IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 February 2024, accepted 1 April 2024, date of publication 12 April 2024, date of current version 30 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3388299

APPLIED RESEARCH

Multi-Resolution Diffusion for Privacy-Sensitive Recommender Systems

DEREK LILIENTHAL, PAUL MELLO, MAGDALINI EIRINAKI, (Member, IEEE), AND STAS TIOMKIN, (Member, IEEE)

Department of Computer Engineering, San Jose State University, San Jose, CA 95192, USA

Corresponding author: Magdalini Eirinaki (magdalini.eirinaki@sjsu.edu)

This work was supported in part by the SJSU College of Engineering Davidson Student Scholar Award.

ABSTRACT While recommender systems have become an integral component of the Web experience, their heavy reliance on user data raises privacy and security concerns. Substituting user data with synthetic data can address these concerns, but accurately replicating these real-world datasets has been a notoriously challenging problem. Recent advancements in generative AI have demonstrated the impressive capabilities of diffusion models in generating realistic data across various domains. In this work we introduce a Score-based Diffusion Recommendation Module (SDRM), which captures the intricate patterns of real-world datasets required for training highly accurate recommender systems. SDRM allows for the generation of synthetic data that can replace existing datasets to preserve user privacy, or augment existing datasets to address excessive data sparsity. Our method outperforms competing baselines such as generative adversarial networks, variational autoencoders, and recently proposed diffusion models in synthesizing various datasets to replace or augment the original data by an average improvement of 4.30% in Recall@k and 4.65% in NDCG@k.

INDEX TERMS Data privacy, diffusion models, machine learning, recommender systems, synthetic data.

I. INTRODUCTION

Recommender systems have become a ubiquitous part of the Web experience, with applications in multiple domains. Such systems are trained on user data that come in the form of explicit ratings or implicitly inferred preferences for items as well as various other sources of information including demographics, item metadata, context data, social network connections, etc. [1]. Inevitably, this heavy reliance on user data makes privacy one of the major challenges for recommender systems [2], [3]. Breaches of user privacy and even violations of rights may occur in various stages of a recommender system's life cycle, including inferences that can be made from the output of the recommendation process [2], [3], [4]. Research into privacy-preserving algorithms and anonymization of user data has made significant progress, but a balance between accurately capturing complex user preferences and data privacy remains challenging [5]. Even when data privacy is not the primary concern, there are other needs which can be addressed by synthetic datasets for recommender systems, such as data sparsity [6]. A dataset is considered sparse when a very small percentage of total user-item preferences are available. In cases where a high number of missing values exists, a synthetic dataset can be used to augment the original dataset by increasing the total number of data points, ideally improving the prediction accuracy of the recommender system.

Generative AI has become increasingly popular due to its ability to generate realistic content across various modalities [7], [8]. Generative models such as Generative Adversarial Networks (GANs) [9], Variational Autoencoders (VAEs) [10], and Diffusion models (DMs) [11] have been shown to model complex data distributions extremely well, and have thus been employed in various domains, including recommender systems. Most prior work employs GANs or VAEs as part of the collaborative filtering process (i.e. to model user-item interactions) [12], [13], [14], [15], [16], [17], [18], with the objective of improving predictions rather than generating synthetic data. DMs, a recent addition to the generative modeling family, have emerged as a prominent framework for data synthesis and have been shown to produce high-quality samples, outperforming GAN

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

architectures [19]. Recently, DMs have been applied to recommendation systems for collaborative filtering [20], [21] and sequential predictions [22], [23] with very promising results.

In this work, we propose a Score-based Diffusion Recommendation Module (SDRM) that allows the generation of synthetic data in the form of user-item interactions. SDRM utilizes a VAE encoding scheme to map user-item preferences to a latent Gaussian distribution, where a diffusion model transforms the Gaussian input and reconstructs it using the VAE decoder. The training of our encoder, decoder, and diffusion model are conducted separately, which allows us to leverage the VAE's variational inference capabilities. Through this architecture, we can generate arbitrarily many synthetic samples that accurately capture the users' preference distribution allowing for a partial or total replacement of the user data.

This approach results in a module that can produce diverse data while preserving the complex distributions of user-item interactions. Through thorough experimental evaluation, our approach has been proven to be more effective than other methods that utilize GANs [24], VAEs [13], [24], and DMs [20], [21] in creating recommendation datasets of varying sizes and densities. We demonstrate SDRM's ability to generate accurate synthetic recommendation data across four datasets and three collaborative filtering recommender algorithms, showcasing the performance increase when using SDRM compared to the original data and other data generation techniques.

Our contributions are summarized as follows:[1]

- A score-based diffusion module (SDRM) for generating high quality synthetic data from user-item preference datasets.
- A novel method to boost the accuracy of a collaborative filtering algorithm through variational inference with a score-based diffusion model and variational autoencoder.
- The introduction of multi-resolution sampling in the diffusion process to further increase the accuracy of synthetic data-based recommendations.
- An extensive experimental evaluation across four datasets and three recommendation algorithms demonstrates increased accuracy by an average improvement of 6.81% in Recall and 7.73% in NDCG for augmented datasets, and 1.42% in Recall and 1.98% in NDCG for synthetic datasets—over competing methods. This is achieved while preserving privacy by attaining 99% dissimilarity to the original data and maintaining similar user and item distributions.

To the best of our knowledge, this is the first work that employs diffusion models with the primary objective to generate synthetic datasets for recommender systems.

The rest of this paper is organized as follows: Section II provides an overview of related work in privacy methods for recommender systems, synthetic data generation using generative models, and generative modeling for recommender systems. Section III provides a background on the core materials used in our approach, namely VAEs and DMs. In Section IV, we present SDRM, detailing our architectural choices, including the proposed objective function and the sampling and training methodologies. Section V presents the setup and results of our experimental evaluation across a range of datasets and evaluation metrics. Finally, in Section VI, we summarize our results, discuss our module's limitations, and outline our plans for future work.

II. RELATED WORK

In this section, we review related work and how it relates to ours from three different angles: privacy in recommender systems, synthetic data generation, and generative models for recommender systems.

A. PRIVACY IN RECOMMENDATIONS

Traditional methods of privatizing data involved removing personally identified information from the datasets, therefore anonymizing the data. However, such methods have been shown to be ineffective as large-scale datasets could easily be de-anonymized [25]. This led to the formulation of differential privacy, which mathematically guarantees that the data or a model trained with differentially private data is private [26].

There exist several works that focus on differentially private recommender systems employing matrix factorization [27] or federated learning approaches to decentralize a users' data from a single location [28]. While differential privacy methods provide a mathematical guarantee the data is private, it incurs a decrease in accuracy when compared to training the model with the original dataset [29]. To counteract this loss in performance, prior works have been proposed that focus on synthesizing data using 'privacy-preserving' techniques such as Liu et al.'s UPC-SDG [5] and Slokom et al.'s CART-based methods [30]. While such techniques do not mathematically guarantee private data, they aim to enhance user privacy without compromising accuracy. UPC-SDG does so through an attention-based module substituting user preferences, while Slokom et al. use partially generated data for privacy. While our objective is the same, in our approach we focus on augmenting or replacing original datasets with synthetic data using a score-based diffusion method for recommender algorithm training.

B. SYNTHETIC DATA GENERATION

Many prior works using diffusion have focused on improving image classification through augmentation. These methods have focused on generating more training examples to improve the performance of downstream tasks, like building

[1] We have made our code available at https://github.com/Multi-resolution-diffusion-recommender/SDRM.

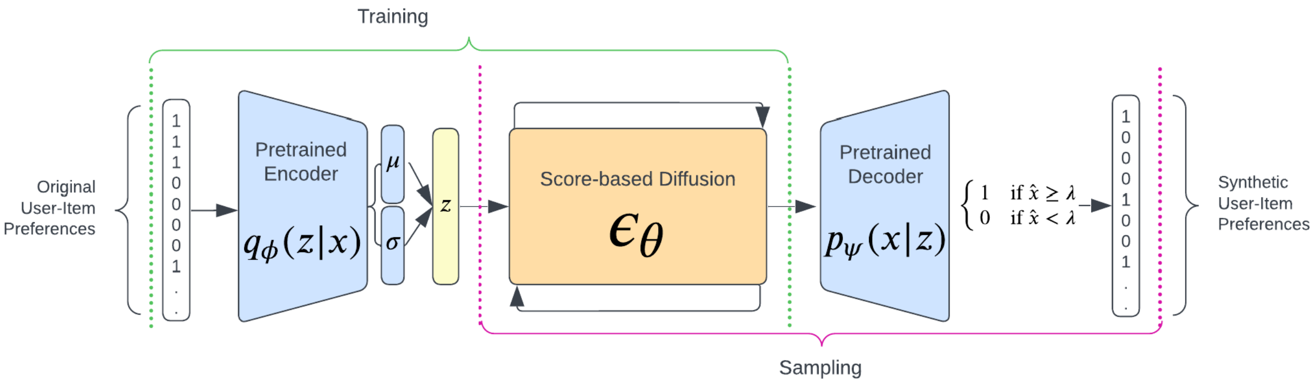

FIGURE 1. **SDRM Training and Sampling.**

more robust classifiers [31], [32] or improving image segmenting for healthcare applications [33].

The need for fully synthetic copies of datasets has been driven by the need of companies to stay compliant with data protection regulations [34], such as the GDPR (General Data Protection Regulation) [35]. Many works have been proposed to create data generators whose objective consists of synthesizing tabular datasets which hold the same statistical properties as the original, but are not identical copies [24], [36], [37]. Both CTGAN [24] and CTAB-GAN+ [37] use Generative Adversarial Networks, while for TabDDPM [11] the authors use a Denoising Diffusion Probabilistic Model (DDPM) [11] to model tabular datasets. However, none of these prior works have benchmarked their models on recommender system datasets, leaving a gap in the current research for examining the generalizability of tabular data generators to model recommendation datasets.

