Collaborative Generative Adversarial Network for Recommendation Systems

Yuzhen Tong
The University of Queensland, Australia
yuzhen.tong@uq.net.au

Yadan Luo
The University of Queensland, Australia
lyadanluol@gmail.com

Zheng Zhang
The University of Queensland, Australia
darrenzz219@gmail.com

Shazia Sadiq
The University of Queensland, Australia
shazia@itee.uq.edu.au

Peng Cui
Tsinghua University, China
cuip@tsinghua.edu.cn

***Abstract*—Recommendation systems have been a core part of daily Internet life. Conventional recommendation models hardly defend adversaries due to the natural noise like misclicking. Recent researches on GAN-based recommendation systems can improve the robustness of the learning models, yielding the state-of-the-art performance. The basic idea is to adopt an interplay minimax game on two recommendation systems by picking negative samples as fake items and employ reinforcement learning policy. However, such strategy may lead to mode collapse and result in high vulnerability to adversarial perturbations on its model parameters. In this paper, we propose a new collaborative framework, namely Collaborative Generative Adversarial Network (CGAN), which adopts Variational Auto-encoder (VAE) as the generator and performs adversarial training in the continuous embedding space. The formulation of CGAN has two advantages: 1) its auto-encoder takes the role of generator to mimic the true distribution of users preferences over items by capturing subtle latent factors underlying user-item interactions; 2) the adversarial training in continuous space enhances models robustness and performance. Extensive experiments conducted on two real-world benchmark recommendation datasets demonstrate the superior performance of our CGAN in comparison with the state-of-the-art GAN-based methods.**

***Index Terms*—Collaborative Filter, Adversarial Training, Auto-encoder, Recommendation Systems**

I. Introduction

With the rapid development of the Internet and information technology, the problem of information overload is getting more and more critical. To help people efficiently figure out the content of interests from the exponentially increasing information, recommendation systems [3], [12], [22], have been extensively incorporated as one of the core parts in various advanced web applications, such as online shopping, news recommendations and personalized advertising.

The essential research of recommendation system is to discover user-user, item-item and user-item relationships. Recommendation methods can be roughly divided into three categories: item-based [4], [16], user-based [18], [23] and model-based [13], [14], [20]. Item-based models recommend similar items to the given user according to user's historical interactions with the target items, while user-based approaches are more concerning the similarities among users. As for the model-based methods, they aims to extract the features from preferred items and recommend items with homologous features. This paper mainly focuses on the model-based recommendation and improves the quality of the extracted features.

Most of the existing recommendation systems rely on users' feedback to detect the user preferences, based on which recommendations are provided for end users. As one of the most widely-used technique in recommendation systems, Collaborative Filtering (CF) [9] aims to exploit the feedback given by users who have the similar preferences to the target users. Among all the CF algorithms, Matrix Factorisation (MF) [11] has become the most popular strategy in the past decade, which learns the latent features of items and users. It factorizes a high-dimensional sparse user-item rating matrix into a low-dimensional latent vector space. Despite its linear nature, MF has been successfully introduced in various applications. By taking advantages of the powerful nonlinear activation functions and deep structures, a number of DNN-based recommendation methods have been proposed to mine the latent factors underlying user-item interactions, resulting in better performance over traditional MF-based models.

The existing methods generally focus on exploring the latent feature vectors from the users feedback to predict the users preference over items. However, the performance of the learning systems are still far away from satisfactory because of the unqualified training data. Specifically, it is worth noting that the training data inevitably contain the natural noise such as misclicking and biased feedback, while the conventional models are very vulnerable to adversarial examples. Although some adversarial training methods have been proposed to alleviate this difficulty, these models have their own drawbacks: 1) DNN-based models [8], [21] are lack of robustness and fragile to adversarial examples due to the natural noise, and 2) adversarial training methods [17], [19] cannot extract nonlinear feature vectors from user-item interactions and usually suffer from mode collapse.

In this paper, we aim to fully explore DNNs nonlinear nature for better feature extraction, meanwhile the power of adversarial training is employed to boost systems robustness to adversarial examples. To this end, a novel Collaborative GAN (CGAN) learning framework is proposed to strengthen

2473-3490/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDEW.2019.00-16

robustness and performance of the recommendation systems. As shown in Figure 1, we leverage the Generative Adversarial Network (GAN) [6] as our basic structure and integrate Variational Auto-Encoder (VAE) [10] as the generator. In this framework, we perform a minimax game in the continuous latent space so that the loss function is differentiable and use the Wasserstein distance to illustrate the distribution divergence between the generator and user-item interaction data. As such, the proposed model can dexterously prevent mode collapse and improve the training speed. By taking the advantage of the VAEs nonlinear nature, our CGAN can learn better aggregated posterior distribution from the given data and generate high-quality fake samples.

