GoRec: A Generative Cold-Start Recommendation Framework

Haoyue Bai
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
baihaoyue621@gmail.com

Min Hou*
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
hmhoumin@gmail.com

Le Wu
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
Institute of Dataspace, Hefei Comprehensive National Science Center
lewu.ustc@gmail.com

Yonghui Yang
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
yyh.hfut@gmail.com

Kun Zhang
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
zhang1028kun@gmail.com

Richang Hong
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
hongrc.hfut@gmail.com

Meng Wang
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
eric.mengwang@gmail.com

ABSTRACT

Multimedia-based recommendation models learn user and item preference representation by fusing both the user-item collaborative signals and the multimedia content signals. In real scenarios, cold items appear in the test stage without any user interaction record. How to perform cold item recommendation is challenging as the training items and test items have different data distributions. These hybrid preference representations contained auxiliary collaborative signals, so current solutions designed alignment functions to transfer learned hybrid preference representations to cold items. Despite the effectiveness, we argue that they are still limited as these models relied heavily on the manually carefully designed alignment functions, which are easily influenced by the limited item records and noises in the training data.

To tackle the above limitations, we propose a ***Generative cold-start Recommendation (GoRec)*** framework for multimedia-based new item recommendation. Specifically, we design a Conditional Variational AutoEncoder (CVAE) based method that first estimates the underlying distribution of each warm item conditioned on the multimedia content representation. Then, we propose a uniformity-enhanced optimization objective to ensure the latent space of CVAE is more distinguishable and informative. In the inference stage, a generative approach is designed to obtain warm-up new item representations from the latent distribution. Please note that GoRec is applicable to arbitrary recommendation backbones. Extensive experiments on three real datasets and various recommendation backbones verify the superiority of our proposed framework. The code is available at https://github.com/HaoyueBai98/GoRec.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

recommender system, cold start, conditional variational auto-encoder

ACM Reference Format:
Haoyue Bai, Min Hou, Le Wu, Yonghui Yang, Kun Zhang, Richang Hong, and Meng Wang. 2023. GoRec: A Generative Cold-Start Recommendation Framework. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3612238

1 INTRODUCTION

Personalized recommendations have become essential in various online applications such as e-commerce and advertising to help users manage information overload [4, 23, 36, 37]. Learning precise user and item representations is the key to building an effective recommender. Among this field, the multimedia-based recommendation is becoming an attractive research area, which fully takes

*Corresponding author.

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada.

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612238

advantage of both user-item interactions and rich-information multimedia content for representation learning [10, 34, 35]. In general, multimedia-based recommendation feeds multimedia content and ID as input to the embedding layer for user and item representation learning, then models interactions by inner product or neural networks. Despite the effectiveness to provide high-quality representations for recommendation demand, multimedia-based recommenders still fail to generalize to cold-start recommendation scenarios. In real scenarios, cold items quickly emerge over time, especially on the news or short-video recommendation platforms. Cold items have only multimedia content while lacking historical interactions, so how to generate high-quality item representations is the key challenge to cold-start item recommendations.

To overcome this problem, a common solution is to use multimedia content (e.g., image [18], text [8], knowledge graph [30]) as the bridge between well-trained warm item representations and cold item representations. To make the distribution of warm and cold representations consistent, previous works[1, 2, 33, 45] attempt to narrow the distance between warm representations and content representations by elaborately designing alignment functions (e.g., sum square error[45], mutual information maximization [33]). Then warm-up cold items by using the corresponding content representations. Despite the effectiveness, we argue that this schema still has some limitations. Concretely, (1) the alignment functions need to be carefully designed, and it is difficult to guarantee that the distribution of warm and cold representations can be consistent through the functions [33]. Moreover, the cold embeddings are generated from contents while the warm item embeddings are learned from both historical interactions and content, making the warm items have inherently more information. The way of bringing the two distributions close to each other may reduce the representation ability of warm items. (2) One-to-one alignment for each discrete sample pair may make the model affected by some noise [31, 44].

In this work, we innovatively break the alignment function-based schema and propose a ***G**enerative c**o**ld **Rec**ommendation (GoRec)* framework for multimedia-based new item recommendation. *GoRec* directly models the conditional distribution of warm embeddings solely based on content. By modeling the distribution, the warm representation can be obtained from the contents during the test phase, irrespective of the availability of historical interaction data. This generative approach fundamentally solves the previous limitations. *GoRec* is a CVAE-based framework. Specifically, we first use pre-train warm user and item representations as the input. Then, we estimate the underlying distribution of latent variable conditioned on the multimedia content representation. Following this, we generate pseudo item representations by conditionally sampling from the estimated distribution. Based on the generated and the pre-trained item representations, we build the well-known ELBO-based optimization objective. Besides, we design a uniformity-enhanced optimization objective to ensure the latent space of CVAE is more distinguishable and informative. In the test stage, for each cold item, we first sample a latent variable from an estimated latent distribution, then combine the corresponding multimedia content to generate item representation for recommendations. Considering that random sampling ignores the characteristics of the cold item, we devise a cluster-aware approach to obtain the item's fuzzy pre-trained representations, then the item's latent distribution is estimated with the combination of the fuzzy representations and multimedia content. Our major contributions are listed as follows:

- We propose a novel CVAE-based generative cold-start item recommendation framework (*GoRec*) for multimedia-based new item recommendation, which can generate high-quality new item representations according to the content features.
- We design a uniformity-enhanced optimization objective to ensure the latent space of CVAE is more distinguishable and informative. We devise a cluster-aware approach to obtain the item's fuzzy pre-trained representations to better generate item representations.
- We conduct extensive experiments on three real-world datasets to demonstrate the superiority and effectiveness of the proposed model in new item recommendation tasks.

2 PROBLEM FORMULATION

Implicit Recommendation. In this paper, we focus on the implicit feedback recommendation scenario and let $\mathcal{U}(|\mathcal{U}| = N)$ and $\mathcal{V}(|\mathcal{V}| = M)$ denote the sets of users and items. Besides, every item has multimedia content features, which can be transformed into content representations via a generic feature extractor. We denote the content representation of items as $\mathbf{C} \in \mathbb{R}^{M \times d_c}$. $\mathbf{c}_j \in \mathbb{R}^{d_c}$ denote content representation of item j. Let $\boldsymbol{R} \in \mathbb{R}^{M \times N}$ be the user-item interaction matrix, $R_{ij} = 1$ if user i has interacted with item j, otherwise $R_{ij} = 0$. The aim of implicit recommendation model $\mathcal{F}_\pi$ is to infer the probability $\hat{y}_{ij}$ of user i preferring item j:

$$\hat{y}_{ij} = \mathcal{F}_\pi(\mathcal{P}(i), \mathcal{Q}(j, \mathbf{c}_j), \mathbf{R}), \tag{1}$$

where $\mathcal{P}$ and $\mathcal{Q}$ are user and item embedding layer, respectively. The parameters are learned during the training process. We denote the $\mathcal{F}_\pi^*$ as the optimal recommendation backbone trained in the warm user and item set (users and items with historical interactions).