Our method, SDRM, utilizes a diffusion process similar to TabDDPM for modeling recommendation data, which could be broadly categorized as tabular. However, SDRM distinguishes itself by compressing categorical features (user-items preferences) into lower-dimensional representations via a Variational Autoencoder to address the high dimensionality of recommendation data and employs multi-resolution sampling for new data generation. In contrast, TabDDPM's approach of concatenating each categorical feature as the input into the denoising MLP (Multilayer Perceptron) network significantly increases the computational complexity due to the thousands of unique categories (items) in recommendation datasets. TabDDPM's complexity made direct comparisons between SDRM infeasible in our evaluation in section V.

C. GENERATIVE MODELS IN RECOMMENDATIONS

Recently, generative models have been used in recommender systems, with most of the prior work involving GANs and VAEs. GAN-based techniques leverage adversarial training to optimize the generator's ability to augment user interactions [12], [38]. The application of GANs to enhance user-item data for training recommender models has gained significant attention, with methods emerging to augment user preferences for collaborative filtering [12], [15], [16], and sequential predictions [39], [40]. Various works on incorporating GANs and reinforcement learning to simulate user behavior have also been explored [41], [42].

Many works have been proposed to leverage VAEs in recommendation systems to predict user ratings [13], [14], [17], [18] by using their ability to compress data into a Gaussian distribution using the re-parameterization trick [10] and applying variational inference to reconstruct the original data. To take advantage of VAE's abilities to reconstruct data from a Gaussian distribution, we propose pretraining a VAE to map data to a low-dimensional Gaussian representation for which we apply diffusion on their latent space. Similar work has been proposed using this method [21], [43], but trained the VAE and diffusion model together, whereas we train them individually to take advantage of the variational inference ability of VAEs.

There have been very few works incorporating diffusion into recommendation systems. Existing research primarily focuses on predicting complete user preferences [20], [21] or making sequential predictions [22], [23]. Walker et al. proposed CODIGEM [20], a DDPM-inspired architecture that stacks autoencoders to denoise the corrupted user-item preferences. Very recently, Wang et al. proposed DiffRec [21], a model that trains a denoising diffusion model by incorporating an array of MultiVAE encoders to map user data to a low-dimensional space and reconstructs the input back to its original high-dimensional form. While DiffRec and CODIGEM show promise as methods to predict user ratings, they are not suitable as data generators and do not take full advantage of diffusion's ability to generate realistic data. We make this claim for a few reasons: first, both DiffRec's and CODIGEM's training approach implements early stopping, which prevents the denoising model from fully learning the entire resolution of the latent data during the denoising process [11]; and second, CODIGEM and DiffRec train for 3 and 10 timesteps T respectively, while training for larger T (such as 1000 steps or more), has been shown to capture more detailed information, due to increased reconstruction time to learn imperceptible details of the data, as demonstrated in [11] and [44].

In contrast to DiffRec and CODIGEM, our model, SDRM, benefits from a simpler architecture and trains over a larger amount of T steps without early stopping and utilizes an ablated training objective which focuses on generating user-item interaction data for enhancing, or completely replacing the original datasets. Additionally, we specifically designed SDRM to create synthetic user preferences that closely follow the intricate distribution of user-item preference data. This design enhances the performance of out-of-the-box recommender models while preserving user privacy, rather than serving as a direct predictor model.

III. BACKGROUND

A. VARIATIONAL AUTOENCODERS

Variational Autoencoders (VAEs) are a probabilistic generative model that aims to learn a joint distribution between the data and a compressed latent distribution, $p_\psi(x, z)$. VAEs use a stochastic encoder network $q_\phi(z|x)$, to encode data to a lower-dimensional latent space z, then reconstruct it using a stochastic decoder network $p_\psi(x|z)$ using variational inference to approximate the probabilistic latent variables. The latent distributions are typically a normal Gaussian, characterized through its mean μ and variance σ.

VAEs learn the stochastic mappings between the observed data and the latent variables through the *evidence lower bound* (ELBO) optimization objective. This objective aims to approximate the log-likelihood of the observed data, which is intractable due to the integral of the marginal likelihood, $p_\psi = \int p_\psi(x, z)dz$, lacking an analytical solution [45]. The ELBO objective is defined as follows:

$$L(x; \psi, \phi) \equiv \mathbb{E}_{q_\phi(z|x)}\left[\log p_\psi(x|z)\right] - KL\left(q_\phi(z|x) \parallel p(z)\right) \quad (1)$$

We can obtain an unbiased estimation of the ELBO by sampling the latent z space $z \sim q_\phi$ and optimizing it through gradient ascent. However, to take gradients with respect to ϕ in this sampling process, we must employ the *reparameterization trick*: $z = \mu_\phi(x) + \epsilon \odot \sigma_\phi(x)$ [10], where $\epsilon \sim \mathcal{N}(0, I_K)$ and K represents the size of the latent space. This trick introduces stochasticity into the sampling procedure, allowing gradients with respect to ϕ to be back-propagated through the latent z space.

Using the reparameterization trick, we can optimize the encoder network parameters, ϕ, and decoder network parameters, ψ, using backpropagation while maintaining the stochasticity needed to model an unbiased latent distribution (such as Gaussian) during the training process. To create new data samples, we need to sample some random noise from a normal Gaussian distribution $\epsilon \sim \mathcal{N}(0, I_K)$ and pass it to the decoder network $p_\psi(x|\epsilon)$ to obtain a new data sample $\hat{x}$ which follows the approximate distribution of an original data sample x.

SDRM leverages the abilities of the decoder network of a VAE to transform Gaussian noise into data samples that approximate the original data distribution. We then model these VAE latent z distributions using a score-based diffusion model. Parameterizing this latent distribution using score-based diffusion allows us to take advantage of diffusion models ability to generate high-quality data samples, which in this case is Gaussian noise at different resolutions, and the variational inference capabilities of VAEs to generate new samples from a lower-dimensional representation.

B. DIFFUSION MODELS

Diffusion Models (DMs) have emerged as a versatile class of generative models. DMs are probability density functions which consist of two processes, a *forward* and a *reverse* [46]. In the forward process, noise perturbations are incrementally added to the initial input x_0, often in the form of a standard normal Gaussian $x_T \sim N(0, \mathbf{I})$, using a noise scheduler β_t for T time steps. This forward process is a conditional distribution x_t given x_{t-1} and can be defined as follows:

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \quad (2)$$

In the reverse process, a neural network θ learns to predict the next state from the prior state by gradually denoising an input from x_T to x_0. Here, $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ defines the mean location and information densities respectively.

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad (3)$$

The reverse process parameters, θ, are maximized through an evidence lower bound (ELBO) objective on the original data distribution [11]. The ELBO in DM*s* can be represented by the mean square error between the original data, corrupted by the forward process, and the reconstruction data, generated by the reverse process. This enables the efficient training of DMs and high-quality data synthesis [11], [47], [48], [49].

Generative models, such as diffusion, can be trained utilizing various training objectives [50], [51], including a score-based approach [47], [52], [53], [54], [55], [56], [57], [58], [59], [60]. Compared to the ELBO, the unbiased nature of score-based objectives naturally lends itself to recommender systems. Score-based objective functions, like score matching, are defined by taking the gradient of the log likelihood of x and minimizing the expected squared error of the score based predictions.

$$\mathbb{E}_{p(x)}\left[\|s_\theta(x) - \nabla \log p(x)\|^2\right] \quad (4)$$

Score matching objectives can suffer from various difficulties including error estimation. When data is sparse, they may not be able to accurately estimate the score function. Simultaneously, access to ground truth data is ideal to accurately capture the score. Our approach avoids these problems by utilizing a VAE to increase the data density and modifies the score-based objective function by introducing noise through $\tilde{x} = x + \epsilon$ and transforming the function into a denoising problem. We leverage this score-based objective in our work to capture the intricacies of user-item interactions.

IV. SCORE-BASED DIFFUSION RECOMMENDER MODULE

We propose SDRM (Score-based Diffusion Recommender Module), a module designed to generate artificial user-item preferences to augment or replace the dataset it is trained on.

Diffusion models are capable of modeling complex distributions of arbitrarily many dimensions, but suffer from intractability when datasets have sparse features. As a result, reducing the overall dimensions before the diffusion process can help with modeling accurate reconstructions over all possible items of the dataset. For this reason, our proposed method employs a VAE to map data into a lower-dimensional Gaussian distribution and enables more efficient training for SDRM.

Generating recommendation data poses a unique problem due to the challenge of training a model on datasets containing tens or hundreds of thousands of user preferences. Recommendation data is highly sparse and skewed [1], making it necessary to reduce the overall dimensions before the diffusion process. Through this method, diffusion can learn to reconstruct all possible items of the dataset. For this reason, our proposed method takes advantage of VAE's ability to map data into a lower-dimensional Gaussian distribution and enables more efficient training for SDRM.

As shown in Figure 1, SDRM uses a pre-trained encoder to map user ratings into a smaller representation, followed by a Multi-Layer Perceptron (MLP) denoising model, and finally, a decoder to decompress the latent vectors into user ratings. We train SDRM in two steps: first, we pretrain a multinomial likelihood variational autoencoder (MultiVAE) on user ratings, and then we train SDRM with the encoder of the MultiVAE. To generate user ratings, we sample from a normal Gaussian distribution, denoise it using SDRM, and pass the denoised latent vectors into the VAE decoder to reconstruct the data into the same dimensions as the original dataset.

