

# Diffusion Cross-domain Recommendation

Yuner Xuan

Chalmers University of Technology  
Computer Science and Engineering Department  
Gothenburg, Sweden  
yuner@chalmers.se

## ABSTRACT

It is always a challenge for recommender systems to give high-quality outcomes to cold-start users. One potential solution to alleviate the data sparsity problem for cold-start users in the target domain is to add data from the auxiliary domain. Finding a proper way to extract knowledge from an auxiliary domain and transfer it into a target domain is one of the main objectives for cross-domain recommendation (CDR) research. Among the existing methods, mapping approach is a popular one to implement cross-domain recommendation models (CDRs). For models of this type, a mapping module plays the role of transforming data from one domain to another. It primarily determines the performance of mapping approach CDRs. Recently, diffusion probability models (DPMs) have achieved impressive success for image synthesis related tasks. They involve recovering images from noise-added samples, which can be viewed as a data transformation process with outstanding performance. To further enhance the performance of CDRs, we first reveal the potential connection between DPMs and mapping modules of CDRs, and then propose a novel CDR model named Diffusion Cross-domain Recommendation (DiffCDR)<sup>1</sup>. More specifically, we first adopt the theory of DPM and design a Diffusion Module (DIM), which generates user's embedding in target domain. To reduce the negative impact of randomness introduced in DIM and improve the stability, we employ an Alignment Module to produce the aligned user embeddings. In addition, we consider the label data of the target domain and form the task-oriented loss function, which enables our DiffCDR to adapt to specific tasks. By conducting extensive experiments on datasets collected from reality, we demonstrate the effectiveness and adaptability of DiffCDR to outperform baseline models on various CDR tasks in both cold-start and warm-start scenarios.

## CCS CONCEPTS

• Information systems → Recommender systems.

## KEYWORDS

Cross-domain Recommendation, Diffusion Probability Model, Cold-start Problem, Mapping Approach

## 1 INTRODUCTION

• **Introduction of CDR.** Recommender systems (RSs) have become a basis technique in various web and mobile applications, including Amazon (e-Commerce), YouTube (Video Platform), Facebook (Social Network Service) and Booking (Online Travel Agency). This field has attracted a large number of researchers, both from academia and industry.

Nonetheless, RSs still face significant challenges in real-world applications. The cold-start user problem is one of the most mentioned challenges. Since a large part of purchasing and rating records are generated by a few users [29], it can be difficult for RSs to provide satisfying recommendations for users who have no interaction in the past. These users are usually named cold-start users. Studies on cross-domain recommendation (CDR) [35] aim to deal with this issue.

The core motivation of CDR is to transfer knowledge from a source domain with richer interaction records of a set of users to a target domain without historical interactions of the same users [45, 48]. CDR models (CDRs) can alleviate the cold-start users problem and improve the recommendation results.

Various deep learning models have been proposed in recent years to boost the performance on knowledge transfer. Embedding-based mapping approach [48] is one of widely studied cross-domain knowledge transfer methods. The first step of this method is training collaborative filtering based models [12, 28, 35] to produce user's and item's embeddings in different domains respectively. Second, it transfers these embeddings across domains via various mapping functions. Typically, the mapping function is trained on data from the same users or items in different domains. In the prediction phase, the model maps corresponding embeddings from one domain to another.

Existing mapping approach CDRs fall into three main types. **1.Mapping function learning:** Mapping function learning methods [9, 21, 47] employ different techniques to learn mapping functions directly, which are used to transfer the knowledge across source and target domains. They take advantage of the data from overlapping users in different domains and learn the mapping patterns. **2.Meta learning:** Meta learning [22] is popular in recent RS studies [14, 24, 50]. For the works related to CDR [50, 51], they adopt meta-learning methods to generate mapping functions and variables to overcome the problem of cold-start. **3.Variational Autoencoder (VAE):** Due to the successful applications of VAE in RSs [3, 16, 34], an increasing number of researchers have proposed VAE-based methods [2, 17, 32] to tackle CDR tasks. These methods align the latent embedding variables of VAEs across domains through regularizers and alignment strategies. By narrowing the distribution divergence of latent variables between domains, they may generate better CDR results both for cold-start and warm-start users.

• **Analysis of existing methods.** The objective of the mapping approach CDRs is straightforward and intuitive. The essential challenge for this approach is how to learn mapping patterns effectively [48]. Although existing works have made significant progress as mentioned above, there are still several issues that deserve attention.

<sup>1</sup><https://github.com/breezeyuner/DiffCDR>

First, to have a better recommendation performance, CDRs are often optimized on task labels (task-oriented) [50], and the quality of the mapping function is treated as an intermediate and auxiliary factor. Therefore, the quality of the knowledge transferring could make way for task-specific objectives. As a consequence, CDRs can easily over-fit on task labels and be deteriorated by the noise existing in label data. According to our ablation experiments, there is no guarantee of achieving better performance by task-oriented learning strategy.

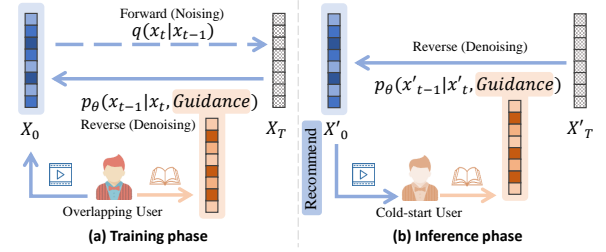
Second, the mapping functions employed in existing mapping-oriented works [9, 21, 42] which are likely to consider the transferring quality for each data sample, rather than narrow the distributional divergence across domains. It may restrict their generalization ability on unseen samples.

Therefore, if a CDR model is powerful to estimate the distribution of the target embedding space and also considers specific recommendation tasks, it is promising to achieve better recommendation performance.

• **Diffusion probability models (DPMs).** DPMs [4, 7, 26, 30] have achieved impressive results in image synthesis recently. In general, DPMs include forward and backward processes. In the forward process, a small amount of noise is gradually added to the original data, which can be overwhelmed by the noise for a sufficient number of steps. Oppositely, the noise will be removed in the reverse process. Since the noise can be simple (e.g. Gaussian noise) with small magnitude in each step of forward process, the denoised data generated in reverse process can achieve satisfying quality. One of the obvious deficiencies of diffusion models is the low speed of inference. In the reverse process, it takes hundreds or even thousands of steps from pure random noise to generate a final sample. A lot of works focus on solving this problem [1, 18, 33, 37], and they have accelerated the sampling process by dozens of times successfully.

• **Motivation.** As a state-of-the-art method of density estimation [10] and high-quality sampling for image data, DPMs is able to play the role as a powerful mapping module in CDRs. DPMs take noisy data as input and generate denoised results, which can be viewed as a process of transferring data from one distribution to another. This property makes the DPMs framework as a promising approach for knowledge transfer in CDR tasks.

To enhance the performance of the mapping functions by employing DPMs, we propose an innovative CDR framework named Diffusion Cross-domain Recommendation (DiffCDR) in this paper. **1.** In DiffCDR, we learn a Diffusion Module (DIM) to transfer knowledge across domains. DIM generates user features in the target domain by reversing the diffusion process conditioned on the corresponding user’s embeddings in source domain. We illustrate the concept of training and prediction phases in Figure (1). Our experiments results (RQ3) show that DIM mainly contributes to generate better quality transferred users embeddings than baseline models. **2.** As a generative method, randomness may be introduced into a DPM both in training and inference phases. To improve the stability of recommendation outcomes, we also design an Alignment Module (ALM). It ensures the consistency between the transformed user’s embeddings and ground-truth latent vectors of the same user in target domain. **3.** In addition, we also employ a target label data



**Figure 1: Illustration of the DiffCDR concept. (a) Learning the Diffusion Module (DIM)  $p_\theta$  on the embeddings of overlapping users (b) Generating transferred embeddings via reverse diffusion process for cold-start users conditioned on corresponding latent vectors from auxiliary domain.**

learning strategy to take the final recommendation quality into consideration.