Item Cold Start Problem. In the item cold-start problem, a cold item has been interacted with by limited users. In this work, we focus on the completely cold-start problem, that performs recommendations to the new items set $\mathcal{V}_{\text{new}}(|\mathcal{V}_{\text{new}}| = M_{\text{new}})$ without any historical interactions. The challenges posed by item cold start problem are the warm items in the training process and the test new items have different data distributions, and it is hard to generate effective new item representation to feed into the recommendation model $\mathcal{F}_\pi^*$. Our work focuses on how to learn a good new item embedding model $\mathcal{Q}_{\text{new}}(\cdot)$. The ultimate goal is to infer the probability $\hat{y}_{ij}$ user i preferring **new item** j:

$$\hat{y}_{ij} = \mathcal{F}_\pi^*(\mathcal{P}(i), \mathcal{Q}_{\text{new}}(j, \mathbf{c}_j)). \tag{2}$$

We aim to build a recommendation model-agnostic framework, and thus the choice of $\mathcal{F}_\pi^*$ can be arbitrary.

3 METHODOLOGY

In this section, we introduce the proposed *Generative cOld Recommendation (GoRec)* framework for cold-start item recommendations, which could be applied to all existing recommendation models. In the following part, we first introduce each part of *GoRec* in detail. After that, we elaborate on the learning algorithm and new item recommendation process.

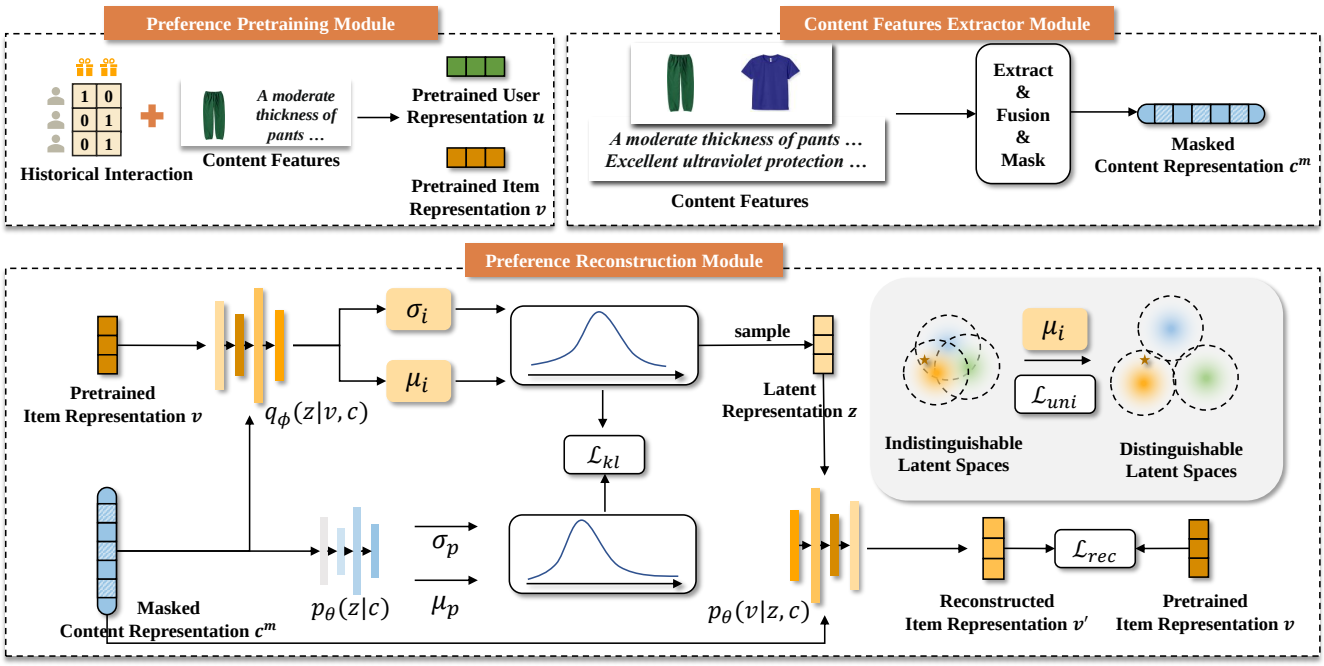


Figure 1: Model overview.

3.1 Framework Architecture

Multimedia-based recommendation models learn user and item preference representation by fusing both the user-item collaborative signals and the multimedia content signals. The challenge caused by cold-start item problems is the absence of new item preference representations, which should contain collaborative signals. In *GoRec*, we aim to estimate the underlying distribution of item preference representation conditioned on the multimedia content representation. Then we can obtain preference representations of new items easily from this distribution. As illustrated in Figure 1, *GoRec* consists of two main components, the preference pretraining module (PPM), and the preference reconstruction module (PRM). The former learns the item preference representations from the warm users and items. Then, the preference reconstruction module is responsible for learning the distribution of item preference representation conditioned on the multimedia content representation. In this subsection, we elaborate on these two modules.

3.1.1 ***Preference Pretraining Module.*** The function of PPM is to learn the item preference representations from the warm user set and warm item set. Generally, PPM contains two steps: (1) the embedding layer to obtain preference representations of users and items, (2) the interaction layer to infer the probability of users preferring items. After training, we get the item preference representation from the first step.

Formally, given the warm user set $\mathcal{U}$ and warm item set $\mathcal{V}$, item content representations $\mathbf{C}$, and user-item interaction matrix $\boldsymbol{R} \in \mathbb{R}^{M \times N}$, PPM serves for inferring the preference probability $\hat{y}_{ij}$ of user i to item j:

$$\hat{y}_{ij} = \mathcal{F}_\pi(\mathcal{P}(i), \mathcal{Q}(j, \mathbf{c}_j), \mathbf{R}), \tag{3}$$

where $\mathcal{P}$ and $\mathcal{Q}$ are user and item embedding layer, respectively. $\mathcal{F}_\pi$ is the recommendation backbone. The architecture of $\mathcal{P}, \mathcal{Q}, \mathcal{F}_\pi$ can be arbitrary, and all existing embedding-based recommender systems can be chosen, such as VBPR [10], LightGCN [11], SimGCL [40]. When training the PPM, the classical BPR loss [20] could be adopted:

$$\mathcal{L}_{\text{BPR}} = \sum_{(i,j,j') \in \mathcal{U} \cup \mathcal{V}} -\ln \sigma\left(\hat{y}_{ij} - \hat{y}_{ij'}\right) + \lambda \|\Theta\|^2, \tag{4}$$

where Θ includes all model parameters, λ is the hyper-parameter of regularization, j' is the negative sample item of i. After the training process, we can obtain item preference representation of warm item j as follows:

$$\mathbf{v}_j = \mathcal{Q}(j, \mathbf{c}_j). \tag{5}$$

3.1.2 ***Preference Reconstruction Module.*** Upon the PPM, we further build PRM which uses a latent variable for learning the distribution of the coherent item preference representation. In PRM, item preference representation $\mathbf{v}$ is generated conditioned on the given corresponding content representation $\mathbf{c}$ and a latent variable $\mathbf{z}$, which captures the distribution of the preference representations. Formally, we define the conditional distribution as $p(\mathbf{v}|\mathbf{c}) = \int_{\mathbf{z}} p(\mathbf{v}|\mathbf{c}, \mathbf{z}) p(\mathbf{z}|\mathbf{c}) d\mathbf{z}$. Since the integration over $\mathbf{z}$ is intractable, we therefore use neural networks to estimate and apply variational inference to optimize the corresponding evidence lower bound (ELBO):

$$\begin{aligned} \log p(\mathbf{v}|\mathbf{c}) &= \log \int_{\mathbf{z}} p(\mathbf{v}|\mathbf{c}, \mathbf{z}) p(\mathbf{z}|\mathbf{c}) d\mathbf{z} \\ &\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{c},\mathbf{v})}\left[\log p_\theta(\mathbf{v}|\mathbf{c}, \mathbf{z})\right] - KL\left(q_\phi(\mathbf{z}|\mathbf{c}, \mathbf{v}) \| p_\theta(\mathbf{z}|\mathbf{c})\right), \end{aligned} \tag{6}$$

where $p_\theta(\mathbf{z}|\mathbf{c})$ is the *prior net*, $q_\phi(\mathbf{v}|\mathbf{c},\mathbf{z})$ is the *preference generator*, and $q_\phi(\mathbf{z}|\mathbf{c},\mathbf{v})$ is the *posterior net*. KL denotes the KL-divergence. We assume that $\mathbf{z}$ follows multivariate Gaussian distribution with a diagonal covariance matrix. Then we describe the three neural networks mentioned above.

Prior Net. The prior net $p_\theta(\mathbf{z}|\mathbf{c})$ aims to encode the given content features to the latent space. We first extract raw multimedia features into low-dimension representations and then project them to latent space of $\mathbf{z}$. Several generic feature extractors (such as ResNet [9] and BERT[5]) are used to extract features in different modals. We then integrate them by concatenation to get the content representation $\mathbf{c}_i \in \mathbb{R}^{d_c}$ of each item i. However, it is impossible to guarantee all new items have the same content representations as the old. To give the model the ability to process unseen content representation, we propose randomly masking part of the content representation for training:

$$\mathbf{C}^m = \mathbf{M} \odot \mathbf{C}, \tag{7}$$

where $\mathbf{M} \in \mathbb{R}^{M \times d_c}$ is a mask matrix randomly generated and $\odot$ represents the element-wise multiplication. We use $\mathbf{c}_i^m$ to denote the masked content representation of item i. Since we assume $\mathbf{z}$ follows isotropic Gaussian distribution, thus $p(\mathbf{z}|\mathbf{c}) \sim \mathcal{N}\left(\mu, \sigma^2 \mathrm{I}\right)$, the prior net estimate the μ and σ use two linear layer as:

$$\mu_i = \mathbf{c}_i^m \cdot \mathbf{W}_\mu + \mathbf{b}_\mu, \quad \sigma_i = \mathbf{c}_i^m \cdot \mathbf{W}_\sigma + \mathbf{b}_\sigma, \tag{8}$$

where $\mathbf{W}_\mu, \mathbf{W}_\sigma \in \mathbb{R}^{d_c \times d_z}$ and $\mathbf{b}_\mu, \mathbf{b}_\sigma \in \mathbb{R}^{d_z}$ are the parameters to be learned, d_z is the dimension of latent variable $\mathbf{z}$.

Posterior Net. The posterior net $q_\phi(\mathbf{z}|\mathbf{c},\mathbf{v})$ can be viewed as an encoder, that encodes both the item preference representations and masked content representation into the latent space. Similarly, we assume $p(\mathbf{z}|\mathbf{c},\mathbf{v}) \sim \mathcal{N}\left(\mu', \sigma'^2 \mathrm{I}\right)$, then we use two linear transformation function and have:

$$\mu_i' = [\mathbf{v}_i; \mathbf{c}_i^m] \cdot \mathbf{W}_{\mu'} + \mathbf{b}_{\mu'}, \quad \sigma_i' = [\mathbf{v}_i; \mathbf{c}_i^m] \cdot \mathbf{W}_{\sigma'} + \mathbf{b}_{\sigma'}, \tag{9}$$

where $\mathbf{W}_{\mu'}, \mathbf{W}_{\sigma'} \in \mathbb{R}^{(d_c+d_v)\times d_z}$ and $\mathbf{b}_{\mu'}, \mathbf{b}_{\sigma'} \in \mathbb{R}^{d_z}$ are the parameters to be learned, and $[\cdot;\cdot]$ indicates the operation of concatenate.

Preference Generator. The preference generator $q_\phi(\mathbf{v}|\mathbf{c},\mathbf{z})$ is a decoder. Given a random sample $\mathbf{z}_i$ in $\mathcal{N}\left(\mu_i', \sigma_i'^2 \mathrm{I}\right)$ and $\mathbf{c}_i^m$, the preference generator reconstructs the item preference representation as $\mathbf{v}_i'$. Direct sampling would make the entire model non-differentiable, rendering existing optimization methods unable to calculate gradients. To address this problem, the reparameterization trick [13] is applied. It works as follows: Instead of directly sampling from $\mathcal{N}(\mu_i', \sigma_i'^2 \mathrm{I})$, we first sample from a standard normal distribution $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, and then we can get:

$$\mathbf{z}_i = \mu_i' + \sigma_i' \odot \epsilon. \tag{10}$$

Since sampling from ϵ does not depend on the network, it makes the proposed model differentiable again. Then we try to reconstruct $\mathbf{v}_i$ by parameterizing $p_\theta(\mathbf{v}|\mathbf{z},\mathbf{c})$ as follows:

$$\mathbf{v}_i' = f_\psi([\mathbf{z}_i; \mathbf{c}_i^m]). \tag{11}$$

Herein, $f_\psi(\cdot)$ is a simple but effective MultiLayer Perception (MLP), and ψ denote parameters of $f_\psi(\cdot)$.

3.2 Uniformity Enhanced Optimization

The learning process consists of two stages. First, we pre-train the preference pretraining module in warm datasets by minimizing the $\mathcal{L}_{\text{BPR}}$ as described in Eqn. 4. After optimization, we train the preference reconstruction module. In the following, we will elaborate on the training details of the preference reconstruction module.