We propose two sampling variations for SDRM: full-resolution sampling (F-SDRM), and multi-resolution sampling (M-SDRM). F-SDRM denoises pure Gaussian noise starting at timestep $T \rightarrow 0$ while M-SDRM starts at a random timestep $t \in T$ and denoises to 0. Because the latent z vector of the pretrained encoder maps to a Gaussian distribution, SDRM learns to denoise from a Gaussian distribution z to a Gaussian distribution $\hat{z}$. By starting at a random timestep t instead of the full timestep T and denoising until we reach timestep 0, we reconstruct Gaussian noise at different resolutions into user data for M-SDRM.

A. SDRM OBJECTIVE

We propose the following score-based objective for training diffusion in recommender systems.

$$\ell(z_t) = \frac{||\big(s_\theta(\hat{z}_t) - s_\theta(z_t)\big) - \Delta_\theta(z_t)||_2^2 + ||s_\theta(z_t) - \Delta_\theta(z_t)||_2^2}{||\Delta_\theta(z_t)||_2^2} \tag{5}$$

where t is the discrete time step, uniformly sampled from the range $[1, \ldots, T]$; z_t is the corresponding latent representation of the input data point (e.g., user-item interactions) encoded by the VAE; $\hat{z}_t = z_t + \nu$ with $\nu \sim \mathcal{N}(0, \sigma_\nu \mathbf{I})$ is its perturbed version by the Gaussian noise, η; $s_\theta(\cdot)$ is the score function in the diffusion model with the parameters θ; and $\epsilon_\theta(z_t)$ is the prediction noise at the time t in the diffusion model. Finally, $\Delta_\theta(z_t) \doteq \epsilon_\theta(z_t) - z_t$ quantifies the difference between the prediction noise, $\epsilon_\theta(z_t)$, and the latent representation, z_t. The denominator, $||\Delta_\theta(z_t)||_2^2$, serves as a normalizing factor to stabilize the loss function. We show, through experimentation on various datasets, that this score-based objective is more effective than standard training objectives applied in other diffusion models, such as CODIGEM [20] and DiffRec [21].

B. SDRM TRAINING AND SAMPLING

We train MultiVAE [13] on the original dataset, using the evidence lower-bound objective defined by

$$\begin{aligned}\mathcal{L}_u(\psi, \phi) = \mathbb{E}_{q_\phi(z_u|x_u)}[\log p_\psi(x_u|z_u)] \\ - \beta \cdot KL(q_\phi(z_u|x_u) \| p(z_u))\end{aligned} \tag{6}$$

where $\log p_\psi(x_u|z_u)$ is the Gaussian log-likelihood for user u, q_ϕ is the approximating variational distribution (inference model), and β is a heuristic annealing parameter. Next, we utilize the MultiVAE encoder to compress user-item preferences into a smaller latent representation for training SDRM with the loss in Equation (5). We prevent gradient updates on the VAE model while training SDRM, ensuring SDRM purely learns to transform the VAE latent variable z to another latent variable $\hat{z}$.

For sampling data with F-SDRM, we follow the same sampling procedure as Ho et al. [11] and map z_0 to the decoder of the MultiVAE. We follow the same process for M-SDRM, but start the denoising from a random $t \in T$ for each sample. Additional details of the training and sampling procedure are included in Algorithms 1 and 2.

Algorithm 1 Training SDRM

```
Sample candidate hyperparameters $x^*$ from $\chi$ and initialize SDRM
while Recall@10 improves do
    for batch size $\in x^*$ do
        // Train VAE
        $\mathcal{L}_u(\psi, \phi) = \mathbb{E}_{q_\phi(z_u|x_u)}[\log p_\psi(x_u|z_u)] - \beta \cdot KL(q_\phi(z_u|x_u) \| p(z_u))$
    end for
end while
for epochs $\in x^*$ do
    // Map user-item preferences to z
    $z_u \leftarrow q_\phi(z_u|x_u)$
    // Train SDRM
    $\ell(z_t) = \frac{||\big(s_\theta(\hat{z}_t) - s_\theta(z_t)\big) - \Delta_\theta(z_t)||_2^2 + ||s_\theta(z_t) - \Delta_\theta(z_t)||_2^2}{||\Delta_\theta(z_t)||_2^2}$
end for
```

Algorithm 2 Sampling SDRM

```
Select $t$ number of denoising timestep from $[1, T]$
Initialize synthetic user-item preference vector with sampled Gaussian noise $z_T \sim N(0, I)$
for $T, \ldots, 1$ do
    // Sample Gaussian noise
    $z \sim N(0, I)$ if $t > 1$, else $z = 0$
    // Transform Gaussian noise to Gaussian noise
    $\hat{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\hat{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}}}\epsilon_\theta(\hat{z}_t, t)\right) + \sigma_t z$
end for
// Reconstruct $\hat{z}$ using VAE decoder to $\hat{x}$
$\hat{x} \leftarrow p_\psi(x|\hat{z})$
// Apply piecewise conditional to $\hat{x}$ with threshold $\lambda$
$f(\hat{x}) = \begin{cases} 1 & \text{if } \hat{x} \geq \lambda \\ 0 & \text{if } \hat{x} < \lambda \end{cases}$
```

V. EVALUATION

We evaluate SDRM by comparing how well the generated user-item data can augment or replace the original data when used as input for any recommendation algorithm. For this purpose, we demonstrate the ability of SDRM to synthesize recommendation datasets by employing three recommendation algorithms, namely SVD [61], MLP [62], and NeuMF [63], and evaluate the generated top-k recommendations in terms of Recall@k and ranking order, using NDCG@k. We compare our model's two variations, namely F-SDRM and M-SDRM, with six generative model baselines on four diverse datasets.

A. EXPERIMENT SETTING

1) DATASETS

We employ four publicly available and commonly used datasets to evaluate SDRM: Amazon Luxury Beauty (ALB) [64], Amazon Digital Music (ADM) [64], MovieLens 100k (ML-100k) [65], and MovieLens 1M (ML-1M) [65]. We selected these datasets because they encompass a wide range of items and vary in user/item/rating cardinality and sparsity, as shown in Table 1. Moreover, their relatively small size allowed us to conduct extensive, lengthy optimization runs. Each dataset is pre-processed by converting ratings greater than three to 1 and ratings three or less to 0. We also filtered out low-frequency users and items, so only users who rated at least five items and items with at least five ratings are included in the datasets.

Each original dataset was split into a training, testing, and validation set using a 70:20:10 split on all the users. Additionally, a small portion of user ratings in the testing set are needed as ground truth users in the training dataset to evaluate the effectiveness of synthetic data. The ratings of those users who are in the training and test set are masked during the evaluation phase.

TABLE 1. Dataset Statistics.

Dataset	#Users	#Items	#Ratings	Sparsity
Amazon Luxury Beauty	1,344	729	15,359	98.43%
Amazon Digital Music	10,621	8,582	108,509	99.88%
MovieLens 100k	938	1,008	95,215	89.93%
MovieLens 1M	3,125	6,034	994,338	94.73%

2) BASELINE GENERATIVE MODELS

We evaluate the ability of SDRM to generate synthetic data against the original dataset (representing the baseline before augmenting or substituting data), and the following generative models:

- CTGAN [24] (Conditional Tabular Generative Adversarial Network) is a GAN designed for synthesizing tabular data.
- TVAE [24] is a conditional Tabular Variational Auto-Encoder designed for synthesizing tabular data, similar to CTGAN.
- CODIGEM [20] is a diffusion-inspired collaborative filtering model that utilizes stacking denoising autoencoders during the reverse denoising process.
- MultiVAE [13] is a Variational Autoencoder that uses a multinomial likelihood loss objective and variational inference to model implicit user data.
- MultiVAE++ is the pre-trained multinomial variational autoencoder used in SDRM, but its hyperparameters are optimized as shown in Table 2. This model represents a true baseline to compare the increased performance of SDRM.
- DiffRec [21] is a denoising diffusion probabilistic model used for predicting whole user-item interactions.

3) BASELINE RECOMMENDER ALGORITHMS

We evaluate the effectiveness of the synthetic data using three state-of-the-art recommendation algorithms: SVD (Singular Value Decomposition) [61], a matrix factorization technique that approximates the original matrix with lower-dimensional matrices representing latent factors; MLP (Multilayer Perceptron) [62], a deep neural network architecture incorporating user and item embeddings which feed into multiple feed-forward layers to model and predict whole user-item preferences; and NeuMF [63], that fuses matrix factorization and an MLP network to predict single user-item preferences.

4) EVALUATION METRICS

Similar to related work, we adopt two top-k metrics, namely Recall@k and NDCG@k. Recall@k is defined as the percentage of ground-truth positives (items) that the system recommends as positive for a recommendation list of size k. The formula for Recall@k is given as follows:

$$\text{Recall@}k = \frac{\text{Total Number of Relevant Items}}{\text{Number of Relevant Items found in top-}k} \tag{7}$$

NDCG@k (Normalized Discounted Cumulative Gain) is defined as the ratio of the cumulative gain to its ideal value, which measures how close the recommended top-k ranking is to the ground truth. The formula for calculating NDCG at a specific rank position k involves first computing the

DCG (Discounted Cumulative Gain) at k, then normalizing that score by the IDCG (Ideal Discounted Cumulative Gain), which represents the maximum possible DCG at k if the results were perfectly ranked. The formula is given as follows:

$$\mathrm{DCG}@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{8}$$

$$\mathrm{IDCG}@k = \sum_{i=1}^{|REL_k|} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{9}$$

$$\mathrm{NDCG}@k = \frac{\mathrm{DCG@k}}{\mathrm{IDCG@k}} \tag{10}$$

where rel_i is the relevance score of the result at position i and $|$ REL$_k$ $|$ is the list of relevance scores sorted in descending order up to position k.