The main contributions of this work are as follows:

- We propose a novel Collaborative Generative Adversarial Network (CGAN) model which seamlessly incorporates VAE into GAN model to enhance the robustness to adversarial examples.
- We introduce the Wasserstein Distance with gradient penalty into the item recommendation tasks to depict the distribution divergence between the generated data and ground-truth data, such that both the prediction performance and training speed are simultaneously improved.
- Extensive experiments performed on two public datasets demonstrate the effectiveness of our framework on both performance and robustness comparing with the state-of-art methods.

The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 formally defines the problem and basic knowledge of VAE and GAN. In Section 4, we introduce the employed models and present the experimental results in Section 5. Finally We conclude in Section 6.

II. RELATED WORK

We first review the related works which strongly motivate ours. They can be roughly summarized into three categories:

A. Matrix Factorization

MF has been recognized as the basic yet most effective model in recommendation for several years. Being a germ of representation learning, MF represents each user and item in a lower dimensional latent space. The core idea of MF is to estimate a user's preference on an item as the inner product between their embedding vectors. MF has two advantages: 1) the computation cost of MF is relatively small due to its simple operations; 2) it helps solve the problem of user-item interaction matrix sparsity by factorizing the sparse matrix into lower dimension matrixes. However, MF can not capture the non-linear latent features of items and users, which results in poor performances in recommendation tasks.

B. Neural Networks for Collaborative Filtering

Comparing with traditional shallow MF-based methods, neural-network-based CF models take the advantages of non-linear activation functions and have better capacity to capture representations for users and items from implicit feedback. And two paper are most related to our methods are Neural Matrix Factorization and Collaborative Denoising Autoencoder, which are the state-of-the-art approaches.

Neural Matrix Factorization (NeuMF) [8] employs Multi-Layer Perceptron (MLP) [5] instead of dot product to extract non-linear interactions between the user and the item latent factors. The experiments demonstrate significant improvement over standard baselines on two small benchmark datasets. The number of parameters of NeuMF grows linearly with both the numbers of the users and the items, which becomes problematic for large-scale datasets.

Collaborative Denoisiong Autoencoder (CDAE) [21] augments the basic DAE by contaminating the input data with noise, which enhances the models robustness and reconstruction capacity. Similar to the NeuMF, CDAE also suffers from parameter explosion on big datasets. We compare our method with NeuMF and CDAE in Section 5.

C. GAN in Item Recommendation

In the last decade, adversarial training techniques have gained great popularity in computer vision field, which aims to strengthen the robustness of conventionally trained model to adversarial examples. Recently, this techniques have been successfully applied to information retrieval tasks, such as IRGAN [19] and GraphGAN [17]. In addition to recommendation tasks, IRGAN can be applied to web search and question answering, and GraphGAN can be applied to link prediction and node classification. From the viewpoint of CF, IRGAN is designed to counterpose the generative and discriminative item recommendation models. Otherwise than conventional GAN-based models which take noise as the generators input and generate fake samples, IRGAN utilizes generative model to predict relevant items to the given user by picking possible items from the negative sample set. By introducing the policy gradient, this method extends GAN from the continuous latent spaces to discrete ones. However, IRGAN employs the KL divergence to measure the difference between the ground-truth and the generated distributions, which results in low train speed and potential mode collapses. In GraphGAN, the generator samples the most confusing items based on the the discriminators feedback, while the discriminator aims to distinguishing the true items from the generated ones. The proposed method in this paper is compared with IRGAN and GraphGAN in Section 5.