ELBO Loss. Following the loss of CVAE, and as mentioned in Eqn. 6, we define the ELBO loss as:

$$\begin{aligned}\mathcal{L}_{ELBO} &= \mathcal{L}_{rec} + \mathcal{L}_{kl} \\ &= -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{v},\mathbf{c})}[\log(p_\theta(\mathbf{v}|\mathbf{z},\mathbf{c}))] + KL[q_\phi(\mathbf{z}|\mathbf{v},\mathbf{c})||p_\theta(\mathbf{z}|\mathbf{c})],\end{aligned} \tag{12}$$

where the $q_\phi(\mathbf{z}|\mathbf{v},\mathbf{c})$, $p_\theta(\mathbf{v}|\mathbf{z},\mathbf{c})$, and $p_\theta(\mathbf{z}|\mathbf{c})$ are posterior net, preference generator, and prior net, respectively. The first optimization objective is the reconstruction loss, and we implement it by minimizing the mean square error between $\mathbf{v}$ and $\mathbf{v}'$:

$$\mathcal{L}_{rec} = -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{v},\mathbf{c})}[\log(p_\theta(\mathbf{v}|\mathbf{z},\mathbf{c}))] = \frac{1}{M}\frac{1}{d_v}\sum_{i=0}^{M}\sum_{j=0}^{d_v}(\mathbf{v}_{ij} - \mathbf{v'}_{ij})^2, \tag{13}$$

where $\mathbf{v}_{ij}$ and $\mathbf{v'}_{ij}$ denote value of the j^{th} dimension of the representation $\mathbf{v}_i$ and $\mathbf{v'}_i$. The second term is in charge of measuring the KL-divergence between the prior distribution $p_\theta(\mathbf{z}|\mathbf{c})$ and the approximate posterior $q_\phi(\mathbf{z}|\mathbf{v},\mathbf{c})$. It is implemented as follows:

$$\mathcal{L}_{kl} = KL[q_\phi(\mathbf{z}|\mathbf{v},\mathbf{c})||p_\theta(\mathbf{z}|\mathbf{c})] = \frac{1}{M}\sum_{i=0}^{M} KL[\mathcal{N}(\mu_i', \sigma_i'^2)||\mathcal{N}(\mu, \sigma^2)]. \tag{14}$$

Uniformity Loss. Since ELBO has no constraints on the distribution of different contents in the latent space, this may lead to insufficient discrimination of samples from the latent space. To alleviate this limitation, we propose making the centers of latent spaces more uniform, which can make different latent spaces more distinguishable and keep their unique information. Our uniformity loss is formulated as:

$$\mathcal{L}_{uni} = log \mathop{\mathbb{E}}_{i,j\in V} e^{-2\|\mu_i - \mu_j\|^2}/2. \tag{15}$$

The grey rectangle in Figure 1 visually shows the role of our goal. $\mathcal{L}_{uni}$ aims to ensure that latent spaces are distinguishable. It makes the latent variable sampled belong to a definite latent space, rather than belonging to several latent spaces at the same time. The upper right corner of the preference reconstruction module in Figure 1 visually shows the role of the Uniformity optimization goals. The pentagram represents the latent variable $\mathbf{z}$, and the uniformity optimization makes $\mathbf{z}$ belong to a certain latent space and not be covered by more than one at the same time.

In summary, the final optimization object of the preference reconstruction module is as follows:

$$\mathcal{L} = \mathcal{L}_{ELBO} + \alpha\mathcal{L}_{uni}, \tag{16}$$

herein, α is used to adjust the weight of uniformity loss. Through optimization, well-trained parameters can parameterize the distributions $q_\phi(\mathbf{z}|\mathbf{v},\mathbf{c})$, $p_\theta(\mathbf{z}|\mathbf{c})$, $p_\theta(\mathbf{v}|\mathbf{z},\mathbf{c})$. The overall training process of *GoRec* is shown in Algorithm 1.

Algorithm 1: The Training of *GoRec*

Input: Pre-trained item preference representation $\mathbf{V}$;
Output: Parameters ψ, $\mathbf{W}_{\mu}$,$\mathbf{b}_{\mu}$,$\mathbf{b}_{\sigma}$,$\mathbf{b}_{\sigma}$ and $\mathbf{W}_{\mu'}$,$\mathbf{b}_{\mu'}$,$\mathbf{b}_{\sigma'}$,$\mathbf{b}_{\sigma'}$;

```
Randomly initialize all parameters;
Randomly mask ρ content representation (Eqn.7);
while not converged do
    Sample a batch of pre-trained item preference representation;
    Calculate prior latent space μ_i and σ_i (Eqn.8);
    Calculate latent space μ'_i and σ'_i (Eqn.9);
    Sample latent variable z by reparameterization (Eqn.10);
    Reconstructed pre-trained item preference representation (Eqn.11);
    Compute reconstruction loss L_rec (Eqn.13);
    Compute KL loss L_kl (Eqn.14);
    Compute uniformity loss L_uni (Eqn.15);
    Update all parameters according to (Eqn.16);
end while
Return ψ, W_μ,b_μ,b_σ,b_σ and W_μ',b_μ',b_σ',b_σ'.
```

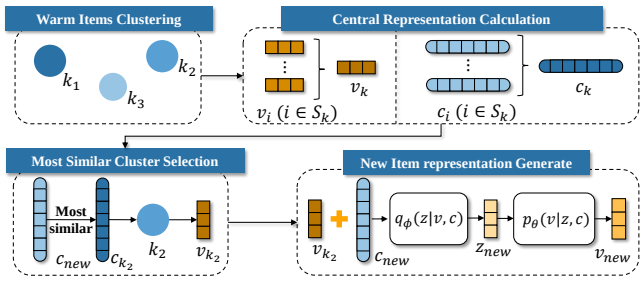


Figure 2: Generation of preference representations of new items.

3.3 New Item Recommendation Process

Given the learned parameters, we can generate preference representations that are related to specific content features. When a new item is coming, a naive CVAE model samples a latent variable $\mathbf{z}_{new}$ from the prior distribution $p(\mathbf{z}|\mathbf{c})$ and then $\mathbf{z}_{new}$ and content representation $\mathbf{c}_{new}$ are used to generate preference representation by $p_\theta(\mathbf{v}|\mathbf{z},\mathbf{c})$. In *GoRec*, we try to provide more information to locate a more accurate latent space using $q_\phi(\mathbf{z}|\mathbf{v},\mathbf{c})$. We propose a cluster-aware item approach to get a new item's fuzzy preference representation input into *GoRec*to generate the final preference representation $\mathbf{v}_{new}$.

Calculation of central representation. We use the K-Means [16] algorithm to cluster old items into k classes according to content representations. For each cluster, we calculate two central representations, the preference central representation $\mathbf{v}_k$, and the content central representation $\mathbf{c}_k$, which can be represented as:

$$\mathbf{v}_k = \frac{1}{N_k}\sum_{i\in S_k}\mathbf{v}_i, \quad \mathbf{c}_k = \frac{1}{N_k}\sum_{i\in S_k}\mathbf{c}_i, \tag{17}$$

where N_k is the number of items in the k^{th} cluster, and S_k is the set of old items that belong to the k^{th} cluster.