5) GENERATING SYNTHETIC DATA

Generating data from CTGAN and TVAE was straightforward, as both approaches use a Gumbel-softmax [66] to directly create categorical features. To generate data from CODIGEM, MultiVAE, MultiVAE++, and DiffRec, we first trained those models, then modified the code base to inject Gaussian noise into the decoder of MultiVAE and denoised Gaussian noise with CODIGEM and DiffRec. This process allows each baseline model to generate synthetic samples starting from Gaussian noise.

For each synthetic dataset, we generated an equal number of samples to match the number of users in each training split of the original dataset. We utilized a conditional piece-wise function to transform the logit outputs from the generative baseline models and SDRM. We used a threshold value λ on the synthetic data, which matches the value at the sparsity quantile of each dataset.

$$f(x) = \begin{cases} 1 & \text{if } x \geq \lambda \\ 0 & \text{if } x < \lambda \end{cases} \tag{11}$$

6) HYPERPARAMETER SEARCH FOR SDRM

During the training procedure, we aim to find the set of hyperparameters x^* that yields the optimal $f(x)$ where a hyperparameter x can take any value in the search space of χ [67]. We formalize the search as $x^* = \mathrm{argmax}_{x \epsilon \chi} f(x)$, where we optimize for Recall@10. We utilize a TPE (Tree-structured Parzen Estimator) [68] and a successive halving pruner [69] to search over the search space χ efficiently. We perform 5-fold cross-validation for each set of hyperparameters used to train SDRM and report the average Recall and NDCG results. The hyperparameter search space used during SDRM training is shown in Table 2.

To find the optimal set of hyperparameters for SDRM, we employed Bayesian optimization search over the search space mentioned in Table 2 using the Optuna framework [70]. Due to the observed variance in performance between runs with the same hyperparameters, we ran each set of candidate hyperparameters five times and averaged the results presented in Tables 3 and 4. We ran each evaluation model (SVD, MLP, NeuMF) with the benchmark datasets and SDRM between 30 to 300 trials. We computed these results using RTX 3080, RTX 4090, T4, V100, and A10 GPUs over several weeks.

TABLE 2. Hyperparameter search space χ for SDRM.

Hyperparameters	Search Space
SDRM epochs	{5, 501, step=5}
SDRM learning rate	{1.e-6, 1.e-4, step=1.e-6}
Gaussian variance	{0.01, 1, step=0.1}
SDRM timesteps	{3, 200, step=5}
SDRM batch size	{30, 1000, step=10}
MLP hidden layers	{0, 5, step=1}
MLP latent neurons	{50, 1000, step=50}
MLP hidden layers	{0, 5, step=1}
VAE hidden layer neurons	{MLP latent neurons, 1000, step=50}
VAE learning rate:	{1.e-4, 1.e-2, step=1.e-2}
VAE batch size	{30, 1000, step=10}

B. TRAINING PROCEDURE

We showcase the ability of diffusion to generate realistic user-item preferences in two conditions: when the synthetic data is used to augment the original training set, and when the synthetic data is used to replace the original training set. For augmenting data, we combine the synthetic data with the training portion of the original dataset and evaluate the results on the holdout ratings from the test set. Similarly, for replacing the original data with synthetic, we train each baseline recommender algorithm with the synthetic data combined with 20% of ground truth user ratings from the test set. This approach is necessary to show how synthetic data can enhance the predictive performance on real users, as it is impractical to measure the effectiveness of synthetic data in a recommendation model without incorporating some actual user data. During the evaluation process, we predict all possible items for each user in the test set and mask any items rated during training to prevent data leakage in our results.

Each generative model was trained five times on each dataset and a synthetic dataset was generated matching the same number of users in its respective training set. Subsequently, each baseline recommender model was trained with and without the original data. Similarly, SDRM was trained five times and each run generated a synthetic dataset for the MultiVAE++ and SDRM using full (F-SDRM) and multi-resolution (M-SDRM) sampling. This way, every MultiVAE++, F-SDRM, and M-SDRM result was calculated from the same exact runs. We generate top-k recommendations for each user in the validation set and compare them to the ground truth using Recall@k and NDCG@k.

To account for the negative sampling needed for training NeuMF, we utilized items with the lowest ratings from each user in the synthetic datasets. For the actual users, we selected the items with a 0 rating.

TABLE 3. Overall performance comparison between baselines by training with synthetic and the original dataset. The best results are in bold and the second best are underlined. Average overall improvement: Recall 4.48%, NDCG 5.07%.

Model	SVD							
Dataset	ALB		ML-100k		ML-1M		ADM	
Baseline	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10
Original	0.3113 ± 0	0.287 ± 0	0.3716 ± 0	0.3973 ± 0	**0.3769** ± 0	**0.4035** ± 0	0.0624 ± 0	0.0427 ± 0
CTGAN	0.2642 ± 0.018	0.2407 ± 0.014	0.3533 ± 0.006	0.3813 ± 0.004	0.3652 ± 0.003	0.3917 ± 0.004	0.0405 ± 0.002	0.0267 ± 0.001
TVAE	0.3113 ± 0	0.2873 ± 0	0.3668 ± 0.003	0.3918 ± 0.003	0.3638 ± 0.004	0.3869 ± 0.003	0.0624 ± 0	0.0427 ± 0
CODIGEM	0.3025 ± 0.003	0.2819 ± 0.002	0.3492 ± 0.006	0.3745 ± 0.01	0.3113 ± 0.004	0.3336 ± 0.004	0.049 ± 0.003	0.0301 ± 0.001
DiffRec	0.2936 ± 0.003	0.2748 ± 0.003	0.37 ± 0.008	0.4021 ± 0.011	0.3598 ± 0.007	0.383 ± 0.008	0.0306 ± 0.002	0.0228 ± 0.003
MultiVAE	0.3226 ± 0.007	**0.2998** ± 0.007	0.3822 ± 0.006	0.4103 ± 0.007	0.3391 ± 0.006	0.3658 ± 0.007	0.0612 ± 0.004	0.0389 ± 0.002
MultiVAE++	0.3163 ± 0.014	0.2912 ± 0.007	0.3878 ± 0.006	0.4126 ± 0.007	0.3717 ± 0.002	0.3946 ± 0.003	0.0613 ± 0.003	0.0406 ± 0.002
F-SDRM	**0.325** ± 0.012	0.2988 ± 0.01	0.3924 ± 0.005	**0.4181** ± 0.005	0.3722 ± 0.002	0.3977 ± 0.002	0.0623 ± 0.003	0.041 ± 0.001
M-SDRM	0.3249 ± 0.004	0.2957 ± 0.005	**0.3971** ± 0.004	0.417 ± 0.004	0.372 ± 0.002	0.3962 ± 0.003	**0.0641** ± 0.004	**0.0442** ± 0.002
Improvement	**0.74 %**	**-0.33 %**	**2.39 %**	**1.33 %**	**-1.26 %**	**-1.44 %**	**2.72 %**	**3.51 %**
Model	MLP							
Original	0.3183 ± 0.006	0.3007 ± 0.004	0.3569 ± 0.007	0.3788 ± 0.004	0.3319 ± 0.006	0.3558 ± 0.005	0.0194 ± 0.001	0.0128 ± 0.001
CTGAN	0.3188 ± 0.013	0.2987 ± 0.005	0.3567 ± 0.007	0.3781 ± 0.006	0.3228 ± 0.005	0.3463 ± 0.005	0.0208 ± 0.004	0.0135 ± 0.002
TVAE	0.2438 ± 0.002	0.2431 ± 0.001	0.3437 ± 0.012	0.3622 ± 0.014	0.2949 ± 0.019	0.3186 ± 0.02	0.0184 ± 0.001	0.012 ± 0.001
CODIGEM	**0.3365** ± 0.014	**0.3071** ± 0.008	0.3479 ± 0.01	0.3686 ± 0.013	0.3132 ± 0.007	0.3377 ± 0.006	0.0596 ± 0.003	0.0385 ± 0.002
DiffRec	0.3283 ± 0.012	0.3013 ± 0.004	0.355 ± 0.011	0.3769 ± 0.008	0.3246 ± 0.001	0.3482 ± 0.002	0.0213 ± 0.002	0.0139 ± 0.001
MultiVAE	0.3328 ± 0.009	0.3048 ± 0.006	0.053 ± 0.014	0.0615 ± 0.016	0.3588 ± 0.003	0.372 ± 0.006	0.0199 ± 0.003	0.0137 ± 0.001
MultiVAE++	0.3316 ± 0.008	0.3 ± 0.007	0.3901 ± 0.007	0.4101 ± 0.009	0.3528 ± 0.005	0.3763 ± 0.006	0.0756 ± 0.01	0.0489 ± 0.007
F-SDRM	0.3246 ± 0.008	0.3004 ± 0.004	0.3839 ± 0.013	0.4055 ± 0.014	**0.3595** ± 0.002	**0.3845** ± 0.001	0.0146 ± 0.01	0.0095 ± 0.008
M-SDRM	0.3343 ± 0.013	0.303 ± 0.007	**0.3947** ± 0.011	**0.419** ± 0.01	0.3591 ± 0.003	0.384 ± 0.003	**0.0798** ± 0.0012	**0.0523** ± 0.002
Improvement	**-3.66 %**	**-1.35 %**	**1.18 %**	**2.17 %**	**0.19 %**	**2.18 %**	**5.55 %**	**6.95 %**
Model	NeuMF							
Original	0.2399 ± 0.009	0.0818 ± 0.012	0.1006 ± 0.009	0.096 ± 0.005	0.0113 ± 0.001	0.0094 ± 0.001	0.0057 ± 0.001	0.0011 ± 0.001
CTGAN	0.2444 ± 0.002	0.0877 ± 0.001	0.0889 ± 0.012	0.086 ± 0.013	0.0219 ± 0.003	0.0192 ± 0.004	0.007 ± 0.002	0.0014 ± 0.001
TVAE	0.2439 ± 0.001	0.0874 ± 0.001	0.1258 ± 0.011	0.1282 ± 0.006	0.0126 ± 0.004	0.0105 ± 0.004	0.0053 ± 0.001	0.0013 ± 0.001
CODIGEM	0.1709 ± 0.116	0.0477 ± 0.035	0.1015 ± 0.016	0.0839 ± 0.016	0.0642 ± 0.001	0.0601 ± 0.002	0.007 ± 0.002	0.0014 ± 0.001
DiffRec	0.2621 ± 0.006	0.0923 ± 0.004	0.1147 ± 0.013	0.1093 ± 0.016	0.0203 ± 0.005	0.0178 ± 0.005	0.0138 ± 0.002	0.0035 ± 0.001
MultiVAE	0.2475 ± 0.057	0.0826 ± 0.026	0.1253 ± 0.012	0.118 ± 0.014	0.0487 ± 0.003	0.0483 ± 0.002	0.0154 ± 0.004	0.0036 ± 0.001
MultiVAE++	0.2687 ± 0.009	0.0877 ± 0.01	0.2357 ± 0.003	0.1858 ± 0.004	0.1046 ± 0.005	0.0972 ± 0.005	0.0224 ± 0.007	0.005 ± 0.002
F-SDRM	**0.3225** ± 0.019	**0.109** ± 0.01	0.232 ± 0.007	**0.1891** ± 0.009	0.1026 ± 0.007	0.0935 ± 0.007	0.0234 ± 0.003	0.0054 ± 0.001
M-SDRM	0.2953 ± 0.019	0.1026 ± 0.006	**0.2425** ± 0.015	0.186 ± 0.008	**0.1059** ± 0.014	**0.099** ± 0.015	**0.0273** ± 0.0059	**0.006** ± 0.001
Improvement	**20.02 %**	**24.28 %**	**2.88 %**	**1.77 %**	**1.24 %**	**1.85 %**	**21.87 %**	**20 %**

