A Survey on Variational Autoencoders in Recommender Systems

SHANGSONG LIANG, School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China
ZHOU PAN, School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China
WEI LIU, School of Artificial Intelligence, Sun Yat-Sen University, Zhuhai, China
JIAN YIN, School of Artificial Intelligence, Sun Yat-Sen University, Zhuhai, China
MAARTEN DE RIJKE, University of Amsterdam, Amsterdam, Netherlands

Recommender systems have become an important instrument to connect people to information. Sparse, complex, and rapidly growing data presents new challenges to traditional recommendation algorithms. To overcome these challenges, various deep learning-based recommendation algorithms have been proposed. Among these, Variational AutoEncoder (VAE)-based recommendation methods stand out. VAEs are based on a flexible probabilistic framework, which is robust for data sparsity and compatible with other deep learning-based models for dealing with multimodal data. In addition, the deep generative structure of VAEs helps to perform Bayesian inference in an efficient manner. VAE-based recommendation algorithms have given rise to many novel graphical models and they have achieved promising performance. In this paper, we conduct a survey to systematically summarize recent VAE-based recommendation algorithms. Four frequently used characteristics of VAE-based recommendation algorithms are summarized, and a taxonomy of VAE-based recommendation algorithms is proposed. We also identify future research directions for, advanced perspectives on, and the application of VAEs in recommendation algorithms, to inspire future work on VAEs for recommender systems.

CCS Concepts: • **Information systems** → **Data mining**; • **Human-centered computing** → *Collaborative and social computing*.

Additional Key Words and Phrases: variational autoencoder, recommender systems, deep learning, Bayesian network

1 INTRODUCTION

Recommender systems (RSs) help users to identify and connect valuable and interesting information and services. At the same time, RSs help platforms, in e-commerce, entertainment, news, etc., to expose items and services to potential users. Recommendation algorithms are designed to process multimodal data, such as item content, interaction data, as well as side information, such as user profiles, reviews, popularity, to recommend a single item or a ranked list of items to users [21, 140, 184].

Collaborative Filtering (CF)-based methods are frequently used in RS, since they are able to achieve good recommendation performance and be implemented easily, e.g., by matrix factorization [89, 119]. The core idea

*Corresponding author.

Authors' Contact Information: Shangsong Liang, School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China; e-mail: liangshangsong@gmail.com; Zhou Pan, School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China; e-mail: panzh8@mail2.sysu.edu.cn; wei liu, School of Artificial Intelligence, Sun Yat-Sen University, Zhuhai, China; e-mail: liuw259@mail.sysu.edu.cn; Jian Yin, School of Artificial Intelligence, Sun Yat-Sen University, Zhuhai, China; e-mail: issjyin@mail.sysu.edu.cn; Maarten de Rijke, University of Amsterdam, Amsterdam, Noord-Holland, Netherlands; e-mail: m.derijke@uva.nl.

ACM /2024/5-ART
https://doi.org/10.1145/3663364

behind CF is that users with similar preferences would like to consume similar items. However, these methods usually suffer from cold-start and data sparsity problems [20]. To tackle these problems, side information such as reviews, item content, user profiles, is combined with CF-based algorithms, forming hybrid recommendation methods. Examples include Collaborative Topic Regression [169], which uses Latent Dirichlet Allocation (LDA) [15] to learn latent representations of texts and Probabilistic Matrix Factorization (PMF) [119] to model user-item interaction data. These methods fail to cope with the increasingly multimodal nature of data associated with items and services to be recommended, such as image, video, GPS coordinates, etc. [100]. Moreover, the latent representations learned by LDA are not robust enough in scenarios where the data is extremely sparse [100, 171]. To tackle these problems, deep learning-based CF methods have been, and continue to be, explored for learning robust and non-linear latent representations from sparse and multimodal data.

Deep learning-based CF recommendation methods can be grouped based on the underlying methods they use:

- Restricted Boltzmann Machines (RBMs) [51, 128, 143],
- Deep Belief Networks (DBNs) [73, 137, 176],
- Autoencoders [11, 63, 146, 166, 167] and variational autoencoders [17, 30, 86, 139],
- Recurrent Neural Networks (RNNs) [70, 158],
- Convolutional Neural Networks (CNNs) [53, 92, 164].
- Attention/self-attention deep learning models [46, 49, 61, 124, 205],
- Deep reinforcement learning models [3, 56, 69, 181, 213], and
- Graph neural networks [39, 66, 150, 175, 198, 216].

Specific applications of these deep learning-based RSs can be found in recent surveys [10, 82, 185, 208].

1.1 Recommendation methods based on Variational AutoEncoders (VAEs)

Among the deep learning-based recommendation methods listed above, the ones based on VAEs [17, 30, 86, 139] stand out due to their unsupervised learning nature and their non-linear probabilistic latent-variable approaches [44], as they are more effective and robust to infer users' latent purchase preferences [218] and the latent semantics of the items. The key characteristics of VAEs can be summarized as follows:

(1) A VAE consists of a flexible encoder and decoder. Both can incorporate other deep learning methods, resulting in a *flexible internal structure*. Although many other designs have encoders, Gaussian priors, or allow for Bayesian inference, VAEs can be considered as a generalization of classical matrix factorization, PMF, factorization machines, SVD feature etc. In the framework of VAEs, the encoder can be implemented using many alternative designs, such as multilayer perceptrons, hierarchical stochastic units, gated linear units, etc. Moreover, by using multiple deep learning models as encoders, a VAE can integrate the learning ability of other special deep learning models; we refer to this as the *encoding capability* of VAEs in this paper.