III. PRELIMINARIES

A. Problem Definition

This paper takes the ratings as training and testing data to complete the recommendation task. In a recommendation setting, there are N users, M items and a set of user-item interaction $r_{ij} \in R^{N \times M}$. Each entry r_{ij} of r corresponds to the rating of user i on item j. The row vector r_i is the preference to each item from the given user i. Let $u_i, v_j \in R^K$ be the latent factor vectors to user i and item j respectively generated by matrix factorisation, where K represents the

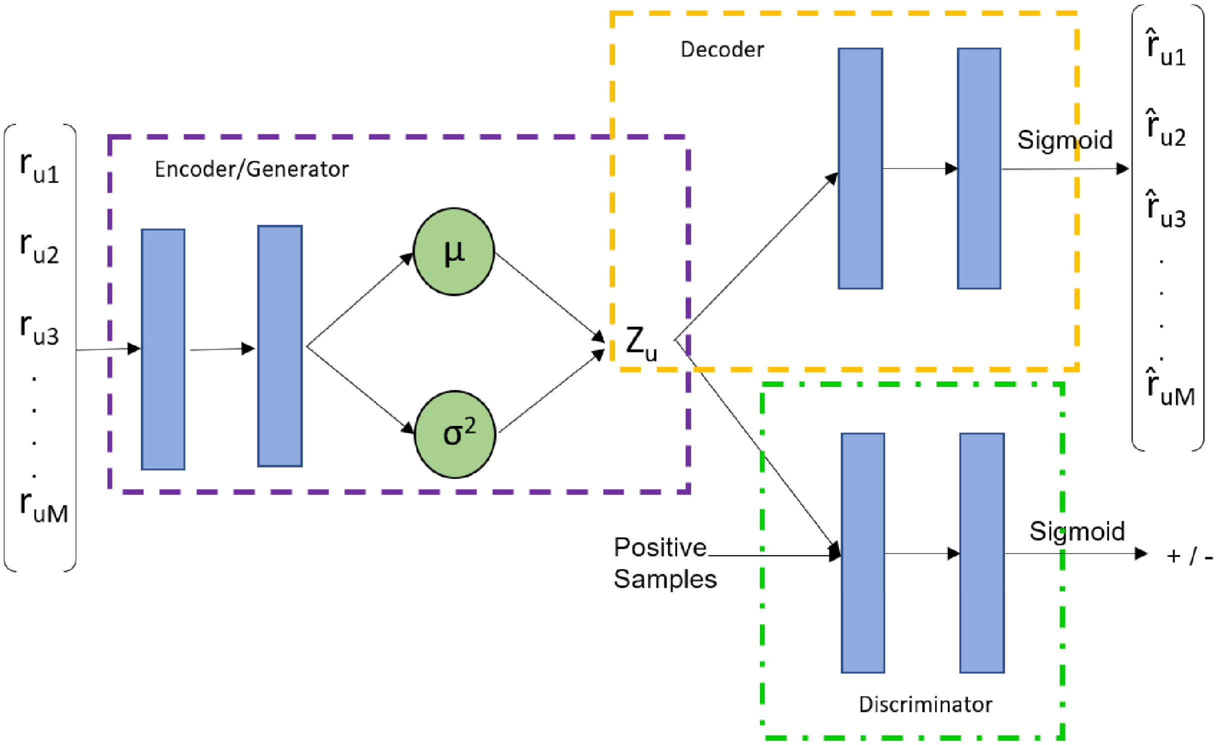

Fig. 1. The illustration of general architecture of the proposed Collaborative Generative Adversarial Network. It consists of three major components: encoder network, decoder network and discriminator model. The input of the encoder is a row vector r_u of the rating matrix and the output of the decoder is the reconstructed row vector $\hat{r}_u$. Each edge resembles a parametrized mapping $f(Wx+b)$ with activation function f and parameters W , b. Without specific label, the activation function is rectified linear. μ and σ^2 represent the mean vector and the standard deviation vector, respectively. Z_u stands for the aggregated posterior distribution of VAE.

dimension of the continuous latent space. The ultimate goal of recommendation systems is to recommend a list of items to each user which suit the user most.

B. Generative Adversarial Networks (GAN)

The GAN is inspired by the Nash equilibrium in game theory. The learning process becomes a procedure of competition between generative model G and discriminative model D to directly shape the output distribution of the network via back-propagation. The generator learns to capture the distribution underlying ground-truth data so that it can generate data that is very similar to the real data. Formally, G and D are playing the following two-player minimax game with the value function $V(G, D)$ that evaluates the cost of training:

$$\begin{aligned}\min_{\theta}\max_{\phi} V(G,D) = E_{x\sim P_r}[logD(x)] \\ + E_{x\sim P_g}[log(1-D(x))]\end{aligned} \tag{1}$$

where $D(x)$ indicates the estimated probability of x being in the ground truth. G and D iterate in optimizing the respective model parameters while minimizing and maximizing $V(G, D)$ respectively.