C. RESULTS

We compare SDRM against the baseline generative models used during SDRM training and present the average and standard deviation of Recall@k and NDCG@k over five runs in Tables 3 and 4. We also report the overall percentage improvement of SDRM over the best-performing baseline model for top-10 generated recommendations. In Table 6, we report the average improvement over MultiVAE++ in Recall@k and NDCG@k, calculated over all the recommender algorithm and dataset combinations for both the augmented and the synthetic dataset training setups. We report additional improvements for various top-k ($k \in 1, 3, 5, 10, 20, 50$) over the MultiVAE++ baseline in the appendix in Tables 7 and 8.

1) TRAINING WITH AUGMENTED DATASET

As shown in Table 3, F-SDRM and M-SDRM achieved the best or second-best results in all Recall@10 and all but one NDCG@10 results. There are many runs where MultiVAE++ placed second-best over either F-SDRM or M-SDRM, highlighting that even doing diffusion over a Gaussian latent variable can boost the performance. Overall, SDRM improves over the baseline generative methods by 4.48% for Recall@10 and 5.07% for NDCG@10 with the single best overall improvement from baseline of 24.28% and 21.87% for Recall and NDCG, respectively.

2) REPLACING TRAINING DATA WITH THE SYNTHETIC DATASET

As shown in Table 4, when substituting the original datasets with synthetic datasets, SDRM again consistently achieves the best or second best overall results for each of the recommender models and averages an overall improvement of 2.08% for Recall@10 and 0.88% for NDCG@10, achieving as high as 14.67% and 8.47% for Recall and NDCG, respectively, on the Amazon Digital Music dataset. This demonstrates the potential of using SDRM as a method to substitute an original dataset and achieve competitive or improving performance.

3) COMPARING SDRM AGAINST MULTIVAE

Since SDRM employs a MultiVAE for compressing and representing user-item data as a smaller latent variable z, then learns to transform one Gaussian representation into another, it raises a legitimate question whether the additional computational effort for SDRM genuinely enhances performance compared to using MultiVAE alone. To address this concern, we compared the percentage increase of max Recall and NDCG value between M-SDRM and F-SDRM (represented as SDRM in Figure 2) against MultiVAE++ from the same runs.

We observe that SDRM improves the overall Recall@k and NDCG@k scores compared to using MultiVAE across different k values. Our method achieves an average 6.81%

TABLE 4. **Overall performance comparison between baselines by training with synthetic data. The best results are in bold and the second best are underlined. Average overall improvement: Recall 2.08%, NDCG 0.88%.**

Model	SVD							
Dataset	ALB		ML-100k		ML-1M		ADM	
Baseline	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10
Original	0.3113 ± 0	0.287 ± 0	0.3716 ± 0	0.3973 ± 0	**0.3769** ± 0	**0.4035** ± 0	0.0624 ± 0	0.0427 ± 0
CTGAN	0.0654 ± 0.054	0.056 ± 0.054	0.2215 ± 0.011	0.225 ± 0.012	0.2623 ± 0.008	0.2769 ± 0.008	0.0152 ± 0.002	0.0095 ± 0.001
TVAE	0.2521 ± 0.005	0.2503 ± 0.004	0.2641 ± 0.008	0.2766 ± 0.006	0.3082 ± 0.006	0.325 ± 0.008	0.0214 ± 0	0.0143 ± 0
CODIGEM	0.3044 ± 0.008	0.2813 ± 0.007	0.2799 ± 0.004	0.286 ± 0.007	0.2624 ± 0.005	0.2774 ± 0.005	0.0481 ± 0.003	0.0295 ± 0.001
DiffRec	0.2702 ± 0.013	0.2541 ± 0.02	0.3476 ± 0.006	0.3624 ± 0.009	0.3354 ± 0.011	0.3595 ± 0.013	0.0258 ± 0.004	0.0197 ± 0.003
MultiVAE	0.3299 ± 0.011	0.3063 ± 0.008	0.3724 ± 0.008	0.3937 ± 0.007	0.3134 ± 0.011	0.3359 ± 0.012	0.0393 ± 0.005	0.0246 ± 0.004
MultiVAE++	0.3406 ± 0.003	0.3188 ± 0.001	0.3889 ± 0.008	0.414 ± 0.008	0.3704 ± 0.004	0.3935 ± 0.004	0.0617 ± 0.003	0.0412 ± 0.001
F-SDRM	**0.3471** ± 0.005	**0.3229** ± 0.007	0.3931 ± 0.006	**0.4176** ± 0.003	0.3707 ± 0.002	0.3944 ± 0.002	**0.0628** ± 0.002	**0.0428** ± 0.001
M-SDRM	0.3384 ± 0.007	0.3174 ± 0.004	**0.3946** ± 0.006	0.4176 ± 0.006	0.3703 ± 0.003	0.3946 ± 0.003	0.0622 ± 0.003	0.0423 ± 0.002
Improvement	**1.87%**	**1.27%**	**1.45%**	**0.86%**	**-1.65%**	**-2.21%**	**0.64%**	**0.23%**
Model	MLP							
Original	0.3183 ± 0.006	0.3007 ± 0.003	0.3569 ± 0.007	**0.3788** ± 0.004	0.3319 ± 0.006	0.3558 ± 0.005	0.0194 ± 0.001	0.0128 ± 0.001
CTGAN	0.2651 ± 0.008	0.2189 ± 0.021	0.2017 ± 0.016	0.2197 ± 0.011	0.2081 ± 0.001	0.2216 ± 0.003	0.0011 ± 0.001	0.0007 ± 0.001
TVAE	0.2438 ± 0.001	0.2431 ± 0.001	0.1334 ± 0.007	0.1586 ± 0.006	0.1569 ± 0.017	0.1674 ± 0.015	0.0153 ± 0.003	0.0095 ± 0.001
CODIGEM	0.1568 ± 0.095	0.1012 ± 0.061	0.1445 ± 0.019	0.1403 ± 0.018	0.1882 ± 0.006	0.187 ± 0.01	0.0249 ± 0.003	0.0173 ± 0.001
DiffRec	0.3065 ± 0.005	0.2947 ± 0.003	0.2997 ± 0.016	0.3107 ± 0.014	0.2349 ± 0.033	0.2526 ± 0.034	0.0181 ± 0.002	0.0115 ± 0.001
MultiVAE	0.311 ± 0.012	0.2925 ± 0.012	0.3359 ± 0.005	0.3465 ± 0.007	0.3385 ± 0.005	0.3633 ± 0.005	0.0172 ± 0.004	0.012 ± 0.002
MultiVAE++	0.3356 ± 0.011	0.3079 ± 0.011	0.3488 ± 0.01	0.3616 ± 0.007	0.3409 ± 0.004	0.3636 ± 0.004	**0.071** ± 0.011	**0.0492** ± 0.007
F-SDRM	0.3375 ± 0.008	0.3103 ± 0.01	**0.3601** ± 0.006	0.3754 ± 0.006	**0.3451** ± 0.006	0.3693 ± 0.006	0.013 ± 0.004	0.0085 ± 0.003
M-SDRM	**0.3474** ± 0.009	**0.3142** ± 0.005	0.3544 ± 0.006	0.3755 ± 0.007	0.3438 ± 0.005	**0.3702** ± 0.004	0.0668 ± 0.005	0.0451 ± 0.004
Improvement	**3.51 %**	**2.04 %**	**0.89%**	**-0.87 %**	**1.23%**	**1.81 %**	**-6.28 %**	**-9.09%**
Model	NeuMF							
Original	0.2399 ± 0.009	0.0818 ± 0.011	0.1006 ± 0.009	0.096 ± 0.005	0.0113 ± 0.001	0.0094 ± 0.001	0.0057 ± 0.001	0.0011 ± 0.001
CTGAN	0.1786 ± 0.083	0.0405 ± 0.018	0.0532 ± 0.014	0.05 ± 0.019	0.0053 ± 0.001	0.0055 ± 0.001	0.0073 ± 0.002	0.0014 ± 0.001
TVAE	0.0957 ± 0.034	0.0231 ± 0.011	0.1289 ± 0.011	0.1326 ± 0.008	0.0237 ± 0.005	0.023 ± 0.005	0.009 ± 0.008	0.0013 ± 0.001
CODIGEM	0.1154 ± 0.108	0.0294 ± 0.032	0.0782 ± 0.014	0.0552 ± 0.012	0.0606 ± 0.006	0.0574 ± 0.005	0.0022 ± 0.001	0.0004 ± 0.001
DiffRec	0.2613 ± 0.017	0.0881 ± 0.006	0.0726 ± 0.025	0.0629 ± 0.027	0.0143 ± 0.007	0.0134 ± 0.006	0.0168 ± 0.005	0.0031 ± 0.001
MultiVAE	0.0834 ± 0.025	0.0196 ± 0.007	0.0951 ± 0.014	0.0894 ± 0.014	0.0198 ± 0.004	0.0195 ± 0.004	0.0174 ± 0.003	0.0042 ± 0.001
MultiVAE++	0.331 ± 0.01	**0.1143** ± 0.002	0.2309 ± 0.013	0.1853 ± 0.009	0.126 ± 0.008	0.1154 ± 0.008	0.0259 ± 0.004	0.0059 ± 0.001
F-SDRM	**0.3339** ± 0.009	0.1137 ± 0.002	**0.2421** ± 0.009	**0.1913** ± 0.007	0.1265 ± 0.011	0.1146 ± 0.012	0.028 ± 0.014	0.0058 ± 0.002
M-SDRM	0.332 ± 0.008	0.1141 ± 0.001	0.2303 ± 0.014	0.1814 ± 0.01	**0.1296** ± 0.014	**0.1212** ± 0.014	**0.0297** ± 0.008	**0.0064** ± 0.002
Improvement	**0.87 %**	**-0.17 %**	**4.85 %**	**3.23 %**	**2.85 %**	**5.02 %**	**14.67 %**	**8.47 %**