Selection of the most similar cluster. When a new item arrives, we first calculate the distance between new item's content representation $\mathbf{c}_{new}$ and each content central representation $\mathbf{c}_k$. Then we select a cluster k_{new} with the smallest distance, which can be represented as:

$$k_{new} = \arg\min_k \|\mathbf{c}_{new} - \mathbf{c}_k\|_2. \tag{18}$$

Generation of the new item representation. $\mathbf{v}_{k_{new}}$ denote the preference central representation of cluster k_{new}. Then we construct latent space according to $q_\phi(\mathbf{z}|\mathbf{v},\mathbf{c})$ as follows:

$$\mu'_{new} = [\mathbf{v}_{k_{new}};\mathbf{c}_{new}]\cdot\mathbf{W}_{\mu'} + \mathbf{b}_{\mu'}, \tag{19}$$

$$\sigma'_{new} = [\mathbf{v}_{k_{new}};\mathbf{c}_{new}]\cdot\mathbf{W}_{\sigma'} + \mathbf{b}_{\sigma'}, \tag{20}$$

$$\mathbf{z}_{new} = \mu'_{new} + \sigma'_{new}\odot\epsilon. \tag{21}$$

Final, the preference representation $\mathbf{v}_{new}$ of new item can be generate by $p_\theta(\mathbf{v}|\mathbf{z},\mathbf{c})$ as follows:

$$\mathbf{v}_{new} = f_\psi([\mathbf{z}_{new};\mathbf{c}_{new}]). \tag{22}$$

Finally, generated $\mathbf{v}_{new}$ is directly fed to the optimal recommendation backbone $\mathcal{F}^*_\pi$ to get the probability of users preferring new items, as described in Eqn. 2.

Table 1: The statistics of datasets.

Dataset		Baby	Clothing	Sports
Train	# Users	19442	39384	35592
	# Items	5640	18427	14686
	# Interactions	128963	222759	237111
	Density	0.00118	0.00031	0.00045
Val	# Users	10342	19801	18578
	# New Items	705	2303	1836
Test	# Users	10474	19858	19876
	# New Items	705	2303	1835

4 EXPERIMENT

In this section, we conduct extensive experiments on three real-world datasets, which aim to answer the following questions:

- **RQ1**: How does our model perform compared with other state-of-the-art baselines on multimedia-based cold-start recommendation scenarios?
- **RQ2**: How do different designed components play roles in our proposed model?
- **RQ3**: How does the pre-trained preference representation affect new item recommendation?
- **RQ4**: How do different hyper-parameters influence recommendation performances of the proposed model?

4.1 Experimental Settings

4.1.1 Datasets Description. We conduct experiments on three widely used Amazon datasets introduced by McAuley et al.[17]: (a) Clothing, Shoes, and Jewelry, (b) Sports and Outdoors, and (c) Baby. To simplify reading. The three datasets have rich multimedia information, including images and texts for each item. We use the extracted visual and textual features [41]. We randomly select 20% items and delete their historical interactions in the training process. Among them, we further divide half as validation and the remaining as test items. The statistics of the pre-processed datasets are summarized in Table 1.

Table 2: Performance comparisons with different Top-N values on three datasets.

Metric	Models	Baby				Clothing				Sports			
		K=10	K=20	K=30	K=40	K=10	K=20	K=30	K=40	K=10	K=20	K=30	K=40
Recall@K	KNN	0.0174	0.0300	0.0418	0.0518	0.0209	0.0354	0.0479	0.0599	0.0206	0.0321	0.0443	0.0545
	DUIF	0.0220	0.0410	0.0601	0.0733	0.0272	0.0457	0.0621	0.0770	0.0231	0.0417	0.0577	0.0720
	DropoutNet	0.0118	0.0244	0.0349	0.0453	0.0142	0.0263	0.0373	0.0475	0.0132	0.0238	0.0338	0.0424
	MTPR	0.0195	0.0390	0.0561	0.0725	0.0179	0.0334	0.0459	0.0586	0.0189	0.0356	0.0496	0.0645
	Heater	0.0232	0.0386	0.0541	0.0680	0.0384	0.0640	0.0842	0.1018	0.0390	0.0625	0.0828	0.1005
	CLCRec	0.0289	0.0497	0.0687	0.0880	0.0378	0.0629	0.0848	0.1026	0.0360	0.0588	0.0756	0.0912
	GAR	0.0299	0.0502	0.0686	0.0849	0.0393	0.0634	0.0851	0.1028	0.0385	0.0651	0.0876	0.1074
	CVAE	0.0261	0.0439	0.0596	0.0727	0.0435	0.0710	0.0932	0.1097	0.0399	0.0663	0.0894	0.1095
	GoRec-VBPR	0.0313	0.0507	0.0707	0.0852	0.0433	0.0713	0.0908	0.1084	0.0402	0.0657	0.0858	0.1027
	GoRec-VLGCN	0.0288	0.0493	0.0692	0.0843	0.0416	0.0670	0.0872	0.1071	0.0388	0.0679	0.0911	0.1108
	GoRec-VSGCL	**0.0456**	**0.0703**	**0.0915**	**0.1082**	**0.0575**	**0.0929**	**0.1168**	**0.1364**	**0.0515**	**0.0828**	**0.1094**	**0.1309**
NDCG@K	KNN	0.0103	0.0137	0.0165	0.0186	0.0126	0.0165	0.0194	0.0219	0.0126	0.0158	0.0187	0.0208
	DUIF	0.0112	0.0164	0.0208	0.0237	0.0144	0.0186	0.0233	0.0264	0.0119	0.0170	0.0208	0.0239
	DropoutNet	0.0056	0.0091	0.0116	0.0138	0.0073	0.0107	0.0132	0.0154	0.0075	0.0105	0.0128	0.0147
	MTPR	0.0103	0.0157	0.0198	0.0233	0.0089	0.0131	0.0160	0.0187	0.0100	0.0146	0.0179	0.0212
	Heater	0.0120	0.0162	0.0198	0.0227	0.0199	0.0269	0.0315	0.0352	0.0214	0.0289	0.0338	0.0376
	CLCRec	0.0168	0.0225	0.0270	0.0311	0.0199	0.0257	0.0333	0.0370	0.0214	0.0277	0.0317	0.0351
	GAR	0.0160	0.0216	0.0260	0.0295	0.0209	0.0275	0.0325	0.0362	0.0224	0.0288	0.0341	0.0384
	CVAE	0.0137	0.0187	0.0223	0.0251	0.0238	0.0288	0.0363	0.0398	0.0223	0.0296	0.0350	0.0394
	GoRec-VBPR	0.0171	0.0225	0.0272	0.0303	0.0218	0.0299	0.0340	0.0377	0.0229	0.0299	0.0346	0.0382
	GoRec-VLGCN	0.0164	0.0221	0.0268	0.0300	0.0226	0.0294	0.0341	0.0383	0.0213	0.0293	0.0348	0.0390
	GoRec-VSGCL	**0.0272**	**0.0341**	**0.0391**	**0.0427**	**0.0308**	**0.0395**	**0.0459**	**0.0500**	**0.0290**	**0.0376**	**0.0439**	**0.0485**

4.1.2 Evaluation Metrics. We select two metrics that are widely used in personalized recommender systems: Recall (Recall@K) and Normalized Discounted Cumulative Gain (NDCG@K).