To alleviate the issue of unstable convergence and time-consuming training, Improved Wasserstein GAN (WGAN-GP) is proposed by Ishaan et al [7]. which removes the log function and adds a gradient penalty term on the original critic loss. The objective function of WGAN-GP is as follow:

$$\begin{aligned}\mathcal{L} &= -E_{x\sim P_r}[D(x)] + E_{x\sim P_g}[D(x)] \\ &\quad + \lambda E_{\hat{x}\sim P_{\hat{x}}(\hat{x})}[(|| \bigtriangledown_{\hat{x}} D(\hat{x})||_2 - 1)^2] \\ \hat{x} &= \epsilon x_r + (1-\epsilon)x_g \quad \epsilon \sim Uniform[0,1]\end{aligned} \tag{2}$$

wherex_r, x_g represent the ground-truth and generated data respectively. $\hat{x}$ is sampled from x_r and x_g by Equation 2. $P_{\hat{x}}$ means the distribution of $\hat{x}$. $D(x)$ indicates the probability of x being in the ground truth estimated by the discriminator model.

C. Variational Auto-encoder (VAE)

VAE is rooted in Bayesian inference and wants to model the underlying probability distribution of data so that it could sample new data from that distribution. To be specific, It has the Lipschitz constraint on encoder, which forces encoder to generate latent vector obeying unit Gaussian distribution. So the loss for VAE contains two parts: the generative loss, which is a mean squared error that measures how accurately the network reconstructed the input data, and a latent loss, which is the KL divergence that measures how closely the latent variables match a unit Gaussian. To optimize the KL divergence, the encoder generates a mean vector μ and a standard deviation vector σ^2 as reconstructed parameters of the latent vector.

$$\mathcal{L}_{VAE} = \mathcal{L}_{gen} + \mathcal{L}_{latent} \tag{3}$$

$$\mathcal{L}_{gen} = \sum_{i=1}^{N}(r_i - \hat{r}_i)^2 \tag{4}$$

$$\mathcal{L}_{latent} = \frac{1}{2}\sum_{i=1}^{N}(\sigma_i^2 - log\sigma_i^2 - 1) + \frac{1}{2}\mu_i^2 \tag{5}$$

where r_i and $\hat{r}_i$ represents the input data and the reconstructed data respectively.

IV. PROPOSED METHOD

In this paper, VAE takes the role of generator, fed with the rating row vector r_i. After encoding, the generator learns the aggregated posterior distribution from the given data and generate fake item vectors through the embedding layer. As for the discriminator, it aims at distinguishing the true item vectors from the generated ones by estimating the probability of item embedding vectors being relevant to the given user vectors.

A. Overall Objective

The proposed method employs the VAE as the generator and performs the adversarial training between G and D. The objective functions of G and D are presented as follows:

$$J^D = \mathcal{L}^D_{GAN} + \beta||\phi||_2 \tag{6}$$

$$J^G = \frac{1}{2}\mathcal{L}^G_{GAN} + \frac{1}{2}\mathcal{L}_{VAE} + \beta||\phi||_2 \tag{7}$$

where $\mathcal{L}^G_{GAN}$ and $\mathcal{L}^D_{GAN}$ are the loss function of G and D respectively; $\mathcal{L}_{VAE}$ stands for the loss function of VAE; $|| \quad ||_2$ represents the L2 norm which helps avoid overfitting weighted by the regularization coefficient β; ϕ and θ indicate the model parameters of D and G. In this paper, we let both J^G and J^D be minimization problems. We train G and D by alternately optimizing J^G and J^D.