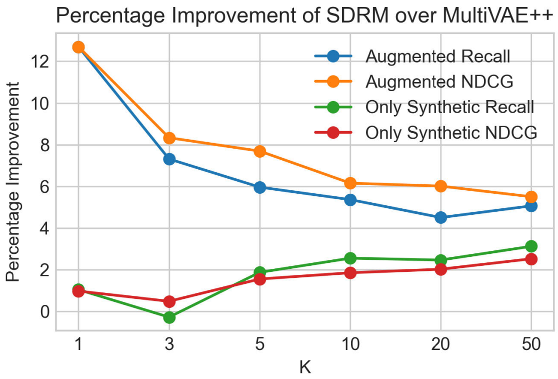

FIGURE 2. **Baseline improvement of SDRM over MultiVAE++ across all datasets for various sizes of top-*k* recommendation lists. Augmented average improvement: Recall@*k*: 6.81%, NDCG@*k*: 7.73%, Synthetic average improvement: Recall@*k*: 1.79%, NDCG@*k*: 1.56%, Combined average improvement: Recall@*k*: 4.30%, NDCG@*k*: 4.65%.**

and 7.73% increase in Recall@k and NDCG@k when augmenting the original data with synthetic and 1.79% and 1.56% increase in Recall@k and NDCG@k. The scores shown in Figure 2 and Table 6 (located in the Appendix) are the average increase across each of the three baseline recommender models and four datasets for each Recall@k and NDCG@k. SDRM has the greatest average performance increase in Recall for lower k values scores when used for augmentation and a higher average performance increase in NDCG for larger k scores when used as a substitution method. These findings further demonstrate that using a score-based diffusion model between the encoder and decoder of a MultiVAE architecture can significantly increase performance.

4) DISCUSSION OF RESULTS

We hypothesize that SDRM generates better synthetic data than MultiVAE++ because we use a score-based diffusion model to model the VAE's prior distribution $p_\theta(z)$. This allows us to create higher resolution noise processing before reconstruction using the decoder.

In the diffusion process, different timesteps t affect data interpolation—larger t values result in coarser and more varied interpolations, whereas smaller t values preserve global features without fine details [11], [46]. By sampling across a range of t values, M-SDRM surpasses F-SDRM in capturing global patterns within latent z space, which can improve recommender systems by better identifying intrinsic user or item clusters [17].

Additionally, as shown by our extensive evaluation, our method surpasses other tabular data generation methods [24], previous recommendation variational autoencoders [13], and earlier recommendation diffusion models [20], [21] in synthesizing recommendation datasets. We intentionally designed our approach to leverage the strengths of both variational autoencoders and diffusion models: the high-quality sampling capabilities of diffusion models and the mode coverage and diversity offered by variational autoencoders.

This strategic combination enables us to outperform all existing data generation methods for recommendation datasets.

D. PRIVACY-PRESERVATION

It has been previously shown that diffusion models and variational autoencoders can effectively reproduce training data even when sampled from a Gaussian distribution [71], [72]. It has also been observed that using a Gaussian distribution for generating synthetic data does not guarantee the privacy of the data [73]. However, considering the high sparsity of recommendation data and the design of SDRM to generate binary item-rating preferences, we posit that our approach does not replicate the original training data, even when adhering to the same sparsity constraints as the original data.

To confirm that the proposed model is privacy-sensitive, we present empirical evidence that demonstrates that the overwhelming majority of SDRM-generated data does not reproduce the training examples from each dataset, while maintaining a similar distribution of rated items and user ratings as to the original.

1) ASSESSING SIMILARITY

To assess the difference between the synthetic and original data, we employ the Jaccard similarity metric, which quantifies the similarity between two sets by dividing the size of the intersection of the sets by the size of their union.

$$J(x, \hat{x}) = \frac{|x \cap \hat{x}|}{|x \cup \hat{x}|} = \frac{|x \cap \hat{x}|}{|x| + |\hat{x}| - |x \cap \hat{x}|} \tag{12}$$

We obtain a score for each pairwise comparison of each synthetic user $\hat{x}$ to genuine user data x using M-SDRM and the MultiVAE++ and present the results in Table 5. We also explore the distribution of the items-per-user and user-per-items cardinalities in the synthetic data compared to the original dataset, as shown in Figure 3.

TABLE 5. Pairwise comparison using Jaccard similarity on original data vs synthetic data with M-SDRM and MultiVAE++ on Amazon Digital Music dataset.

Similarity	M-SDRM	Total %	MultiVAE++	Total %
0.0 - 0.1	90,331,367	98.88%	89,949,754	98.46%
0.1 - 0.2	989,059	1.08%	1,365,015	1.49%
0.2 - 0.3	32,636	< 1%	38,417	< 1%
0.3 - 0.4	1,842	< 1%	1,796	< 1%
0.4 - 0.5	320	< 1%	289	< 1%
0.5 - 0.6	77	< 1%	64	< 1%
0.6 - 0.7	43	< 1%	16	< 1%
0.7 - 0.8	12	< 1%	8	< 1%
0.8 - 0.9	8	< 1%	2	< 1%
0.9 - 1.0	0	< 1%	3	< 1%

Even though each original dataset has been pre-processed to account for each user having a minimum of 5 ratings per item, SDRM produces synthetic users with less than five items (something that is not necessarily bad, given that it better reflects real datasets that have such cold-start users and items). Even so, SDRM still adheres to the natural long-tail distribution in recommendation data, where a smaller number of users (items) account for the majority of related item (user) ratings in the dataset, leading us to believe SDRM accurately models intricate item-rating distributions in a recommendation dataset.

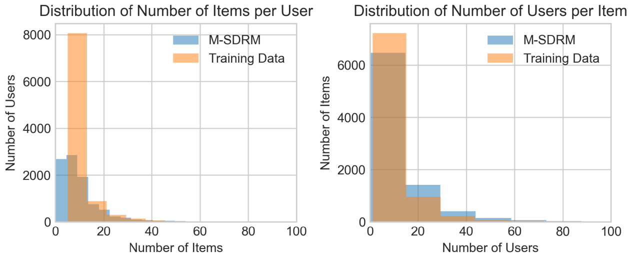

FIGURE 3. Distribution of the number of items and users for the Amazon Digital Music dataset.

2) ASSESSING PRIVATE SYNTHETIC DATA

Table 5 reveals that using M-SDRM to generate datasets of the same size and sparsity as the original does not reproduce (copy) the training examples, resulting in almost 99% dissimilarity. This significant difference is partly illustrated in Figure 3, demonstrating that the synthetic dataset's long-tail distribution of rated items is less right-skewed than the original. Despite this, the user-per-item distribution remains consistent between the synthetic and original datasets, highlighting SDRM's effectiveness in synthesizing data while preserving preference distributions per item.

Overall, our analysis demonstrates that SDRM-generated data can balance the ability to protect users in privacy-sensitive circumstances while still offering the ability to train recommender system models with equal or better performance than the original, something that true differential privacy implementations inherently struggle with [29].

