.
- [34] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-Bias-Corrected Neural Modeling for Large Corpus Item Recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) (RecSys '19). Association for Computing Machinery, New York, NY, USA, 269–277. <https://doi.org/10.1145/3298689.3346996>
- [35] Yantao Yu, Weipeng Wang, Zhoutian Feng, and Daiyue Xue. 2021. A dual augmented two-tower model for online large-scale recommendation. *DLP-KDD* (2021).
- [36] Yuan Zhang, Xue Dong, Weijie Ding, Biao Li, Peng Jiang, and Kun Gai. 2023. Divide and Conquer: Towards Better Embedding-based Retrieval for Recommender Systems From a Multi-task Perspective. arXiv:2302.02657 [cs.IR]
- [37] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2018. Deep Interest Evolution Network for Click-Through Rate Prediction. arXiv:1809.03672 [stat.ML]
- [38] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) (KDD '18). Association for Computing Machinery, New York, NY, USA, 1059–1068. <https://doi.org/10.1145/3219819.3219823>
- [39] Yunqin Zhu, Chao Wang, Qi Zhang, and Hui Xiong. 2024. Graph Signal Diffusion Model for Collaborative Filtering. arXiv:2311.08744 [cs.IR] <https://arxiv.org/abs/2311.08744>