4.1.3 Baselines. The baseline models can be divided into several categories: (1) content-based methods, KNN [22], DUIF [7]. (2) robustness-based methods, DropoutNet [28], MTPR [6]. (3) constraint-based methods, Heater [45], CLCRec [33]. (4) generative methods, GAR [3], CVAE.

- **KNN** [22] and **DUIF** [7] do not involve explicit alignments. The former exploits content similarity and the latter learns user preferences according to the item's content features.
- **DropoutNet** [28] and **MTPR** [6] strategically discard partial preference information in the training stage to simulate the cold-start condition.
- **Heater** [45] and **CLCRec** [33] design explicit alignment functions. The former uses the sum squared error loss to align pre-trained preference representation and content representation and the latter uses contrastive learning.
- **GAR** [3] trains a generator and a recommender adversarially, making the cold representation that is similar to the old representation. **CVAE** uses a naive CVAE model to reconstruct pre-trained preference representations.

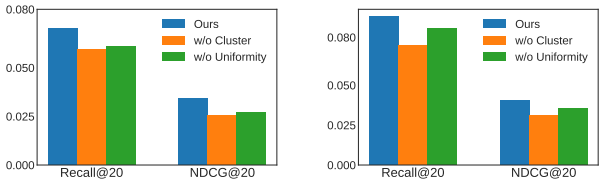
As *GoRec* can collaborate with any embedding-based recommendation model and benefit from it, we select three representative recommendation backbones to implement *GoRec*. Specifically, we adopt matrix factorization (VBPR [10]), graph learning (LightGCN [11] with multimedia features, denoted by VLGCN), and self-supervised graph learning based (SimGCL [40] with multimedia features, denoted by VSGCL) recommenders as backbones to achieve pre-trained preference representations.

4.1.4 Hyper-Parameter Settings. We implement our *GoRec* and all baselines with Pytorch framework. The dimension of preference representation is fixed as 64. The batch size is set to 256. The number of layers of the MLP is set to 2. During training, we employ Adam [12] as the optimizer and set the learning rate at 0.001, the early stop strategy is employed to avoid over-fitting. For our *GoRec* model, we turn the clustering number k from 200 to 3000. Besides, we carefully search the best parameter of α and find *GoRec* achieves the best performance when $\alpha = 15$ on Baby, $\alpha = 5$ on Clothing, and $\alpha = 15$ on Sports dataset. For all baselines, we search the parameters carefully for fair comparisons. We repeat all experiments 5 times and report the average results.

4.2 Overall Comparisons (RQ1)

As shown in Table 2, we compare our model with other baselines on three datasets. We have the following observations:

- By reconstructing the preference representation from SOTA recommendation methods, *GoRec* shows a significant improvement over all baselines. Specifically, *GoRec*-VSGCL improves the strongest baseline $w.r.t$ NDCG@20 by 51.26%, 29.39%, and 32.32% on Baby, Clothing, and Sports dataset, respectively. Extensive empirical studies verify the effectiveness of the proposed *GoRec* .
- With the exception of DUIF, all baseline models use the same pre-trained representation as *GoRec*-VSGCL. We observed that *GoRec*-VSGCL outperformed these baselines on all three datasets. For example, compared with GAR and Heater, *GoRec*-VSGCL improves $w.r.t$ NDCG@20 by 57.78% and 111.24% on Baby dataset, respectively. This suggests that



(a) Baby dataset (b) Clothing dataset

Figure 3: Ablation experiments.

Table 3: Running time (x in the brackets represents times).

Models	Time(s)	Epoch	Total Time(s)
GoRec	0.7	27	18.9
GAR [1]	4.6 (6.5x)	26	119.6 (6.3x)
DUIF [2]	22.6 (32.2x)	41	926.6 (49.0x)
Heater [3]	3.8 (5.4x)	15	57 (3.0x)
CLCRec	7.6 (10.8x)	7	53 (2.8x)
MTPR [4]	110.1 (157.2x)	72	7927.2 (419.4x)
DropoutNet [5]	4400 (6285.7x)	5	22000 (1164.0x)

GoRec can inherit preference information from pre-trained representations better than others.

- The CVAE model in baseline uses naive CVAE to reconstruct pre-trained preference representations. In many cases, naive CVAE surpasses other new item recommendation methods. This demonstrates the advantage of the framework that generates representations for new items by reconstructing pre-trained representations. Meanwhile, compared with naive CVAE, *GoRec* shows obvious improvement in all datasets, which indicates that our novel design on CVAE is effective.

4.3 Ablation Study (RQ2)

To exploit the effectiveness of each component of the proposed *GoRec* , we conduct the ablation study on different datasets. As shown in Figure 3, we compare *GoRec*-VSGCL and corresponding variants on Top-20 recommendation performance. GoRec-*w/o Cluster* denotes that remove cluster-aware approach to obtain the item's fuzzy pre-trained representations and sample latent variable according to prior net. GoRec-*w/o Uniformity* denotes that remove the uniformity-enhanced optimization. From Figure 3, we observed that each component of the *GoRec* contributed to the final superior performance. The cluster-aware approach and optimization of uniformity of μ learn and locate more distinguishable latent space.

In addition, referring to [40], we plot latent variable distributions with Gaussian kernel density estimation (KDE) in $\mathbb{R}^2$ and KDE on angles $\mathcal{S}^1$. As shown in Figure 4, we can observe that uniformity optimization results in a more uniform distribution of latent variable, which make them more distinguishable.

[1]https://github.com/zfnWong/GAR
[2]https://github.com/duxy-me/MTPR
[3]https://github.com/Zziwei/Heater–Cold-Start-Recommendation
[4]https://github.com/duxy-me/MTPR
[5]https://github.com/layer6ai-labs/DropoutNet/tree/master/torch

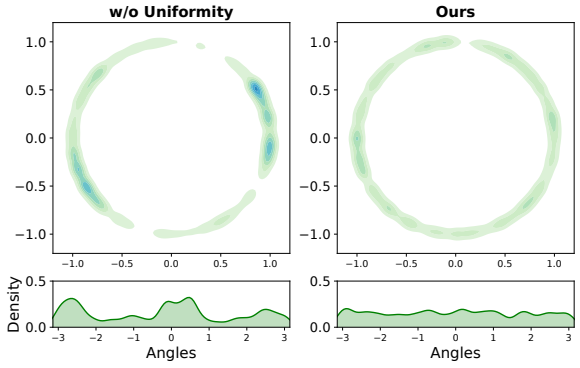


Figure 4: Uniformity of latent variable on Baby dataset.