VI. CONCLUSION

In this work, we propose SDRM, a novel approach to generating synthetic recommendation data through score-based diffusion modeling. We take advantage of the VAE framework to capture compressed latent representations into a Gaussian distribution and apply multi-resolution sampling to further improve distributional modeling for recommendation systems. Due to the disjointed training process and modeling of the latent z vector using score-based diffusion, we increase the performance of the variational inference process in a VAE when sampling new data, leading to significant improvements in performance in synthesizing recommendation datasets. Through various ablation studies, we show that our model is capable of modeling complex distributions and achieves an overall average improvement of 4.30% in Recall@k and 4.65% in NDCG@k when applying these novel samples to augment or replace existing datasets. We presented evidence that our method produces data that is 99% dissimilar from the original dataset while still retaining the characteristics of the

TABLE 6. Baseline improvement over MultiVAE++ across all datasets. Augmented average improvement: Recall@k: 6.81%, NDCG@k: 7.73%, Synthetic average improvement: Recall@k: 1.79%, NDCG@k: 1.56%, Combined average improvement: Recall@k 4.30%, NDCG@k 4.65%.

Metrics	R@1	R@3	R@5	R@10	R@20	R@50	N@1	N@3	N@5	N@10	N@20	N@50
Augmented	12.7 %	7.31 %	5.96 %	5.36 %	4.5 %	5.06 %	12.7 %	8.33 %	7.69 %	6.15 %	6.01 %	5.50 %
Only Synthetic	1.04 %	-0.28 %	1.87 %	2.55 %	2.46 %	3.12 %	0.97 %	0.48 %	1.55 %	1.85 %	2.02 %	2.52 %

TABLE 7. SDRM augmented data percentage improvement over MultiVAE++.

Datasets	ML-100k			ALB			ML-1M			ADM		
Baseline Models	SVD	NeuMF	MLP	SVD	NeuMF	MLP	SVD	NeuMF	MLP	SVD	NeuMF	MLP
Recall@1	1.16	18.81	2.43	6.04	40.28	5.29	2.57	7.92	3.91	5.00	51.72	7.26
Recall@3	0.45	4.33	3.57	2.56	9.62	0.96	1.37	6.16	2.99	16.67	30.95	8.12
Recall@5	0.29	4.60	1.68	1.94	13.89	-0.47	1.07	1.76	2.66	16.20	20.29	7.64
Recall@10	2.40	2.89	1.18	1.86	20.02	0.81	0.13	1.24	1.90	4.57	21.88	5.56
Recall@20	1.03	-1.95	1.76	0.76	18.38	0.55	0.40	1.11	2.10	2.60	20.11	7.18
Recall@50	0.84	2.00	0.10	2.36	17.60	-0.05	-0.02	-1.08	2.99	5.15	22.31	8.61
NDCG@1	1.16	18.81	2.43	6.04	40.28	5.29	2.57	7.92	3.91	5.00	51.72	7.26
NDCG@3	0.83	7.06	3.91	3.43	18.63	1.47	1.83	6.80	2.88	16.07	28.57	8.49
NDCG@5	0.70	5.27	2.65	2.80	22.31	0.84	1.49	3.60	2.64	15.89	25.93	8.20
NDCG@10	1.33	1.78	2.17	2.68	24.29	1.00	0.79	1.85	2.18	8.87	20.00	6.95
NDCG@20	0.62	2.82	1.83	2.33	22.52	0.80	0.71	2.25	2.18	7.18	21.43	7.48
NDCG@50	0.82	3.08	1.12	2.71	21.20	0.49	0.42	0.67	2.49	7.24	17.65	8.12

TABLE 8. SDRM synthetic training data percentage improvement over MultiVAE++.

Datasets	ML-100k			ALB			ML-1M			ADM		
Baseline Models	NeuMF	MLP	SVD	NeuMF	MLP	SVD	NeuMF	MLP	SVD	NeuMF	MLP	SVD
Recall@1	-6.94	13.36	2.42	0.52	0.00	2.48	9.88	3.90	1.30	-15.11	-9.06	9.72
Recall@3	2.07	6.40	-2.04	-1.91	2.05	-2.01	8.69	3.07	0.74	-10.07	-12.86	2.48
Recall@5	3.73	5.23	0.64	-0.62	2.26	-0.65	8.08	2.22	1.04	4.74	-9.88	5.67
Recall@10	4.85	3.24	1.47	0.88	3.52	1.91	2.86	1.23	0.08	14.67	-5.92	1.78
Recall@20	2.31	-1.02	0.17	-0.70	1.72	1.76	2.07	1.10	0.11	31.18	-9.68	0.52
Recall@50	2.69	0.35	1.37	-1.87	1.68	2.88	1.03	0.23	-0.02	33.84	-6.02	1.29
NDCG@1	-6.94	13.36	2.42	0.52	0.00	2.48	9.88	3.90	1.30	-15.91	-9.06	9.72
NDCG@3	0.64	7.92	-0.90	-0.92	1.44	-0.39	8.94	3.28	0.77	-8.70	-11.97	5.59
NDCG@5	2.08	5.47	0.58	0.00	1.48	0.19	8.53	2.71	0.88	1.30	-10.92	6.27
NDCG@10	3.24	3.84	0.87	-0.17	2.05	1.29	5.03	1.82	0.28	8.47	-8.33	3.88
NDCG@20	2.75	2.00	0.27	-0.65	1.36	1.14	3.79	1.55	0.27	19.57	-9.80	1.93
NDCG@50	2.54	2.44	0.73	-0.90	1.00	1.58	2.82	1.05	0.26	24.24	-8.10	2.55

original data distribution. This work addresses the problem of dataset sparsity but also creates a novel way for organizations to preserve the privacy of their users' data while still training robust recommender systems.

Our approach to synthesizing recommendation datasets using diffusion models addresses a gap in prior literature, which has primarily focused on synthesizing tabular datasets but often overlooks recommendation datasets in their benchmarks. We bridge this gap by not only comparing against popular tabular generators, but also evaluating recommender diffusion models and variational autoencoders.

A. LIMITATIONS

One limitation of our method is its reliance on a Multi-Layer Perceptron (MLP) network for denoising Gaussian data to generate user data one sample at a time. Unlike diffusion models used in image processing that utilize segmentation models for local pixel relationships, our approach may miss the chance to generate user data in batches, which could capture clustering patterns among users. Furthermore, our evaluations were limited to smaller recommendation datasets due to resource constraints. The largest dataset we tested, Amazon Digital Music, showcased SDRM's varying performance across different recommender models. This underscores the need for further research to evaluate SDRM's applicability and scalability to larger datasets. Additionally, our current method may not adequately model tabular datasets with continuous and categorical features of high cardinality, due to the heuristic approach used to enforce sparsity in SDRM's stochastic output.

B. FUTURE WORK

While we have demonstrated consistent improvements across various recommendation datasets, several avenues remain open for future research. Future efforts will aim to integrate auxiliary and temporal data to create synthetic data conditioned on user or group attributes. We believe this strategy can help mitigate biases and enhance fairness in recommendation datasets, enabling the training of fairness-aware recommendation models from a data-driven perspective. Further exploration could also yield success in experimenting with alternative noise perturbation techniques, network architectures, sampling methods, adversarial loss functions, and training methodologies.

APPENDIX
ADDITIONAL RESULTS TABLES

See Tables 6–8.

REFERENCES

[1] C. C. Aggarwal, "Recommender systems," in *The Textbook*, 1st ed. Cham, Switzerland: Springer, Apr. 2016, doi: 10.1007/978-3-319-29659-3.

[2] D. Paraschakis, "Recommender systems from an industrial and ethical perspective," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 463–466.

[3] S. Milano, M. Taddeo, and L. Floridi, "Recommender systems and their ethical challenges," *AI Soc.*, vol. 35, no. 4, pp. 957–967, Dec. 2020, doi: 10.1007/s00146-020-00950-y.

[4] A. Friedman, B. Knijnenburg, K. Vanhecke, L. Martens, and S. Berkovsky, *Privacy Aspects of Recommender Systems*, 2nd ed. Berlin, Germany: Springer Nature, 2015, pp. 649–688.

[5] F. Liu, Z. Cheng, H. Chen, Y. Wei, L. Nie, and M. Kankanhalli, "Privacy-preserving synthetic data generation for recommendation systems," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 1379–1389.

[6] N. Idrissi and A. Zellou, "A systematic literature review of sparsity issues in recommender systems," *Social Netw. Anal. Mining*, vol. 10, no. 1, p. 15, Dec. 2020, doi: 10.1007/s13278-020-0626-2.

[7] H. Gm, M. K. Gourisaria, M. Pandey, and S. S. Rautaray, "A comprehensive survey and analysis of generative models in machine learning," *Comput. Sci. Rev.*, vol. 38, Nov. 2020, Art. no. 100285. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013720303853

[8] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7327–7347, Nov. 2022, doi: 10.1109/TPAMI.2021.3116668.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–12.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, Banff, AB, Canada, Apr. 2014.

[11] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.

[12] Q. Wang, H. Yin, H. Wang, Q. V. H. Nguyen, Z. Huang, and L. Cui, "Enhancing collaborative filtering with generative augmentation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 548–556.

[13] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proc. World Wide Web Conf.*, 2018, pp. 689–698.

[14] B. Askari, J. Szlichta, and A. Salehi-Abari, "Variational autoencoders for top-K recommendation with implicit feedback," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2021, pp. 2061–2065.

[15] D.-K. Chae, J.-S. Kang, S.-W. Kim, and J.-T. Lee, "CFGAN: A generic collaborative filtering framework based on generative adversarial networks," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 137–146.

[16] Z. Wang, M. Gao, X. Wang, J. Yu, J. Wen, and Q. Xiong, "A minimax game for generative and discriminative sample models for recommendation," in *Proc. Adv. Knowl. Discovery Data Mining*, vol. 23. Berlin, Germany: Springer-Verlag, 2019, pp. 420–431.