B. Generator

generative model $p_\theta(v|u)$ aims to mimic the underlying distribution $p_r(v|u)$ from the training dataset and based on that, generates item vectors in order to fool the discriminative model D. Specifically, while keeping the discriminator $s_\phi(i,u)$ fixed after its minimization, we optimize the generative model via minimizing its objective function J^G. J^G consists of two loss function, $\mathcal{L}^G_{GAN}$ and $\mathcal{L}_{VAE}$, and L_2 regularization term. the loss function of generator can be denoted as:

$$\mathcal{L}^G_{GAN} = -E_{i\sim P_\theta(i|u)}[D(v|u)] \tag{8}$$

We adopt VAE as our generator. The VAE is trained to generate its output as similar as its input. After being fed with the sparse rating row vector r_i as input, it aims at predicting potential rating score to reconstruct a dense rating vector, denoted as $\hat{r}_i$. The loss for VAE is shown as follows:

$$\begin{aligned}\mathcal{L}_{VAE} = \sum_{i=1}^{N}\{[(r_i - \hat{r}_i) \times b_i]^2 \\ + \frac{1}{2}(\sigma_i^2 - log\sigma_i^2 - 1) + \frac{1}{2}\mu_i^2\}\end{aligned} \tag{9}$$

where b_i is based on r_i, where $b_{ij} = 1$ if i has rated an item j, $b_{ij} = 0$ otherwise.

C. Discriminator

The objective for the discriminator is to maximize the likelihood of correctly distinguishing true item vectors from generated item samples. With the observed relevant items, and the ones generated by the current optimal generative model, one can then obtain the optimal parameters for the discriminative model. With the implementation of WGAN-GP in Equation 2, the loss function of the discriminator is shown as follows:

$$\begin{aligned}\mathcal{L}^D_{GAN} = -E_{v\sim P_r(v|u)}[D(v|u)] + E_{x\sim P_\theta(v|u)}[D(v|u)] \\ + \lambda E_{\hat{v}\sim P_{\hat{v}}(\hat{v}|u)}[(|| \bigtriangledown_{\hat{v}} D(\hat{v}|u)||_2 - 1)^2]\end{aligned} \tag{10}$$

where P_r and P_θ stands for the distributions of the ground-truth and the generated data, $\hat{v}$ is sampled from the generated and true item embedding vectors.

To illustrate the power of our method, we utilize a simple scoring function $s_\phi(v,u)$ for the preference of user latent vector u to item latent vector v and employ a sigmoid function as the discriminator model to estimate the probability of item embedding vector v being relevant to the given user u.

$$\begin{aligned}s_\phi(v,u) = u^T v \\ D(v|u) = sigmoid(s_\phi(v,u)) = \frac{exp(s_\phi(v,u))}{1 + exp(s_\phi(v,u))}\end{aligned} \tag{11}$$

Algorithm 1 Collaborative Generative Adversarial Network (a.k.a CGAN)

```
Input: Rating matrix $R \in R^{N \times M}$
  Initialize $p_\theta(v|u)$ , $s_\phi(v,u)$ with random weights $\theta$, $\phi$ Pretrain $p_\theta(v|u)$ , $s_\phi(v,u)$ using $R$
  repeat
    for g-steps do
      $p_\theta(v|u)$ generates $K$ items embedding vectors for each user $u$
      Update generator parameters via minimizing the generator objective function $J^G$
    end for
    for d-steps do
      Use current $p_\theta(v|u)$ to generate fake item vectors and combine with given positive item vectors
      Train discriminator $s\phi(v,u)$ by minimizing the discriminator objective function $J^D$
    end for
  until CGAN converges
```

D. Overall Algorithm

Before the adversarial training, the generator and discriminator can be initialized by pretrained models. Then during the adversarial training stage, the generator and discriminator are trained alternatively. The details of proposed methods are shown in the Algorithm 1.

V. EXPERIMENTS

A. Datasets

We conduct experiments on two widely-used recommendation system datasets: Movielens (1M) and Netflix(100M). Their details are shown in Table I.

TABLE I
CHARACTERISTICES OF THE DATASETS

Dateset	Users	Items	Ratings
Movielens	3,706	6,040	1,000,209
Netflix	480,189	17,770	100,480,507

1.Movielens. The Movielens is a widely used benchmark dataset for recommendation system. We choose the 1million subset which contains users rating towards movies in the scale of 1-5.

2.Netflix. This implicit feedback dataset is constructed for video recommendation and has more than 100M 1-5 rating records.

B. Compared methods

The proposed methods are compared with six popular recommender methods:

ItemPop [2] ranks items by the descending order of popularity measured by the number of interactions in the training set. It is a non-personalized method but is widely regarded as a benchmark to evaluate other personalized recommendation models.

MF-BPR [15] aims to optimize MF with Bayesian Personalized Ranking loss function. Due to BPR's pairwise nature, it is a highly competitive method for personalized item recommendation. We tune the learning rate, latent factor dimensions and regularization coefficients.