4.4 Impact of Pre-trained Representations (RQ3)

As we introduced in methodology, our proposed *GoRec* can directly benefit from the PPM. Comparing the performance of *GoRec* with different pre-trained preference representations in Table 2, we can observe the relationship between *GoRec* performance and the quality of pre-trained preference representations. With the improvement of the performance of the pre-trained preference representations expression on the old item recommendation task, the performance of the corresponding *GoRec* model on the new item recommendation task also gradually improved.

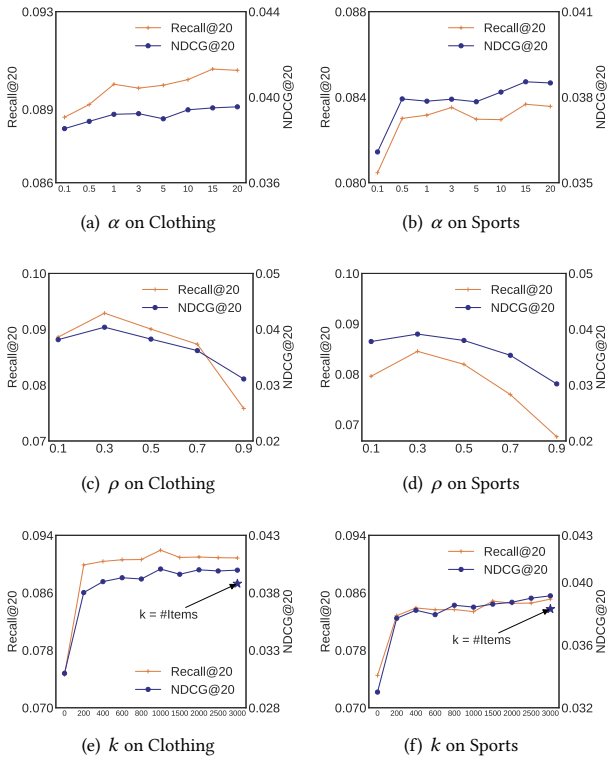
Besides, we reconstruct item preference representations instead of reconstructing historical interactions. This change in training mode saves training resources. We demonstrate the advantage of *GoRec* in running time through experiments. We report the real running time that the compared methods cost for one epoch and the total time that each model needs to achieve the best performance on the Clothing dataset. The results in Table 3 are collected on an Intel(R) Core(TM) i9-10900X CPU and a GTX TITAN X GPU. We calculate how many times slower the other methods are when compared with *GoRec*. Instead of reconstructing historical interaction records but reconstructing the preference representation, *GoRec* has an absolute advantage in running time.

4.5 Hyper-Parameter Sensitivities (RQ4)

In this part, we analyze the impact of hyper-parameters in *GoRec*. We exploit the effect of uniformity loss weight α, mask ratio ρ in training, and cluster number k.

Effect of Uniformity Loss Weights α. As illustrated in Figure 5(a) and Figure 5(b), we carefully tune the uniformity loss weights α on the Clothing and Sports datasets. We observe that *GoRec* achieves the best performance when $\alpha = 15$ on both the Clothing and Sports datasets. The performance increases first and then stops increasing. It indicates the latent space is already distinguishable and stronger uniformity constraints can no longer provide valuable help.

Effect of Mask ratio ρ. As introduced in the previous works, we can mask part of content representations to enhance the generalization ability of *GoRec* . From Figure 5(c) and Figure 5(d), we can obverse that a proper mask ratio can improve model performance,



(a) α on Clothing (b) α on Sports

(c) ρ on Clothing (d) ρ on Sports

(e) k on Clothing (f) k on Sports

Figure 5: Performance of different hyperparameters.

however, too high mask ratio lose too much valuable information, resulting in a poor model effect.

Effect of Cluster Number k. To investigate the effect of cluster numbers, we set different cluster numbers in the new item recommendation process. We illustrate the experimental results in Figure 5(e) and Figure 5(f). Experiments show that when the number of clusters increases, we can provide more accurate information to construct latent space and further improve performance. However, when the number of clusters is too large, the fuzzy preference will be more clear but may be inconsistent with the actual preference of the new item, thus reducing the performance of the model, such as the point represented by the pentagram in the figure.

5 RELATED WORK

5.1 Cold-start Recommendation

Due to the different data distributions between warm and cold items, recommendation models are hard to generate effective new items representation. Existing cold-start methods can be broadly divided into two categories, robustness-based methods, and enhanced-based methods. Robustness-based methods randomly drop CF signals in the training stage to simulate the scenario of new item recommendation [6, 28]. For instance, DropoutNet [28] and MTPR [6] strategically discard CF information during the training phase to simulate the cold-start scenario. Enhanced-based methods use the pre-trained representation to provide additional CF signals to enhance content representation. These methods dedicate to designing various functions to model the correlation and narrow the difference between CF information and content features [3, 26, 32, 33, 38, 45]. For example, Heater [45] extracts content representations and uses the sum squared error loss to align pre-trained CF representations and content representations. However, these methods focus on the use of content features to express preferences and make insufficient use of CF information. Besides, some multimedia recommendation methods can alleviate the cold-start problem [19, 41]. Unlike our problem scenario, these methods usually require that the content information of a new item be available at the training stage in order to model the correlation between items.

5.2 Applications of VAEs on Recommendation

Variational Auto-Encoder (VAE) is a generative method widely used in machine learning [13, 21]. It assumes that the input data can be generated from variables with some probability distribution. VAEs are widely used in recommendation systems to reconstruct users' interactions with items. Mult-VAE [14] models user-item interactions using multinomial distribution and parameterizes users with neural networks. RecVAE [24] introduces a composite prior distribution for the latent encoder. BiVAE [27] proposes bilateral inference models to estimate the user-item and item-user distributions. CVGA [42] combines GNNs and VAEs to reconstruct the user-item bipartite graph using variance inference. CVAE is a variation of VAE that can generate samples based on some specific conditions [25]. In other words, it is a generative model that can condition its output on additional information, such as class labels or other relevant side information. CVAE is widely used in generative tasks in natural language processing and computer vision [15, 29, 39, 43]. However, there are few studies about the application of CVAE in recommender systems.