• **Contributions.** We summarize the main contributions of our work as follows: **(1)** In order to improve the performance of mapping approach CDRs, we employ DPM framework and design a Diffusion Module (DIM), which transfers user’s preference from interaction data in auxiliary domain. To the best of our knowledge, this is the first work to apply DPM framework to CDR tasks. **(2)** To alleviate the negative influence of randomness and ensure the performance of cross-domain knowledge transfer, we employ an Alignment Module (ALM), which takes the transferred embeddings of DIM as input and generates aligned user embedding vectors. To ensure the recommendation performance, DiffCDR employs a task label data learning strategy as well. **(3)** For evaluation, we conduct extensive experiments on the Amazon reviews dataset to demonstrate the effectiveness and adaptability of DiffCDR in both cold-start and warm-start CDR scenarios. The ablation experiments show that all components of DiffCDR are critical to contribute to the improvements.

## 2 BACKGROUND

### 2.1 Problem Setting

In our CDR tasks, there is a source domain and a target domain. In each domain, we have a user set  $\mathcal{U} = \{u_1, u_2, \dots\}$ , an item set  $\mathcal{V} = \{v_1, v_2, \dots\}$ , and a rating matrix  $\mathcal{R}$ . For each element of  $\mathcal{R}$ ,  $r_{ij}$  denotes the rating by  $u_i$  on the item  $v_j$ . We use  $\mathcal{U}^s, \mathcal{V}^s, \mathcal{R}^s$  to denote the source domain, and  $\mathcal{U}^t, \mathcal{V}^t, \mathcal{R}^t$  for the target domain. Overlapping users between two domains are defined as  $\mathcal{U}^o = \mathcal{U}^s \cap \mathcal{U}^t$ . Commonly,  $\mathcal{V}^s$  and  $\mathcal{V}^t$  do not have overlapping elements in two domains.

The collaborative filtering-based models will transform the users and items into dense embedding vectors for source and target domain respectively.  $\mathbf{u}_i^s, \mathbf{v}_j^s$  denote the embedding vectors for  $u_i$  and  $v_j$  in source domain, and  $\mathbf{u}_i^t, \mathbf{v}_j^t$  are the corresponding vectors in target domain. The dimensionality of these vectors can be different or the same, here we set they all have  $k$  dimensions. The Embedding vectors can be viewed as the outputs of a pre-trained base model with a loss function defined as  $\mathcal{L} = \frac{1}{|\mathcal{R}|} \sum_{r_{ij} \in \mathcal{R}} (r_{ij} - \mathbf{u}_i^s \mathbf{v}_j^s)^2$ . We note that in addition to predicting ratings, the evaluation metrics

of recommender systems can be various such as click-through rate and ranking.

CDRs aim to transfer the knowledge from source domain, and improve the recommendation performance for  $\mathcal{U}^t$ . In order to evaluate the effectiveness of models,  $\mathcal{U}^o$  will be split into a train set and a test set. The ratings of users in train set, which includes both source and target domain interactions, will be used to train a CDR model. For users in the test set, their rating records in target domain will be used for evaluation.

## 2.2 Cross-domain Mapping

For CDR models with embedding-based mapping approach [48], mapping function is the core module they focus on. We use  $f_{map}$  to denote the mapping function. The user embedding in source  $\mathbf{u}_i^s$  can be transformed into target domain by  $\hat{\mathbf{u}}_i^t = f_{map}(\mathbf{u}_i^s)$ . To optimize the mapping function, we can define a commonly used loss function to minimize the distance between the transferred user embeddings and the truth user embeddings  $\mathbf{u}_i^t$ :

$$\mathcal{L}_{map} = \frac{1}{|\mathcal{U}^o|} \sum_{u_i \in \mathcal{U}^o} \|\hat{\mathbf{u}}_i^t - \mathbf{u}_i^t\|^n \quad (1)$$

The parameter  $n$  denotes  $L_1$ -norm when  $n = 1$ , or  $L_2$ -norm when  $n = 2$ . It reveals that the motivation of mapping function is having  $\hat{\mathbf{u}}_i^t$  closer to  $\mathbf{u}_i^t$ .

In terms of the recommendation part, CDR methods will combine  $\hat{\mathbf{u}}_i^t$  and item embedding in target domain  $\mathbf{v}_j^t$  to give a score which indicates the preference of  $u_i$  on  $v_j$ . Here we use the rating score directly and define the task loss function as follows:

$$\mathcal{L}_{task} = \frac{1}{|\mathcal{R}_o^t|} \sum_{r_{ij} \in \mathcal{R}_o^t} (r_{ij} - \hat{\mathbf{u}}_i^t \mathbf{v}_j^t)^2 \quad (2)$$

$\mathcal{R}_o^t = \{r_{ij} | u_i \in \mathcal{U}^o, v_j \in \mathcal{V}^t\}$  denotes the ratings of overlapping users in target domain.

## 2.3 Diffusion Probabilistic Models (DPMs)

We put the details of DPMs in Appendix (A) to keep coherence.

## 3 MODEL

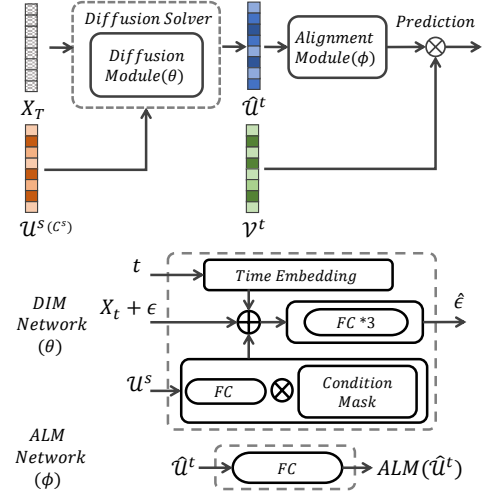
We illustrate the architectures of DiffCDR with the network of DIM and ALM in Figure (2).

### 3.1 Diffusion Module (DIM)

Starting from random noise  $\mathbf{x}_T \sim \mathcal{N}(0, \mathcal{I})$ , we design the DIM to remove the noise via reverse process and produces  $\mathbf{x}_0$  as transferred user embeddings  $\hat{\mathbf{u}}_i^t$ . It is guided by user's features in source domain. For CDR tasks, it is natural to expect  $\hat{\mathbf{u}}_i^t$  can reflect the preference of that user in source domain. Based on Equation (9), we define the classifier-free guidance score function as:

$$\hat{\epsilon}_\theta(\mathbf{x}_t | \mathbf{u}_i^s) = \epsilon_\theta(\mathbf{x}_t) + s \cdot (\epsilon_\theta(\mathbf{x}_t | \mathbf{c}_i^s) - \epsilon_\theta(\mathbf{x}_t)) \quad (3)$$

where  $\mathbf{c}_i^s$  is the conditional guidance embedding  $\mathcal{C}^s$  of  $u_i$  in source domain. In each step, we can derive  $\mathbf{x}_{t-1}$  from  $\hat{\epsilon}_\theta(\mathbf{x}_t | \mathbf{u}_i^s)$  [7] and obtain the sample  $\mathbf{x}_{t-1}$  by  $\mathbf{x}_{t-1} = \text{DIM}(\hat{\epsilon}_\theta, \mathbf{x}_t, \mathbf{c}_i^s, t)$ . By utilizing DIM repeatedly, we can have  $\mathbf{x}_T, \mathbf{x}_{T-1}, \dots, \mathbf{x}_0$ . We use the final step sample  $\mathbf{x}_0$  as the transferred user embeddings in target domain, that is  $\hat{\mathbf{u}}_i^t = \mathbf{x}_0$ .



**Figure 2: Architecture of DiffCDR.** It consists of a **Diffusion Module (DIM)** and an **Alignment Module (ALM)**. We also employ a diffusion solver to accelerate the inference of the DIM. The recommendation results are produced by element-wisely multiplication with the output of ALM and the item embeddings in target domain.