CDAE [21] applies the Denoising Auto-encoder to item recommendation, which adds noise to the input data before feeding into the auto-encoder. We use the original model released by the author and tune the noise level and regularizer coefficients.

NeuMF [8] is a state-of-the-art item recommendation model, which combines multi-layer perceptrons (MLP) and MF to extract better non-linear features from the user-item interactions. We followed the setting of the paper and tune the hyper-parameters in the hidden layers to reach its best performance.

GraphGAN [17] adopts GAN into graph representation learning, which learns better low-dimension embedding with adversarial training. We use the original implementation released by the authors and followed the setting of the paper. Then we tune the learning rate and epoch numbers of generator and discriminator.

IRGAN [19] combines two recommendation systems by performing a minimax game. The generative model aims to pick possible items from the negative sample pool, while the discriminative model determines whether those items from the ground truth data or generated. We use the original implementation released by the authors and tune the hyper-parameters for generator and discriminator separately.

This set of comparing models represents the state-of-the-art performance in the task of item recommendation. Specifically, GraphGAN and IRGAN take the advantages of generative adversarial networks, which show good performances on item recommendation. CDAE and NFC utilize the non-linear nature of DNN outperforming shallow methods like MF.

C. Evaluation Metrics

Our ultimate goal is to recommend more relevant item to the given user. We use four common metrics as follows:

Precision at K: Precision at K (P@K) is the number of retrieved relevant items $|G \cap R|$ divided by the number of known relevant items $|G|$ among the top K, denoted as:

$$P@K = \frac{|G \cap R|}{|G|} \tag{12}$$

Normalized discounted cumulative gain (NDCG): The discounted cumulative gain (DCG) measures the ranking quality of the recommended items to evaluate the usefulness, or gain, of a item based on its position in the result list. It increases when relevant items are placed higher in the list. The normalized DCG (NDCG) is determined dividing the DCG by the ideal DCG (IDCG). In the latter, the recommended items are perfectly ranked. Equation 13 shows the DCG, while Equation 14 shows the IDCG. When a recommended item is relevant, rel is 1, while it is 0 otherwise. $|REL|$ represents the list of relevant items in the corpus up to the position P.

$$DCG = rel_1 + \sum_{i=2}^{P} \frac{rel_i}{log_2(i+1)} \tag{13}$$

$$IDCG = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{14}$$

$$NDCG = \frac{DCG}{IDCG} \tag{15}$$

Mean Reciprocal Rank (MRR). The reciprocal rank (RR) assesses the reciprocal of the position at which the first relevant item occurs in R. It is 1 if a relevant item is in the first position, 0.5 if a relevant item occurs in the second position, and so on. The mean reciprocal rank (MRR), shown in Equation 17, is the RR averaged across the recommended list. The position of the first relevant item in the list of recommendations for the i-th user is denoted by $rank_i$, the number of users is denoted as N.

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_i} \tag{16}$$

TABLE II
TOP-K RECOMMENDATION PERFORMANCE ON MOVIELENS-1M AT K=3,K=5 AND K=10. THE BEST RESULT OF EACH SETTING IS HIGHLIGHTED IN BOLD FONT. (MOVIELENS-1M)

Alg.	P@3	P@5	P@10	MAP	NDCG@3	NDCG@5	NDCG@10	MRR
ItemPop	0.274	0.246	0.221	0.165	0.191	0.212	0.256	0.292
MF-BPR	0.378	0.334	0.305	0.213	0.322	0.374	0.412	0.549
CDAE	0.394	0.367	0.336	0.238	0.334	0.388	0.430	0.569
NeuMF	0.427	0.394	0.363	0.261	0.346	0.402	0.447	0.581
GraphGAN	0.335	0.312	0.293	0.187	0.281	0.323	0.369	0.432
IRGAN	0.402	0.375	0.351	0.245	0.313	0.382	0.415	0.530
CGAN	**0.449**	**0.428**	**0.398**	**0.287**	**0.369**	**0.417**	**0.458**	**0.597**

TABLE III
TOP-K RECOMMENDATION PERFORMANCE ON MOVIELENS-1M AT K=3,K=5 AND K=10. THE BEST RESULT OF EACH SETTING IS HIGHLIGHTED IN BOLD FONT. (NETFLIX)