6 CONCLUSION

In this paper, we focus on the recommendation for new items. Different from the previous work, we propose modeling the item preference distribution conditioned on multimedia content. Our model is better able to inherit valuable information from preference representations. And greatly save the cost of training. Specifically, under the guidance of multimedia content features, we use a CVAE model to reconstruct pre-trained preference representations and generate preference representations for new items. Besides, we propose a novel uniformity-enhanced optimization to make different latent spaces more distinguishable and keep their unique information. Empirical studies on three public datasets clearly show the effectiveness of the proposed framework.

ACKNOWLEDGMENTS

This work was supported in part by grants from the National Key Research and Development Program of China (Grant No. 2021ZD0111802), the National Natural Science Foundation of China (Grant No. 72188101, 61972125, 62006066, U19A2079 U22A2094), and Major Project of Anhui Province (Grant No. 202203a05020011).

REFERENCES

[1] Iman Barjasteh, Rana Forsati, Dennis Ross, Abdol-Hossein Esfahanian, and Hayder Radha. 2016. Cold-Start Recommendation with Provable Guarantees: A Decoupled Approach. *TKDE* (2016).

[2] Oren Barkan, Noam Koenigstein, Eylon Yogev, and Ori Katz. 2016. CB2CF: a neural multiview content-to-collaborative filtering model for completely cold item recommendations. *RecSys* (2016).

[3] Hao Chen, Zefan Wang, Feiran Huang, Xiao Huang, Yue Xu, Yishi Lin, Peng He, and Zhoujun Li. 2022. Generative Adversarial Framework for Cold-Start Item Recommendation. *SIGIR* (2022).

[4] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting Graph based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach. *AAAI* (2020).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL* (2019).

[6] Xiaoyu Du, Xiang Wang, Xiangnan He, Zechao Li, Jinhui Tang, and Tat-Seng Chua. 2020. How to Learn Item Representation for Cold-Start Multimedia Recommendation? *MM* (2020).

[7] Xue Geng, Hanwang Zhang, Jingwen Bian, and Tat-Seng Chua. 2015. Learning Image and User Features for Recommendation in Social Networks. *ICCV* (2015).

[8] Casper Hansen, Christian Hansen, Jakob Grue Simonsen, Stephen Alstrup, and Christina Lioma. 2020. Content-aware Neural Hashing for Cold-start Recommendation. *SIGIR* (2020).

[9] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CVPR* (2015).

[10] Ruining He and Julian McAuley. 2015. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. *AAAI* (2015).

[11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. *SIGIR* (2020).

[12] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *ICLR* (2014).

[13] Diederik P. Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *ICLR* (2013).

[14] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. *WWW* (2018).

[15] Fenglong Ma, Yaliang Li, Chenwei Zhang, Jing Gao, Nan Du, and Wei Fan. 2019. MCVAE: Margin-based Conditional Variational Autoencoder for Relation Classification and Pattern Generation. *WWW* (2019).

[16] J. MacQueen. 1967. Some methods for classification and analysis of multivariate observations.

[17] Julian McAuley, Christopher Targett, Javen Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. *SIGIR* (2015).

[18] Kaixiang Mo, Bo Liu, Lei Xiao, Yong Li, and Jie Jiang. 2015. Image Feature Learning for Cold Start Problem in Display Advertising. *IJCA* (2015).

[19] Zongshen Mu, Yueting Zhuang, Jie Tan, Jun Xiao, and Siliang Tang. 2022. Learning Hybrid Behavior Patterns for Multimedia Recommendation. *MM* (2022).

[20] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. *UAI* (2009).

[21] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *ICML* (2014).

[22] Suvash Sedhain, Scott Sanner, Darius Braziunas, Lexing Xie, and Jordan Christensen. 2014. Social collaborative filtering for cold-start recommendations. *RecSys* (2014).

[23] Vikas Sethi and Rajneesh Gujral. 2022. Survey of Different Recommendation Systems to Improve the Marketing Strategies on E-commerce. *ICML* (2022).

[24] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. 2020. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. *WSDM* (2020).

[25] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. *NeurIPS* (2015).

[26] Changfeng Sun, Han Liu, Meng Liu, Zhaochun Ren, Tian Gan, and Liqiang Nie. 2020. LARA: Attribute-to-feature Adversarial Learning for New-item Recommendation. *ICDM* (2020).

[27] Quoc-Tuan Truong, Aghiles Salah, and Hady W Lauw. 2021. Bilateral variational autoencoder for collaborative filtering. *WSDM* (2021).

[28] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing Cold Start in Recommender Systems. *NeurIPS* (2017).

[29] Jacob Walker, Carl Doersch, Abhinav Kumar Gupta, and Martial Hebert. 2016. An Uncertain Future: Forecasting from Static Images Using Variational Autoencoders. *ECCV* (2016).

[30] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. *CIKM* (2018).

[31] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet A. Orgun. 2019. Sequential Recommender Systems: Challenges, Progress and Prospects. *IJCAI* (2019).

[32] Shuai Wang, Kun Zhang, Le Wu, Haiping Ma, Richang Hong, and Meng Wang. 2021. Privileged Graph Distillation for Cold Start Recommendation. *SIGIR* (2021).

[33] Yin wei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive Learning for Cold-Start Recommendation. *MM* (2021).

[34] Yin wei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-Refined Convolutional Network for Multimedia Recommendation with Implicit Feedback. *MM* (2020).

[35] Yin wei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal Graph Convolution Network for Personalized Recommendation of Micro-video. *MM* (2019).

[36] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2021. A Survey on Accuracy-Oriented Neural Recommendation: From Collaborative Filtering to Information-Rich Recommendation. *TKDE* (2021).

[37] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2020. DiffNet++: A Neural Influence and Interest Diffusion Network for Social Recommendation. *TKDE* (2020).

[38] Le Wu, Yonghui Yang, Lei Chen, Defu Lian, Richang Hong, and Meng Wang. 2020. Learning to Transfer Graph Embeddings for Inductive Graph based Recommendation. *SIGIR* (2020).

[39] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. 2015. Attribute2Image: Conditional Image Generation from Visual Attributes. *ECCV* (2015).

[40] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Li zhen Cui, and Quoc Viet Hung Nguyen. 2021. Are Graph Augmentations Necessary?: Simple Graph Contrastive Learning for Recommendation. *SIGIR* (2021).

[41] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. 2021. Mining Latent Structures for Multimedia Recommendation. *MM* (2021).

[42] Yi Zhang, Yiwen Zhang, Dengcheng Yan, Shuiguang Deng, and Yun Yang. 2022. Revisiting Graph-based Recommender Systems from the Perspective of Variational Auto-Encoder. *TIST* (2022).

[43] Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders. *ACL* (2017).

[44] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to Warm Up Cold Item Embeddings for Cold-start Recommendation with Meta Scaling and Shifting Networks. *SIGIR* (2021).

[45] Ziwei Zhu, Shahin Sefati, Parsa Saadatpanah, and James Caverlee. 2020. Recommendation for New Users and New Items via Randomized Training and Mixture-of-Experts Transformation. *SIGIR* (2020).