- **Structure of DIM.** Many image generative diffusion models [10, 15, 37] use U-net [31] as their backbone. Considering the dimensionality of the data in our tasks, we employ layers of perceptron to construct DIM, which already achieve good performance in our experiments. In order to be compatible with diffusion solvers, we add sinusoidal position embedding [41] to build time step embeddings. We utilize linear layers to adjust the time step embedding and the conditional guidance to have the same units as the input. We then add these three parts element-wisely to give the estimation of added noise  $\hat{\epsilon}$ .

- **Fast sampling of DIM.** The inference of DIM can be time-consuming. Instead of a step-by-step evaluation with DIM, we adopt a fast inference solver for diffusion ODEs to speed up the sampling process with much fewer steps:

$$\hat{\mathbf{u}}_i^t = \mathbf{x}_0 = \text{Solver}(\hat{\epsilon}_\theta, \mathbf{x}_T, \mathbf{c}_i^s) \quad (4)$$

We can choose the *Solver* from existing works [1, 18, 25]. Moreover, the fast inference of DIM enables us to have a stable training procedure for a specific recommendation task.

### 3.2 Alignment Module (ALM)

Since the randomness could be introduced by diffusion model both in training and inference phases, we employ an alignment module (ALM) to ensure the consistency between the generated user embedding  $\hat{\mathbf{u}}_i^t$  from DIM and the ground-truth user representation in target domain  $\mathbf{u}_i^t$ . We then obtain the rating prediction as  $\hat{r}_{ij} = \text{ALM}_\phi(\hat{\mathbf{u}}_i^t) \mathbf{v}_j^t$ . The motivation of ALM is to alleviate the effects of randomness, which improves the quality of user preference transferring.

In our experiments, we implement ALM by simply adding one fully connected layer  $\text{FC}(\hat{\mathbf{u}}_i^t)$ , which takes  $\hat{\mathbf{u}}_i^t$  as input. We note that

the ALM along cannot not give satisfying recommendation results. In the ablation experiments, we analyze the influence of ALM in detail.

### 3.3 Loss Computation

• **DIM Loss.** In order to generate a training sample, we choose a  $t \in [0, T]$  randomly, and add noise by pre-defined noise schedule on  $\mathbf{u}_i^t$  as

$$q(\mathbf{x}_t | \mathbf{u}_i^t) = \mathcal{N}(\mathbf{x}_t; \tilde{\alpha}(t)\mathbf{u}_i^t, \sigma^2(t)\mathcal{I}) \quad (5)$$

Following the loss defined in (8), the loss function of DIM is:

$$\mathcal{L}_{DIM_\theta} = \mathcal{L}_{\hat{\epsilon}_\theta} = \mathbb{E}_{t \in [0, T], \mathbf{x}_0 \sim q(\mathbf{u}_i^t), \epsilon \sim \mathcal{N}(0, \mathcal{I})} [\|\epsilon - \hat{\epsilon}_\theta(\mathbf{x}_t, t | \mathbf{c}_i^s)\|],$$

where  $\epsilon \sim \mathcal{N}(0, \mathcal{I})$ , and  $\hat{\epsilon}_\theta$  is defined in (3). For the purpose of training  $\epsilon_\theta(\mathbf{x}_t, t)$  and  $\epsilon_\theta(\mathbf{x}_t, t | \mathbf{c}_i^s)$  at the same time, we will mask the conditional information  $\mathbf{c}_i^s$  randomly during training phase with a certain ratio.

• **ALM and Task Loss.** To form the ALM loss function, we add the ALM into the  $\mathcal{L}_{map}$  defined in (1):

$$\mathcal{L}_{ALM_\phi} = \frac{1}{|\mathcal{U}^o|} \sum_{\mathbf{u}_i \in \mathcal{U}^o} \|\text{ALM}_\phi(\hat{\mathbf{u}}_i^t) - \mathbf{u}_i^t\|^n \quad (6)$$

Following the architecture of PTUPCDR[50], we also form a task loss to optimize the our model with ratings data directly. This task-oriented strategy has our model adapt to the specific recommendation tasks. Consider both the loss defined in (2), we give the task loss function as

$$\mathcal{L}_{task_\phi} = \frac{1}{|\mathcal{R}_o^t|} \sum_{r_{ij} \in \mathcal{R}_o^t} (r_{ij} - \text{ALM}_\phi(\hat{\mathbf{u}}_i^t)\mathbf{v}_j)^2.$$

Since  $\mathcal{L}_{ALM_\phi}$  and  $\mathcal{L}_{task_\phi}$  both depend on the output of DIM and share same trainable variables  $\phi$ , we optimize these two losses together via:

$$\mathcal{L}_{ALM\_task_\phi} = \mathcal{L}_{ALM_\phi} + \lambda \mathcal{L}_{task_\phi} \quad (7)$$

where  $\lambda$  is used to adjust the weight.

We will also analyze the influence of task loss in ablation experiments.

### 3.4 Training Strategy

In the training phase, we first need to generate the transferred user embeddings via DIM and then use them as input to ALM. We choose DPM-solver [19] as the fast inference solver here, which enables us to accelerate the inference process and generate  $\mathbf{x}_0$  with much fewer steps. Therefore, we take a training strategy to optimize  $\mathcal{L}_{DIM_\theta}$  and  $\mathcal{L}_{ALM\_task_\phi}$  alternately in each batch.

We describe the training procedure formally in Algorithm (1).

## 4 EXPERIMENTS

• **Research Questions (RQ).** We intend to answer the following research questions via experiments: **RQ1.**How does DiffCDR perform on CDR tasks comparing to state-of-art baseline methods? **RQ2.**How do DIM and ALM contribute to the whole method? What is the influence of task loss on the performance for DiffCDR? **RQ3.**Why can we have better performance with DiffCDR? **RQ4.**As a generative-based method, what is the throughout of DiffCDR?

### Algorithm 1 Training Procedure of Diffusion Cross-Domain Recommendation (DiffCDR)

**Input:**  $\mathcal{U}^s, \mathcal{V}^s, \mathcal{R}^s, \mathcal{C}^s, \mathcal{U}^o, \mathcal{V}^t, \mathcal{R}^t$

**Input:** Time steps  $t \in [0, T]$ , Noise schedule  $\tilde{\alpha}(t), \sigma(t)$ .

**Input:** Number of training epochs  $N$

**Input:** Diffusion score function  $\hat{\epsilon}_\theta$ .

**Input:** Alignment module  $\text{ALM}_\phi$ .

**Input:** Fast inference solver *Solver*.

**Pre-training Stage:**

1: Learning a source model with  $\mathbf{u}^s, \mathbf{v}^s$ .

2: Learning a target model with  $\mathbf{u}^t, \mathbf{v}^t$ .

**DiffCDR Training Stage:**

3: Initializing  $\hat{\epsilon}_\theta$  randomly

4: Initializing  $\text{ALM}_\phi$  randomly

5: **for all**  $N$  **do**

6:   **for all** *Batches* **do**

7:     Generating training samples by Equation (5)

8:     Updating  $\text{DIM}_\theta$  by minimizing  $\mathcal{L}_{DIM_\theta}$

9:     Inferring  $\hat{\mathbf{u}}^t$  by Equation (4)

10:    Updating  $\text{ALM}_\phi$  by minimizing  $\mathcal{L}_{ALM\_task_\phi}$

11:   **end for**

12: **end for**

### 4.1 Experimental Settings

Table 1: Data Statistics of CDR tasks.