Alg.	P@3	P@5	P@10	MAP	NDCG@3	NDCG@5	NDCG@10	MRR
ItemPop	0.279	0.251	0.233	0.124	0.191	0.212	0.256	0.312
MF-BPR	0.381	0.369	0.344	0.163	0.362	0.414	0.432	0.549
CDAE	0.449	0.425	0.402	0.185	0.384	0.428	0.449	0.599
NeuMF	0.461	0.442	0.418	0.197	0.417	0.430	0.462	0.649
GraphGAN	0.354	0.329	0.299	0.125	0.319	0.335	0.341	0.412
IRGAN	0.445	0.433	0.392	0.172	0.409	0.420	0.449	0.637
CGAN	**0.472**	**0.459**	**0.435**	**0.207**	**0.438**	**0.469**	**0.478**	**0.662**

Mean Average Precision (MAP): MAP is a precision metric that emphasizes ranking relevant items higher. r_i represents the i-th relevant item and The position of the first relevant item in the list of recommendations for the i-th user is denoted by $rank_i$, the number of users is denoted as N.

$$MAP = \frac{1}{N}\sum_{i=1}^{|G \cap R|}\frac{r_i}{rank_i} \tag{17}$$

D. Implement Details

The proposed method converts all the ratings in both Movielens and Netflix into 1. The datasets are split into three subsets: 80% ratings as training set, 10% for the validation set and 10% for the test set.

For the VAE, we use symmetric encoder and decoder models, which are parameterized by Multilayer Perceptrons. The input, hidden layer and output dimensionality of the encoder is M, 600 and 300 respectively, where M is the total number of items, while for the decoder we use layers with dimensionality 300, 600, and M, respectively. The rectified linear function is used as a non-linear activation between all layers except the output layer of the encoder. We apply dropout with probability 0.5 at the input layer, use a batch size of 500 users and update the model weights with the Adam optimizer for 30 epochs at the training.

In the GAN framework, we pre-trained the generator and the discriminator separately with the training set for 10 epochs and the coefficient λ for the gradient penalty is set to 10.

E. Experimental Results and Analysis

Table II and Table III report the evaluation results on the Movielens-1M and Netflix datasets. On Movielens-1M, CGAN's relative improvement in performance over IRGAN are 14.13% for P@5, 17.14% for MAP, 9.16% for NDCG@5 and 12.64% for MRR. On Netflix, CGANs relative improvement over IRGAN are 6.02% for P@5, 20.34% for MAP, 11.98% for NDCG@5 and 3.92% for MRR.

The relative performance improvements over IRGAN indicate that CGAN is an effective extension to IRGAN. Because CGAN has the same discriminator with IRGAN, we argue that the combination of VAE and WGAN with gradient penalty leads to better representation learning which in fact improves recommendation performance. Compared to GraphGAN, which also adopts GAN in item recommendation tasks, CGAN is consistently better, achieving the relative performance improvements of 37.17% for P@5, 53.47% for MAP, 29.10 % for NDCG@5 and 38.19% for MRR on Movielens-1M. On Netflix, CGANs relative improvements over GraphGAN are 39.51% for P@5, 65.60% for MAP, 40.12% for NDCG@5 and 60.67% for MRR.

We observe that CDAE and NeuMF have comparable performance to CGAN on both datasets. All the three models perform significantly better than IRGAN and GraphGAN which indicates that using NN-based models instead of the conventional matrix factorization methods is a way to extract better latent features to achieve better recommendation performance. However, CGAN outperforms CDAE and NeuMF on both datasets, demonstrating that the adversarial training boosts the recommendation models in term of accuracy.

F. Ablation study

To validate the effect of the adversarial training by utilizing the Wasserstein distance with gradient penalty. A series of experiments is conducted and the results are shown in Table IV. In the table, GAN refers to the original generative adversarial network which employs the Kullback-Leibler (KL) to depict the distribution divergence between the generator and the ground-truth. WGAN [1] replaces the KL divergence with the Wasserstein distance on the original GAN. WGAN-GP adds the gradient penalty on the framework of WGAN. It can be clearly observed that with the same pretrained generator and discriminator, the model with WGAN has faster training speed than the model with original GAN which demonstrates the Wasserstein distance is better to depict the divergence than KL in item recommendation tasks, and WGAN-GP significantly outperforms WGAN in epoch 5, epoch 10 and epoch 15 on both metrics which indicates the gradient penalty boosts the WGANs performance in terms of speed and accuracy.