[17] J. Ma, C. Zhou, P. Cui, H. Yang, and W. Zhu, "Learning disentangled representations for recommendation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[18] N. Sachdeva, G. Manco, E. Ritacco, and V. Pudi, "Sequential variational autoencoders for collaborative filtering," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 600–608.

[19] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 8780–8794.

[20] J. Walker, T. Zhong, F. Zhang, Q. Gao, and F. Zhou, "Recommendation via collaborative diffusion generative model," in *Knowledge Science, Engineering and Management* (Lecture Notes in Computer Science), vol. 13370, G. Memmi, B. Yang, L. Kong, T. Zhang, and M. Qiu, Eds. Cham, Switzerland: Springer, 2022, doi: 10.1007/978-3-031-10989-8_47.

[21] W. Wang, Y. Xu, F. Feng, X. Lin, X. He, and T.-S. Chua, "Diffusion recommender model," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, vol. 46, Jul. 2023, pp. 832–841, doi: 10.1145/3539618.3591663.

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

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

[24] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[25] A. Narayanan and V. Shmatikov, "How to break anonymity of the Netflix prize dataset," 2006, *arXiv:cs/0610105*.

[26] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, Aug. 2013, doi: 10.1561/0400000042.

[27] J. Hua, C. Xia, and S. Zhong, "Differentially private matrix factorization," in *Proc. 24th Int. Conf. Artif. Intell.* Washington, DC, USA: AAAI Press, 2015, pp. 1763–1770.

[28] M. Asad, S. Shaukat, E. Javanmardi, J. Nakazato, and M. Tsukada, "A comprehensive survey on privacy-preserving techniques in federated recommendation systems," *Appl. Sci.*, vol. 13, no. 10, p. 6201, May 2023.

[29] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–10.

[30] M. Slokom, M. Larson, and A. Hanjalic, "Partially synthetic data for recommender systems: Prediction performance and preference hiding," 2020, *arXiv:2008.03797*.

[31] S. Azizi, S. Kornblith, C. Saharia, M. Norouzi, and D. J. Fleet, "Synthetic data from diffusion models improves ImageNet classification," 2023, *arXiv:2304.08466*.

[32] B. Trabucco, K. Doherty, M. Gurinas, and R. Salakhutdinov, "Effective data augmentation with diffusion models," 2023, *arXiv:2302.07944*.

[33] W. Yao, Y. Shen, F. Nicolls, and S. Q. Wang, "Conditional diffusion model-based data augmentation for Alzheimer's prediction," in *Proc. Int. Conf. Neural Comput. Adv. Appl. (NCAA)*, vol. 1869. Singapore: Springer, 2023, doi: 10.1007/978-981-99-5844-3_3.

[34] A. Lesnikowski, G. de Souza Pereira Moreira, S. Rabhi, and K. Byleen-Higley, "Synthetic data and simulators for recommendation systems: Current state and future directions," 2021, *arXiv:2112.11022*.

[35] European Parliament and Council of the European Union. (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council*. [Online]. Available: https://data.europa.eu/eli/reg/2016/679/oj

[36] A. Kotelnikov, D. Baranchuk, I. Rubachev, and A. Babenko, "TabDDPM: Modelling tabular data with diffusion models," in *Proc. 40th Int. Conf. Mach. Learn.*, 2023, pp. 1–16.

[37] Z. Zhao, A. Kunar, R. Birke, and L. Y. Chen, "CTAB-GAN: Effective table data synthesizing," in *Proc. Asian Conf. Mach. Learn.*, 2021, pp. 97–112.

[38] M. Gao, J. Zhang, J. Yu, J. Li, J. Wen, and Q. Xiong, "Recommender systems based on generative adversarial networks: A problem-driven perspective," *Inf. Sci.*, vol. 546, pp. 1166–1185, Feb. 2021.

[39] H. Bharadhwaj, H. Park, and B. Y. Lim, "RecGAN: Recurrent generative adversarial networks for recommendation systems," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 372–376.

[40] W. Zhao, B. Wang, J. Ye, Y. Gao, M. Yang, and X. Chen, "Plastic: Prioritize long and short-term information in top-N recommendation using adversarial training," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, vol. 27, 2018, pp. 3676–3682, doi: 10.24963/ijcai.2018/511.

[41] X. Chen, S. Li, H. Li, S. Jiang, Y. Qi, and L. Song, "Generative adversarial user model for reinforcement learning based recommendation system," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 1052–1061.

[42] X. Bai, J. Guan, and H. Wang, "A model-based reinforcement learning with adversarial training for online recommendation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[43] A. Vahdat, K. Kreis, and J. Kautz, "Score-based generative modeling in latent space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 11287–11302.

[44] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, Jul. 2021, pp. 8162–8171.

[45] D. P. Kingma and M. Welling, *An Introduction to Variational Autoencoders*. Boston, MA, USA: Now, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/9051780

[46] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2256–2265.

[47] C. Luo, "Understanding diffusion models: A unified perspective," 2022, *arXiv:2208.11970*.

[48] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 21696–21707.

[49] D. P. Kingma and R. Gao, "Understanding the diffusion objective as a weighted integral of ELBOS," 2023, *arXiv:2303.00848*.

[50] H. Chefer, Y. Alaluf, Y. Vinker, L. Wolf, and D. Cohen-Or, "Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–10, Jul. 2023, doi: 10.1145/3592116.

[51] B. Zhang, C. Luo, D. Yu, H. Lin, X. Li, Y. Ye, and B. Zhang, "MetaDiff: Meta-learning with conditional diffusion for few-shot learning," 2023, *arXiv:2307.16424*.

[52] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Comput.*, vol. 23, no. 7, pp. 1661–1674, Jul. 2011.

[53] Y. Song and S. Ermon, *Generative Modeling By Estimating Gradients of the Data Distribution*. Red Hook, NY, USA: Curran Associates, 2019.

[54] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," 2020, *arXiv:2011.13456*.

[55] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, and C. Etmann, "Conditional image generation with score-based diffusion models," 2021, *arXiv:2111.13606*.

[56] T. Dockhorn, A. Vahdat, and K. Kreis, "Score-based generative modeling with critically-damped Langevin diffusion," 2021, *arXiv:2112.07068*.

[57] V. De Bortoli, E. Mathieu, M. Hutchinson, J. Thornton, Y. W. Teh, and A. Doucet, "Riemannian score-based generative modelling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 2406–2422.

[58] V. Voleti, C. Pal, and A. Oberman, "Score-based denoising diffusion with non-isotropic Gaussian noise models," 2022, *arXiv:2210.12254*.

[59] D. Gnaneshwar, B. Ramsundar, D. Gandhi, R. Kurchin, and V. Viswanathan, "Score-based generative models for molecule generation," 2022, *arXiv:2203.04698*.

[60] H. Sun, L. Yu, B. Dai, D. Schuurmans, and H. Dai, "Score-based continuous-time discrete diffusion models," 2022, *arXiv:2211.16750*.

[61] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2007, pp. 1–8.

[62] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for Youtube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 191–198.

[63] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.

[64] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 188–197. [Online]. Available: https://aclanthology.org/D19-1018

[65] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. 19, Jan. 2016, Art. no. 19, doi: 10.1145/2827872.

[66] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–12.

[67] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020.

[68] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2011, pp. 2546–2554.

[69] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, J. Ben-Tzur, M. Hardt, B. Recht, and A. Talwalkar, "A system for massively parallel hyper-parameter tuning," in *Proc. Conf. Mach. Learn. Syst.*, vol. 2, 2020, pp. 230–246.

[70] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2623–2631.

[71] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace, "Extracting training data from diffusion models," in *Proc. 32nd USENIX Secur. Symp. (USENIX Security)*, 2023, pp. 5253–5270.

[72] S.-C. Li, B.-C. Tai, and Y. Huang, "Evaluating variational autoencoder as a private data release mechanism for tabular data," in *Proc. IEEE 24th Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, Dec. 2019, pp. 198–1988.

[73] G. Heller and E. Fetaya, "Can stochastic gradient Langevin dynamics provide differential privacy for deep learning?" in *Proc. IEEE Conf. Secure Trustworthy Mach. Learn. (SaTML)*, Feb. 2023, pp. 68–106.

DEREK LILIENTHAL received the B.S. degree in computer science and minor in statistics from California State University, Monterey Bay, in 2022. He is currently pursuing the M.S. degree in artificial intelligence with San José State University. His research interests include deep learning, generative models, and recommendation systems.

PAUL MELLO received the B.S. degree in computer science from Sacramento State University, in 2016. He is currently pursuing the M.S. degree in artificial intelligence with San José State University, where he is researching diffusion models. His research interests include generative AI, computer vision, and deep learning.

MAGDALINI EIRINAKI (Member, IEEE) received the Ph.D. degree in computer science from the Department of Informatics, Athens University Economics and Business, in 2006. She is currently a Professor with the Computer Engineering Department, San José State University, CA, USA, where she is also the Director of the M.S. in Artificial Intelligence Program. Her research interests include recommender systems, machine learning, data mining, social graphs, and deep learning applications. She was a recipient of the 2019 Newnan Brothers Award for Faculty Excellence, the 2017 Applied Materials Award for Excellence in Teaching, and has received the SJSU Distinguished Faculty Mentor Award, in 2015, 2019, 2020, 2022, and 2023.

STAS TIOMKIN (Member, IEEE) received the B.Sc. degree in computer engineering and in physics from Hebrew University, the M.Sc. degree from the Viterbi Faculty of Electrical and Computer Engineering, Technion Israel Institute of Technology, and the Ph.D. degree in computer science and engineering from Hebrew University, Israel. He is currently an Assistant Professor in computer engineering with San José State University. Previously, he was a Postdoctoral Researcher with BAIR, UC Berkeley.