CDR Tasks	Domain		Item		Rating	
	Source	Target	Source	Target	Source	Target
Task1	Video	Music	50 K	64 K	1,697 K	1,097 K
Task2	Book	Video	368 K	50 K	8,898 K	1,698 K
Task3	Book	Music	368 K	64 K	8,898 K	1,097 K

CDR Tasks	Domain		Overlap	User	
	Source	Target		Source	Target
Task1	Video	Music	18 K	124 K	75 K
Task2	Book	Video	37 K	604 K	124 K
Task3	Book	Music	17 K	604 K	75 K

• **Dataset.** Following the previous works [9, 42, 46, 50], we utilize three real-word public datasets built on Amazon review data<sup>2</sup>. We use the 5-core subsets, in which all users and items have at least 5 reviews. And we choose 3 categories with large data volume from all 24 ones, which are movies\_and\_tv (Video), cds\_and\_vinyl (Music), and books (Book). We set three CDR tasks on Amazon dataset: Task 1: Video  $\rightarrow$  Music, Task 2: Book  $\rightarrow$  Video, and Task 3: Book  $\rightarrow$  Music. The data statistics are in Table (1).

• **Metrics.** Since the Amazon reviews data contains the rating records (0-5), it is a typical regression problem. We follow [9, 46, 50] and adopt the metrics including Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). In addition, we report the Normalized DCG (nDCG) and hit rate for highest 20 scores (N@20, H@20). Noting that the nDCG and hit rate scores are calculated by ranking in all target domain items. To obtain these ranking metrics, we conduct additional experiments by adding same negative samples to each CDR models. For each experiment, we report the mean values of five runs.

• **Construction Datasets.** In terms of data split, we follow a similar strategy as [50]. First, we randomly select a fraction of overlapping users as test users, and the remaining overlapping users as training

<sup>2</sup><http://jmcauley.ucsd.edu/data/amazon/>

users. We utilize their rating records to train CDR baseline models and our DiffCDR. We set the proportion of test users  $\beta$  as three cold-start levels: 20%, 50% and 80% of all overlapping users. The larger the  $\beta$  is, the more difficult a CDR task will be. In the cold-start experiments, the ratings of test users are not available during training. For the warm-start experiments, we divide 50% of test users' rating records as fine-tune data. During that data split operation, we consider the order of time series to avoid potential leakage.

• **Baselines.** (1) **TGT** denotes a simple MF model, which is only trained on target domain data. (2) **CMF** [35] shares same embeddings for overlapping users belonging to different domains. It is collectively trained on both datasets from source and target domains. (3) **EMCDR** [21] is a classic CDR model which adopts a network as the mapping function to transfer the user embeddings from source domain into target domain. The initial user embeddings in this method are produced by two separate MF models that are trained in two domains separately. (4) **SSCDR** [9] is an embedding-based mapping approach model which leverages semi-supervised method. (5) **LACDR** [42] employs encoder-decoder structure to form the mapping function. It aligns the low-dimension embedding spaces of different domains. (6) **PTUPCDR** [50] is a meta mapping approach CDR method. It generates personalized mapping functions (bridge functions) by a meta-network for each user, which transfer knowledge across domains. We discuss VAE framework CDRs in Section 5.2 and explain why we do not include VAE-based baselines.

• **Experimental Steps.** We conduct our experiments in the following steps: (1) Train all models on the training dataset which only includes training users. In each group of experiments, we use the same pre-trained base models to produce embedding variables. (2) Evaluate the exact cold-start performance of models on test datasets. (3) Finally, we will fine-tune the models with warm-start samples, and evaluate the warm-start performance.

• **Hyper-parameters Setting.** (1) We set the dimensionality to 10 for both user and item embedding variables. (2) We use the Adam optimizer and set learning rate as 0.001. (3) To train the score function with and without guidance simultaneously, the overall masking probability for condition input is 10%. (4) We set the weight parameter  $\lambda$  in Equation (7) by grid search among [0.01, 0.1, 1, 2, 4, 8]. The parameter  $n$  in the Equation (6) is set as  $n = 1$ , which obtains better performance than  $n = 2$ . (5) Instead of Gaussian noise, we directly utilize source-domain user embeddings as input to DIM for inference phase in our experiments. We find that it can further reduce the number of inference steps while maintaining similar performance. (6) In terms of the architecture of DIM, we use 3 layers of perceptrons with 128 hidden units (10 units for input and output). For the ALM, we utilize one fully connected layer without bias variable, which has the same input and output dimensionality as user embedding (=10). (7) The value of guidance strength  $s$  in Equation (3) is set as 0.1 in our experiments, which is decided by grid search from [0.05, 0.1, 0.2, 0.5, 1, 2, 5]. The number of function evaluations used in DPM-solver is 30, which trades off quality against inference speed.

## 4.2 Cold- and Warm-start Experiments (RQ1)

We present the results of cold- and warm-start experiments and analyze the performance of DiffCDR for each scenario in this section.

• **Cold-start Experiments.** Following the prior works [50], we evaluate DiffCDR in 3 CDR tasks with different values of  $\beta$ . We have several insights from the experimental results in Table (2). (1) **TGT** only takes the data from target domain, that means the embedding variables of test users are untrained due to data sparsity. As a result, TGT falls behind in the experiments. In contrast, other methods utilize data from both domains and achieve better results. Consequently, exploiting additional data from auxiliary domain is an effective way to promote the recommendation quality. (2) **CMF** is a variant of TGT, it combines the data from source and target domains. Thus its embedding space is shared across domains, and it obtains better results than TGT model. Due to the differences in the embedding spaces between two domains and CMF does not consider this fact, CDRs outperform the CMF in our experiments. For CDRs, they transfer the knowledge from source to target domain through various mapping functions and alleviate the discrepancy of embedding spaces. It is valuable to design a certain mechanism to extract supplementary information from source domain to boost the performance of cold-start recommendation. (3) Task2 may be easier than the other two, such that the differences between models are close, but DiffCDR still achieves comparable performance on it. All results show the advantage of DiffCDR in cold-start CDR tasks.

• **Warm-start Experiments.** The warm-start scenario represents a situation that a fraction of the interactions from test users are available during the training phase, which we can expect to encounter frequently in real-world applications. It can be viewed as a pre-trained stage for initialization [50]. With the CDR methods, this initialization can help a recommender system to generate better results.

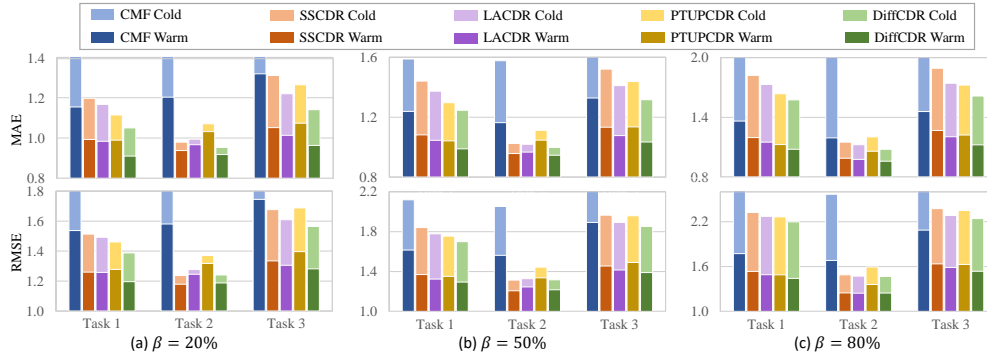
We validate CMF, SSCDR, LACDR, PTUPCDR and our DiffCDR. From the results in Figure 3, we have insights as follows: (1) Additional data can be helpful to improve recommendation performance. All models have better metrics than the cold-start scenario. (2) CDRs not only achieve better performance for cold-start tasks, they also can improve the results further with more interaction data. The results show that CDRs are valuable for real-world applications (3) Again, it is difficult to gain further improvement on task 2, but DiffCDR still has better results in all experiments. Together with Table (2), they demonstrate the effectiveness and adaptability of DiffCDR in both cold-start and warm-start scenarios.