TABLE IV
THE IMPACT OF WASSERSTEIN DISTANCE AND GRADIENT PENALTY ON MOVIELEN-1M

Epoch	5		10		15	
Alg.	P@5	N@5	P@5	N@5	P@5	N@5
GAN	0.359	0.385	0.367	0.394	0.371	0.399
WGAN	0.361	0.389	0.372	0.402	0.386	0.411
WGAN-GP	0.374	0.392	0.383	0.417	0.418	0.426

To verify the effect of the VAE model, we take MF and auto-encoder with multi-layer perceptrons as the generator in our proposed model. And the results are shown in Table V. The results demonstrate that auto-encoder model has better performance than the MF model in all evaluation metrics on Movielens-1M, which indicates the NN-based model has better representation learning capability due to its non-linear nature. Comparing with the auto-encoder model, the proposed methods relative improvements in performance are 14.13% for P@5, 17.14% for MAP, 9.16% for NDCG@5 and 12.64% for MRR, indicating that the variational auto-encoder extracts better representations than conventional auto-encoder who has the same depth of networks.

TABLE V
THE IMPACT OF VAE ON MOVIELEN-1M

Alg.	P@3	P@5	P@10	N@3	N@5	N@10
MF	0.362	0.313	0.293	0.325	0.364	0.402
AE	0.381	0.363	0.331	0.335	0.372	0.423
VAE	0.449	0.428	0.398	0.369	0.417	0.458

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented Collaborative GAN for item recommendation. Specifically, we introduce the VAE to recommendation systems with adversarial training, which can improve the performance of the latent representation of users and items by imposing an adversarial regularization. Meanwhile, we employ the Wasserstein distance with gradient penalty to boost model's training speed and robustness to the adversarial examples. In our evaluation, we conduct extensive experiments on two widely used item recommendation benchmark datasets, which shows significant improvements over strong baselines. In the ablation study, we conduct two sets of experiments to demonstrate the highly positive effect of the Wasserstein distance, the gradient penalty and the variational auto-encoder.

In future, we plan to extend the variational auto-encoder by replacing the standard Gaussian prior with user-depend priors, which encodes both collaborative information and user preferences, benefiting the personalized recommendation. Secondly, we will replace the conventional point-wise collaborative filtering model with pair-wise ranking models like the Bayesian personalized ranking (BPR) [15] to boost the capability of discriminator, which improves the performance of the whole GAN framework. Thirdly, we will employ Collaborative GAN on the recently developed neural CF models such as NeuMF [8] to further advance the performance of item recommendation. Lastly, inspired by IRGAN [19] and GraphGAN [17] which have good performance in many different fields, we aims to apply our method to other IR tasks, such as web search, question answering, graph embedding representation, text retrieval.

VII. ACKNOWLEDGEMENT

This work is partially supported by ARC FT130101530, NSFC No.61628206 and ARC DP170103954.

REFERENCES

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46. ACM, 2010.

[3] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, et al. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 293–296. ACM, 2010.

[4] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[5] M. W. Gardner and S. Dorling. Artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[7] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[9] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. Ieee, 2008.

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

[11] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[12] D. W. McDonald and M. S. Ackerman. Expertise recommender: a flexible recommendation system and architecture. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 231–240. ACM, 2000.

[13] B. Mobasher, R. Burke, and J. J. Sandvig. Model-based collaborative filtering as a defense against profile injection attacks. In *AAAI*, volume 6, page 1388, 2006.

[14] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles. Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 473–480. Morgan Kaufmann Publishers Inc., 2000.

[15] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press, 2009.

[16] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[17] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, and M. Guo. Graphgan: graph representation learning with generative adversarial nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[18] J. Wang, A. P. De Vries, and M. J. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508. ACM, 2006.

[19] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 515–524. ACM, 2017.

[20] Q. Wang, H. Yin, Z. Hu, D. Lian, H. Wang, and Z. Huang. Neural memory streaming recommender networks with adversarial training. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2467–2475. ACM, 2018.

[21] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 153–162. ACM, 2016.

[22] H. Yin, L. Zou, Q. V. H. Nguyen, Z. Huang, and X. Zhou. Joint event-partner recommendation in event-based social networks. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 929–940. IEEE, 2018.

[23] Z.-D. Zhao and M.-S. Shang. User-based collaborative-filtering recommendation algorithms on hadoop. In *Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on*, pages 478–481. IEEE, 2010.