## 4.3 Ablation Experiments (RQ2)

To show the contributions of DIM, ALM and task loss, we conduct ablation experiments of cold-start scenario. **Ablation 1: DiffCDR DA.** We removes task loss from DiffCDR and only remain DIM and ALM ("DA") to have fair comparison to the mapping-oriented baselines (denoted as "CDR M", including CMF, EMCDR, SSCDR and LACDR) which are not directly optimized on target domain label data. Here  $n$  in the Equation (6) is set as  $n = 2$  which obtains better results. **Ablation 2: DiffCDR AT.** We only includes ALM and task loss learning ("AT") in DiffCDR to illustrate the effect of DIM. Since a single ALM is equivalent to EMCDR, we do not conduct separate ablation experiments for this setting. Noting that we do not conduct

**Table 2: Metrics of cold-start experiments. Each result is an average of five runs. \* means significant difference by paired t-test with  $\rho = 0.05$  compared with the best baseline. The column of "Improve" indicates the rate of improvement.**

	$\beta$	Metrics	TGT	CMF	EMCDR	SSCDR	LACDR	PTUPCDR	DiffCDR	Improve
Task1 Video -> Music	20%	MAE	4.4546	1.4642	1.3596	1.1757	1.1295	1.1099	<b>1.0435*</b>	<b>6.0%</b>
		RMSE	5.1338	1.9571	1.6615	1.4911	1.4358	1.4543	<b>1.3840*</b>	<b>3.6%</b>
		N@20	0.00253	0.00508	0.00977	0.00932	0.00984	0.00978	<b>0.01026*</b>	<b>4.3%</b>
		H@20	0.00033	0.00084	0.00229	0.00212	0.00228	0.00236	<b>0.00238*</b>	<b>1.1%</b>
	50%	MAE	4.4884	1.6710	1.6891	1.4320	1.3502	1.2842	<b>1.2367*</b>	<b>3.7%</b>
		RMSE	5.1790	2.2076	2.0368	1.8248	1.7510	1.7340	<b>1.6859*</b>	<b>2.8%</b>
		N@20	0.00251	0.00403	0.00898	0.00793	0.00893	0.00828	<b>0.00915*</b>	<b>1.9%</b>
		H@20	0.00033	0.00068	0.00193	0.00164	0.00199	0.00179	<b>0.00202*</b>	<b>1.9%</b>
	80%	MAE	4.4959	2.2327	2.1980	1.8162	1.6886	1.6174	<b>1.5606*</b>	<b>3.5%</b>
		RMSE	5.1830	2.8868	2.5713	2.3090	2.2238	2.2429	<b>2.1754*</b>	<b>2.2%</b>
		N@20	0.00248	0.00348	0.00622	0.00578	0.00606	0.00545	<b>0.00665*</b>	<b>6.9%</b>
		H@20	0.00033	0.00051	0.00124	0.00111	0.00124	0.00107	<b>0.00136*</b>	<b>9.7%</b>
Task2 Book -> Video	20%	MAE	4.1807	1.4742	1.1305	0.9774	0.9681	1.0728	<b>0.9476*</b>	<b>2.1%</b>
		RMSE	4.7496	1.9180	1.4215	1.2356	<b>1.2311</b>	1.3745	1.2338	-0.2%
		N@20	0.00245	0.00578	0.01898	0.02066	0.01850	0.01821	<b>0.02073</b>	<b>0.3%</b>
		H@20	0.00043	0.00124	0.0064	0.00676	0.0056	0.00594	<b>0.00697*</b>	<b>3.1%</b>
	50%	MAE	4.1951	1.5651	1.1863	1.0193	1.0077	1.1116	<b>0.9953</b>	<b>1.2%</b>
		RMSE	4.7693	2.0341	1.4993	1.3089	<b>1.3051</b>	1.4425	1.3155	-0.8%
		N@20	0.00274	0.00536	0.01924	0.02041	0.01875	0.01785	<b>0.02047</b>	<b>0.3%</b>
		H@20	0.00044	0.00107	0.00642	0.00675	0.00535	0.00575	<b>0.0068</b>	<b>0.7%</b>
	80%	MAE	4.2384	2.2379	1.3445	1.1469	1.1151	1.2072	<b>1.0846*</b>	<b>2.7%</b>
		RMSE	4.8198	2.7538	1.6946	1.4871	<b>1.4660</b>	1.5987	1.4695	-0.2%
		N@20	0.00258	0.00412	0.01906	0.01949	0.01710	0.01520	<b>0.01960</b>	<b>0.6%</b>
		H@20	0.00040	0.00073	0.00628	<b>0.00636</b>	0.00512	0.00484	0.00634	-0.3%
Task3 Book -> Music	20%	MAE	4.5190	1.7976	1.6425	1.3073	1.1945	1.2556	<b>1.1220*</b>	<b>6.1%</b>
		RMSE	5.1838	2.3545	1.9873	1.6599	1.5771	1.6730	<b>1.5390*</b>	<b>2.4%</b>
		N@20	0.00196	0.00383	0.01193	0.01179	0.01367	0.01006	<b>0.01374</b>	<b>0.5%</b>
		H@20	0.00035	0.00071	0.00323	0.00313	0.0037	0.00275	<b>0.00382*</b>	<b>3.2%</b>
	50%	MAE	4.4953	2.0002	1.9364	1.5183	1.3925	1.4304	<b>1.3077*</b>	<b>6.1%</b>
		RMSE	5.1685	2.6001	2.2966	1.9467	1.8644	1.9475	<b>1.8255*</b>	<b>2.1%</b>
		N@20	0.00200	0.00341	0.00994	0.00964	0.01058	0.00804	<b>0.01082*</b>	<b>2.3%</b>
		H@20	0.00028	0.00059	0.00253	0.00247	0.00277	0.00206	<b>0.00281*</b>	<b>1.7%</b>
	80%	MAE	4.5133	2.5014	2.3448	1.8849	1.7107	1.7016	<b>1.5871*</b>	<b>6.7%</b>
		RMSE	5.1960	3.1740	2.7035	2.3517	2.2468	2.3248	<b>2.2110*</b>	<b>1.6%</b>
		N@20	0.00170	0.00275	0.00705	0.00652	0.00658	0.00682	<b>0.00722*</b>	<b>2.3%</b>
		H@20	0.00027	0.00046	0.00176	0.00158	0.00165	0.00107	<b>0.00179*</b>	<b>1.6%</b>



**Figure 3: Results of warm-start experiments. The light and dark colored represent the cold-start and warm-start respectively.**

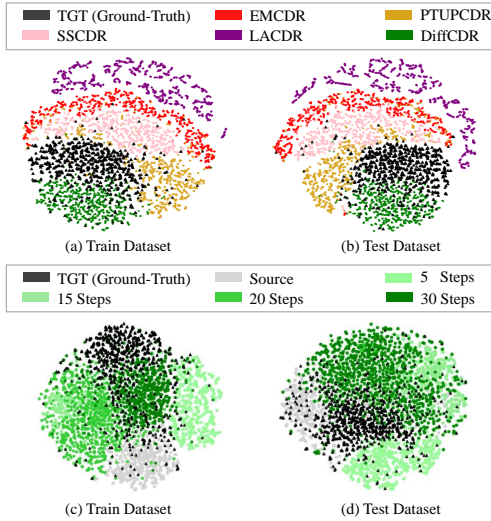
**Table 3: Metrics of ablation experiments. <sup>†</sup> "Best CDR M" denotes the best results of CDR models without target label learning, including CMF, EMCDR, SSDDR and LACDR. <sup>‡</sup> "DA" denotes the DiffCDR only includes DIM and ALM, "AT" denotes the DiffCDR only includes ALM and target label data learning, while "DAT" denotes the full version of DiffCDR.**

Metric		MAE									RMSE								
$\beta$		20%			50%			80%			20%			50%			80%		
CDR Task		Task1	Task2	Task3	Task1	Task2	Task3	Task1	Task2	Task3	Task1	Task2	Task3	Task1	Task2	Task3	Task1	Task2	Task3
Mapping-Oriented	Best CDR M <sup>†</sup>	1.1295	0.9681	1.1945	1.3502	1.0077	1.3925	1.6886	1.1151	1.7107	1.4358	<b>1.2311</b>	1.5771	1.7510	<b>1.3051</b>	1.8644	2.2238	<b>1.4660</b>	2.2468
	DiffCDR DA <sup>‡</sup>	1.0843	0.9506	1.1843	1.2764	0.9976	1.3549	1.6458	1.1007	1.6425	1.4157	1.2354	1.5961	1.7241	1.3175	1.8650	2.2259	1.4770	2.2426
Task-Oriented	PTUPCDR	1.1099	1.0728	1.2556	1.2842	1.1116	1.4304	1.6174	1.2072	1.7016	1.4543	1.3745	1.6730	1.7340	1.4425	1.9475	2.2429	1.5987	2.3248
	DiffCDR AT	1.1184	1.0912	1.2790	1.2888	1.1257	1.4311	1.6002	1.2093	1.6814	1.4597	1.3947	1.6995	1.7321	1.4581	1.9377	2.2019	1.5968	2.2822
Mapping & Task	DiffCDR DAT	<b>1.0435</b>	<b>0.9476</b>	<b>1.1220</b>	<b>1.2367</b>	<b>0.9953</b>	<b>1.3077</b>	<b>1.5606</b>	<b>1.0846</b>	<b>1.5871</b>	<b>1.3840</b>	1.2338	<b>1.5390</b>	<b>1.6859</b>	1.3155	<b>1.8255</b>	<b>2.1754</b>	1.4695	<b>2.2110</b>

additional experiments on ALM module, since the ALM alone is equivalent to EMCDR[21].

We summarize the observation of results from Table (3) as follows. **(1)** Since label data could include noises, target label data





**Figure 4: User embedding visualization.** We randomly sample 1000 users and visualize embeddings of these users in both train and test stages. (a,b) involve different CDRs, and (c,d) are produced by DiffCDR with various inference steps.

learning methods do not always have better performance. The effect of task-orientated methods varies according to different setting of  $\beta$  and tasks. (2) Even without DIM, DiffCDR DA already achieves at least comparable performance to other mapping-oriented CDRs. (3) DiffCDR DAT includes all modules which both consider the mapping and task factors. It further improves the performance of DiffCDR DA. Based on these ablation experiments, DIM, ALM and task label learning are all critical to contribute to the superior performance of DiffCDR.

#### 4.4 Improvement Analysis (RQ3)

In this section, we analyze the user embeddings generated by four CDRs including our DiffCDR. We reveal the relation between the ground-truth embeddings in target domain and transferred factors.

We use t-SNE [40] implemented in Scikit-learn package to visualize user embeddings in Task1 with  $\beta = 0.2$ . To obtain the best user latent factors as ground-truth, we add the test users into train set and train an additional TGT model. The setting of data splitting keeps unchanged for other CDR models. For DiffCDR, we derive the user embeddings directly from DIM rather than the output of ALM, since we prefer to show the ability of knowledge transfer across domains. We randomly sample 1000 users and all CDRs share the same base models.

• **Comparison of CDRs.** Figure 4(a,b) visualize the embeddings of users in training and test set respectively. Each color denotes a CDR model, and black triangles are produced by TGT. Our observations are as follows. (1) A satisfied CDR model is able to transfer the factors from one domain to another, and the transferred factors can be fused with the target data points. Although the boundaries between transferred user features and the factors generated by TGT in figures are obvious, DiffCDR (green) has a better fusion with ground-truth (black). For other baselines, either the transferred

user features are clearly separated from black triangles or there is a large distance. We can see similar patterns in the train Figure 4(a) and test dataset Figure 4(b). (2) The group distance between TGT and DiffCDR is the shortest compared to other methods, it implies that our DiffCDR has the best result in transferring the knowledge into target domain. The most important reason why DiffCDR achieves competitive results is that the distribution of transferred user embeddings by DiffCDR is closer to the latent factors of target domain.

• **Comparison of Inference Steps with DiffCDR.** To illustrate the effects of inference step, we visualize the user embedding from DiffCDR with various setting of function evaluation numbers in DPM-solver as Figure 4 (c,d). The grey dots are samples from source domain. They show that the distance between the transferred and ground-truth user embeddings shrinks as the number of inference steps increases. It also demonstrates how DIM transfers user embeddings across domains during the reverse diffusion process.

#### 4.5 Throughput Analysis (RQ4)

**Table 4: Throughput Statistics of CDR (K samples/second)**

Phrase	EMCDR	SSCDR	LACDR	PTUPCDR	DiffCDR
Training	39.2 K	8.3 K	7.5 K	32.7 K	2.9 K
Test	125.1 K	133.4 K	131.5 K	100.6 K	61.5 K

We report the mean values of CDRs throughput in Table (4). As generative model, DPM involves time consuming sampling process. With fast diffusion inference solver, our DiffCDR has acceptable throughput. The statistics show that DiffCDR has 38.7% throughput of SSCDR for training and 61.1% throughput of PTUPCDR for inference. It is worth noting that we conduct our experiments with 1 core of 2GHz Xeon(R) CPU and 1 core of P100 GPU. Therefore, we could reduce the time consumption further by sampling in parallel.

### 5 RELATED WORK

#### 5.1 Mapping Approach CDRs

Mapping approach CDRs [48] mainly transfer pre-trained embeddings across domains by various methods. EMCDR [21] learns a mapping function to infer latent factors for new items/users in target domain. DCDCSR [47] considers the sparsity degree to construct benchmark factors with top-K similar entities first, and trained a model to map latent factors into these factors. SSCDR [9] is a semi-supervised method, it consists of a supervised learning module trained on overlapping user interactions in different domains, and an unsupervised module that learns from records of user and item interactions in the source domain. LACDR [42] employs encoder-decoder architecture, which includes separate encoders and decoders for different domains. It minimizes the reconstruction errors for each domain and the alignment errors of the low-dimensional hidden variables produced by encoders across domains.

Inspired by the successful applications of meta-learning, researchers have proposed a number of meta-based CDRs recently. MWUF [51] takes item features and interacted users' features to generate a meta scaling network and a meta shifting network respectively, which warms up cold-start items' embedding. TMCDR

[49] adopts meta learning to alleviate over-fitting problems on limited data of overlapping users. It divides the samples of overlapping users into two groups to construct training tasks, which contain a learning phase and a cold-start phase. PTUPCDR [50] is a method which considers the various preferences for each user. Instead of learning a shared constant mapping function, this model trains a meta network which generates a personalized mapping function (bridges) for each user.

Our DiffCDR belongs to the class of mapping approach CDRs as well. We leverage probabilistic diffusion model as a powerful method for distribution transforming. It transfers knowledge by generating user embeddings in target domain conditioned on source domain guidance.

In addition, there are also many other works to solve CDR falling into mapping approach, but we cannot compare with them directly. CATN [46] and RC-DFM[5] leverage reviews of users to extract aspect level information, and performs recommendation by aspect correlations across domains. DACIR [44] aligns the cross-domain embeddings with a reinforcement learning based interactive recommender method, which takes both the static and dynamic interaction patterns into consideration. One-to-many transfer method [13] adopts meta transferring to implement CDRs on multiple domains rather than apply separate models. Ma et al. [20] propose a  $\pi$ -Net which contains a shared account filter unit and a cross-domain transfer unit. This model gives recommendations sequentially for same users in different domains.

## 5.2 VAE Framework CDRs

Recently, variational autoencoders (VAEs) based recommendation methods have achieved excellent results [16, 34, 39]. In terms of CDR problems, VAE framework-based models focus on the alignment of latent embedding spaces across domains, which improves the recommendation quality by exploiting domain-sharing information. SA-VAE [32] utilizes a trained VAE model in source domain to operate soft-alignment, and ensures the variational models in different domains share similar latent spaces. CDRIB [2] creates global graph structure of users and items by variational graph encoder and creates domain-shared representations with information bottleneck regularizers. VDEA [17] extracts user preference by variational inference first, and then generates user groups to cluster users with similar rating behaviors. Both information captured from overlapping users and non-overlapping users can be exploited to improve cross-domain performance in this method.

The VAE framework CDRs need to simultaneously approximate the posterior distribution and optimize recommendation task. For that reason, VAE framework CDRs cannot directly utilize user representations from pre-trained models. Therefore, it is difficult to have a controlled comparison with our method, even if both VAEs and DPMs are generative models.

## 5.3 Diffusion Probabilistic Models (DPMs)

To train a generative model, VAEs normally need to build a posterior model at the same time. GANs require an additional discriminator network in addition to the generator, and Normalizing Flow models often require invertible transformation functions. In contrast to those methods, DPMs only need to train a generator to fit each

step of reverse diffusion process. As a result, DPMs have fewer constraints on the choice of model architecture.

Following the early work [36] of this topic, DDPM [7] achieves impressive success in image generation tasks. After that, conditional DPM [4] improves the performance of image synthesis tasks by class-conditional generation. Classifier-free guided DPM [8] provides generation guidance without a separate classifier model. Large scale image generation APIs based on DPMs also have become popular [23, 27, 30] recently. Besides image synthesis, researchers have proposed innovative diffusion models in other fields [6, 11, 15].

DiffRec[43] leverages diffusion model recommendation tasks. It corrupts users' historical interactions in forward steps and recover original interactions iteratively in reverse process. The recovered results are used to recommend non-interacted items. Since we focuses on the CDR problem, the application and motivation of DiffRec are different from our approach.

Time-consuming inference is one obvious deficiency of DPMs. In order to deal with this it, DDIM [37] adopts non-Markovian noising process to reduce the steps in inference process. Distillation is also an effective solution to accelerate the sampling procedure [33]. When we consider the DPMs from continues-time aspect [38], researchers propose SED solvers [1, 25] and ODE solvers [18, 19] for DPMs, which can be used to perform inference dozens of times faster.

## 6 CONCLUSION AND FUTURE WORK

We study the purpose and existing methods of CDR in this paper first, and explain the mechanism of CDRs with mapping approach in detail. we also investigate DPMs in image synthesis applications, and reveal the motivations of combining DPMs with recommendation models to solve CDR tasks. We then propose a novel Diffusion Cross-domain CDR (DiffCDR) method. In our DiffCDR, a DIM is in charge of cross-domain knowledge transferring and an Alignment Module to have stabilized recommendation results. Moreover, the task-oriented loss function further improves its performance. Based on extensive experiments with three real-world datasets, the results reveal the superiority of DiffCDR in a variety of cold-start and warm-start CDR tasks. Moreover, ablation experiments show that all the components of DiffCDR contribute critically to its performance.

This work explores a new direction to tackle CDR challenges by DPMs with guidance. For the future, we will continue to emphasize the importance of the diffusion model architecture specially for CDR tasks. In addition, it is also meaningful to explore the effects of other noise schedule for recommendation applications.



## REFERENCES

- [1] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. 2022. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503* (2022).
- [2] J. Cao, J. Sheng, X. Cong, T. Liu, and B. Wang. 2022. Cross-Domain Recommendation to Cold-Start Users via Variational Information Bottleneck. (may 2022), 2209–2223. <https://doi.org/10.1109/ICDE53745.2022.00211>
- [3] Yifan Chen and Maarten de Rijke. 2018. A collective variational autoencoder for top-n recommendation with side information. In *Proceedings of the 3rd workshop on deep learning for recommender systems*. 3–9.
- [4] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 8780–8794.
- [5] Wenjing Fu, Zhaohui Peng, Senzhang Wang, Yang Xu, and Jin Li. 2019. Deeply Fusing Reviews and Contents for Cold Start Users in Cross-Domain Recommendation Systems (AAAI’19/IAAI’19/EAAI’19). AAAI Press, Article 12, 8 pages. <https://doi.org/10.1609/aaai.v33i01.330194>
- [6] Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. 2022. CARD: Classification and regression diffusion models. *arXiv preprint arXiv:2206.07275* (2022).
- [7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.
- [8] Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).
- [9] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-supervised learning for cross-domain recommendation to cold-start users. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1563–1572.
- [10] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. 2021. Variational diffusion models. *Advances in neural information processing systems* 34 (2021), 21696–21707.
- [11] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2020. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761* (2020).
- [12] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
- [13] Adit Krishnan, Mahashweta Das, Mangesh Bendre, Hao Yang, and Hari Sundaram. 2020. Transfer learning via contextual invariants for one-to-many cross-domain recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1081–1090.
- [14] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1073–1082.
- [15] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. 2022. Diffusion-LM Improves Controllable Text Generation. *arXiv preprint arXiv:2205.14217* (2022).
- [16] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*. 689–698.
- [17] Weiming Liu, Xiaolin Zheng, Jiajie Su, Mengling Hu, Yanchao Tan, and Chaochao Chen. 2022. Exploiting variational domain-invariant user embedding for partially overlapped cross domain recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 312–321.
- [18] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. *arXiv preprint arXiv:2206.00927* (2022).
- [19] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. DPM-Solver++: Fast Solver for Guided Sampling of Diffusion Probabilistic Models. arXiv:2211.01095 [cs.LG]
- [20] Muiyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019.  $\pi$ -Net: A Parallel Information-Sharing Network for Shared-Account Cross-Domain Sequential Recommendations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) (SIGIR’19). Association for Computing Machinery, New York, NY, USA, 685–694. <https://doi.org/10.1145/3331184.3331200>
- [21] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-domain recommendation: An embedding and mapping approach. In *IJCAI*, Vol. 17. 2464–2470.
- [22] Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *International Conference on Machine Learning*. PMLR, 2554–2563.
- [23] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).
- [24] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 695–704.
- [25] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, Mikhail Kudinov, and Jiansheng Wei. 2021. Diffusion-based voice conversion with fast maximum likelihood sampling scheme. *arXiv preprint arXiv:2109.13821* (2021).
- [26] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).
- [27] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*. PMLR, 8821–8831.
- [28] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
- [29] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*. Springer, 1–35.
- [30] 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*. 10684–10695.
- [31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.
- [32] Aghiles Salah, Thanh Binh Tran, and Hady Lauw. 2021. Towards Source-Aligned Variational Models for Cross-Domain Recommendation. In *Fifteenth ACM Conference on Recommender Systems*. 176–186.
- [33] Tim Salimans and Jonathan Ho. 2022. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512* (2022).
- [34] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. 2020. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 528–536.
- [35] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 650–658.
- [36] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. PMLR, 2256–2265.
- [37] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).
- [38] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020).
- [39] Quoc-Tuan Truong, Aghiles Salah, and Hady W Lauw. 2021. Bilateral variational autoencoder for collaborative filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 292–300.
- [40] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
- [41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [42] Tianxin Wang, Fuzhen Zhuang, Zhiqiang Zhang, Daixin Wang, Jun Zhou, and Qing He. 2021. Low-Dimensional Alignment for Cross-Domain Recommendation (CIKM ’21). Association for Computing Machinery, New York, NY, USA, 3508–3512. <https://doi.org/10.1145/3459637.3482137>
- [43] Wenjie Wang, Yiyang Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023. Diffusion Recommender Model. arXiv:2304.04971 [cs.IR]
- [44] Junda Wu, Zhihui Xie, Tong Yu, Handong Zhao, Ruiyi Zhang, and Shuai Li. 2022. Dynamics-Aware Adaptation for Reinforcement Learning Based Cross-Domain Interactive Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 290–300.
- [45] Tianzi Zang, Yanmin Zhu, Haobing Liu, Ruohan Zhang, and Jiadi Yu. 2021. A Survey on Cross-domain Recommendation: Taxonomies, Methods, and Future Directions. *CoRR abs/2108.03357* (2021). arXiv:2108.03357 <https://arxiv.org/abs/2108.03357>
- [46] Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. 2020. CATN: Cross-domain recommendation for cold-start users via aspect transfer network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 229–238.
- [47] Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, Mehmet Orgun, and Jia Wu. 2020. A deep framework for cross-domain and cross-system recommendations. *arXiv preprint arXiv:2009.06215* (2020).
- [48] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-domain recommendation: challenges, progress, and prospects. *arXiv preprint arXiv:2103.01696* (2021).

- [49] Yongchun Zhu, Kaikai Ge, Fuzhen Zhuang, Ruobing Xie, Dongbo Xi, Xu Zhang, Leyu Lin, and Qing He. 2021. Transfer-meta framework for cross-domain recommendation to cold-start users. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1813–1817.
- [50] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1507–1515.
- [51] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1167–1176.

## A BACKGROUND OF DPMS

DPMS define a forward process  $\{x_t | t \in [0, T], T > 0\}$  given  $x_0 \sim q(x_0)$ . Noise will be added to the data, and the distribution of  $x_t$  conditioned on  $x_{t-1}$  is defined as  $q(x_t | x_{t-1}) = \mathcal{N}(x_t | \sqrt{\alpha_t} x_{t-1}, (1 - \alpha_t)I)$ . If the noise  $1 - \alpha_t$  in each time step is small enough, the reverse process  $q'(x_{t-1} | x_t)$  can be approximated by Gaussian distribution too [7, 23]. Then we can have a model to learn this distribution by  $p_\theta(x_{t-1} | x_t) = \mathcal{N}(\mu_\theta, \Sigma_\theta)$ . If the total noise  $\{(\sqrt{\alpha_1}, (1 - \alpha_1)), (\sqrt{\alpha_2}, (1 - \alpha_2)), \dots\}$  is large enough, then the state  $x_T$  will be close to  $\mathcal{N} \sim (0, I)$ . Now we can start from  $x_T$  and use  $p_\theta$  iteratively to produce  $x_{T-1}, \dots, x_0$ .

It is possible to generate samples  $x_t \sim q(x_t | x_0)$  with arbitrary  $t$  by adding Gaussian noise to  $x_0$ , that gives

$$q(x_t | x_0) = \mathcal{N}(x_t; \bar{\alpha}(t)x_0, \sigma^2(t)I).$$

Here  $\bar{\alpha}(t)$  and  $\sigma(t)$  is noise schedule of DPM, and let them denote as  $\bar{\alpha}_t, \sigma_t$  respectively for neatness.

With these noised samples, we can train a function  $\epsilon_\theta$  to approximate the added noise, we have its loss from the work [7] as:

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{t \in [0, T], x_0 \sim q(x_0), \epsilon \sim \mathcal{N}(0, I)} [\|\epsilon - \epsilon_\theta(x_t, t)\|^2] \quad (8)$$

where  $\epsilon \sim \mathcal{N}(0, I)$ . The authors also show that  $\mu_\theta$  can be derived from  $\epsilon_\theta$  with constant  $\Sigma_\theta$ .

### A.1 Guided DPMS

With gradients from a classifier  $p_\phi(y | x_t)$ , Dhariwal et al. [4] propose a conditional DPM. They define a new noise approximate function  $\hat{\epsilon}_\theta(x_t) = \epsilon_\theta - s \cdot \Sigma_\theta \nabla_{x_t} \log p_\phi(y | x_t)$ . Here  $s$  is the gradient scale. It is claimed that a larger value of  $s$  can produce higher accurate but less diverse samples.

A recent work [8] proposes a classifier-free guided DPM, which does not require an additional classifier to provide gradient by a conditional distribution  $p_\phi(y | x_t)$ . In this work, authors replace  $\log p_\phi(y | x_t)$  by  $\epsilon_\theta(x_t | y)$  and  $\epsilon_\theta(x_t)$ , and derive the noise approximate function without an extra classifier:

$$\hat{\epsilon}_\theta(x_t | y) = \epsilon_\theta(x_t) + s \cdot (\epsilon_\theta(x_t | y) - \epsilon_\theta(x_t)) \quad (9)$$

By Bayes rule, an implicit classifier  $p_\phi^i(y | x_t) \propto \frac{p(x_t | y)}{p(x_t)}$ , and we can have  $\nabla_{x_t} \log p_\phi^i(y | x_t) \propto \nabla_{x_t} \log p(x_t | y) - \nabla_{x_t} \log p(x_t) = -\frac{1}{\Sigma_\theta} (\epsilon_\theta^*(x_t | y) - \epsilon_\theta^*(x_t))$ ,  $\epsilon_\theta^*$  is the truth score function.  $\frac{1}{\Sigma_\theta}$  can be viewed as a part of  $s$  if we set it constant.

### A.2 Sampling Acceleration

DPMs generate samples according to the reverse process  $q'(x_{t-1} | x_t)$ , thus it usually requires hundreds of steps to produce predictions.

It is time-consuming compared to other generative models with VAE and GAN frameworks, which may limit its application in high-throughput scenarios. Efficient samplers have become a central research topic in DMP.

Before introducing the fast diffusion solvers we adopt, let us give a brief about diffusion ordinary differential equation (ODE). Kingma et al. [10] prove that the stochastic differential equation (SDE)  $dx_t = f(t)x_t dt + g(t)dw_t$  has the same transition probability as  $q(x_t | x_{t-1})$ , where  $w_t$  is standard Wiener process and  $f(t) = \frac{d \log \bar{\alpha}_t}{dt}$ ,  $g^2(t) = \frac{d \sigma_t^2}{dt} - 2 \frac{d \log \bar{\alpha}_t}{dt} \sigma_t^2$ .

Song et al. [38] shows that a reverse process  $[T, 0]$  is equivalent to the forward process:  $dx_t = [f(t)x_t - g^2(t)\nabla_x \log q_t(x_t)]dt + g(t)d\bar{w}_t$ , where  $\bar{w}_t$  is standard Wiener process. They also prove the probability flow ODE is  $\frac{dx_t}{dt} = f(t)x_t - \frac{1}{2}g^2(t)\nabla_x \log q_t(x_t)$ , and they define the diffusion ODE as

$$\frac{dx_t}{dt} = f(t)x_t + \frac{g^2(t)}{2\sigma_t} \epsilon_\theta(x_t, t). \quad (10)$$

DPM-Solver [19] is a fast training-free sampling method for DPMs. It takes advantage of the semi-linear structure of diffusion ODE and can produce comparable results within only 20 function evaluations. The authors of this work observe that the RHS of diffusion ODE Equation (10) consists of a linear and a nonlinear function of  $x_t$ , and this structure is referred to as semi-linear ODE. Whereas previous works treat the two terms together, this approach treats the two parts separately, which can reduce the approximation error. Given the reverse process with  $M + 1$  time steps  $\{t_i\}_{i=0}^M$  from  $t_0 = T$  to  $t_M = 0$ , and the initial value  $\tilde{x}_{t_0} = x_T$ , they simplify the nonlinear part and define the DPM-solver-1 as  $\tilde{x}_{t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{x}_{t_{i-1}} - \sigma_{t_i} (e^{h_i} - 1) \epsilon_\theta(\tilde{x}_{t_{i-1}})$ , where  $\lambda_{t_i} = \log \left( \frac{\alpha_{t_i}}{\sigma_{t_i}} \right)$ ,  $h_i = \lambda_{t_i} - \lambda_{t_{i-1}}$ . In addition, they propose higher-order solvers that are claimed to converge with fewer inference steps.

## B ETHICAL CONSIDERATIONS

As a recommender model, our DiffCDR is able to improve user experience and satisfaction by leveraging behavioral records. However, recommender systems may also create filter bubbles that limit the diversity of content exposed to users. As a result, its application may introduce stereotypes or amplify inequalities, and lead to biased recommendation results that may bring negative influence on certain individuals or groups in society. In order to train the recommender model, the user's profile and behavioral records are collected. During this process, the user's privacy may be violated, which may harm the user's benefits, both mentally and physically.

To mitigate the potential risks and challenges in terms of ethics, related applications of recommender systems should adhere to data policies such as GDPR strictly, and give users the right to turn off the personal recommendation function.