(2) A VAE is a deep *generative* model that is able to learn implicit patterns in data and use *Bayesian* inference for model optimization. Bayesian inference is also the main difference between VAEs and traditional autoencoders [86, 146, 166]. Bayesian inference as used in VAEs is efficient because a VAE combines the graphical model and deep learning, and uses back propagation [68] to learn parameters with the help of the reparameterization trick [41, 86, 87]. Its Bayesian nature enables a VAE to design diverse graphical models to capture the causal relations between different variables in a RS, e.g., variables denoting users, items, or side information.

(3) A VAE consists of flexible priors and preference distributions. In addition to a Gaussian prior, the priors of VAEs can be extended to user-dependent priors [80], the VampPrior [83], or composite prior [149]. In classical matrix factorization models, their extensions and deep learning designs, it is usually assumed that the user preference distribution on items follows a Gaussian distribution. With VAEs and their Bayesian

nature, the user preference distribution can be assumed to be a multinomial distribution [102] or a Bernoulli distribution [28].

To sum up, there are four main characteristics that make VAEs attractive in the context of RSs: (i) their encoding capability (representation learning), (ii) their generative nature, (iii) their Bayesian nature, and (iv) their flexible internal structure.

1.2 Emergence of VAEs for RSs

As early as in 2017, Li and She [100] used VAEs to learn latent representations of item content (including multimedia items) to address the data sparsity and cold-start problem. In the subsequent year, Liang et al. [102] used a VAE to model the generative process of user-item interaction data so that the user's preference can be inferred. Lee et al. [93] explored graphical models designed with VAEs for handling different causal relations between variables in RSs. Meanwhile, VAEs can also facilitate explorations of other topics in RSs, such as novel item generation [168], fairness in RSs [16], multi-criteria recommendation [99]. Compared with [100], Liang et al. [102] demonstrated the practical utility of their VAE-based recommendation model by directly extracting user representations from interactions. In 2021, Borges and Stefanidis [17] proposed a VAEs-based recommendation method that penalizes scores given to items according to historical popularity for mitigating the bias and promoting diversity in the recommendation results. Subsequently, by adopting autoencoders as the base model, Choi et al. [30] proposed a Local Collaborative Autoencoders (LOCA) that aims at capturing latent non-linear patterns representing meaningful user-item interactions within sub-communities of the users. Zhang et al. [210] adapted the Wasserstein autoencoders [160] to address the CF problem. Hence, one can see that the number of research publications that apply VAEs in RSs has been increasing dramatically in the past few years.

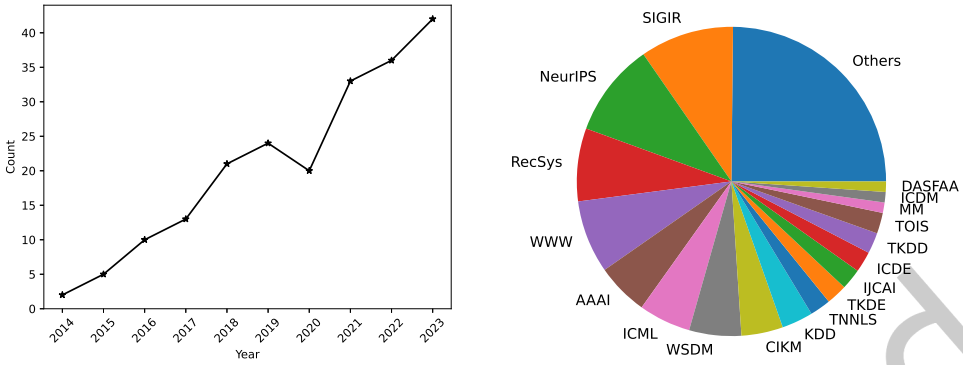
Given the advantages of using VAEs in RSs and given the proliferation of research that applies VAEs in this context, the time is right to conduct a comprehensive survey to systematically summarize the recommendation methods that apply VAEs and provide inspiration, in terms of model design and applications of VAEs, to other researchers who are interested in recommendation algorithms.

1.3 Differences between this survey and others

To the best of our knowledge, this paper is the first survey on applications of VAEs in RSs. Although there have been many publications that use VAEs to address specific problems in RSs or that attempt to improve recommendation performance, there is no in-depth summary of applications of VAEs in RSs.

Several surveys of traditional recommendation algorithms [20, 47, 81, 91, 155] and deep learning-based recommendation algorithms [10, 14, 82, 106, 204, 208] have been published in recent years. Surveys on traditional recommendation algorithms and on deep learning-based recommendation algorithms [14, 82, 106] seldom mention VAE-based recommendation methods. Zhang et al. [208], Batmaz et al. [10] and Khan et al. [82] reviewed publications on deep learning for RSs, but they only introduce a limited number of recommendation methods that use a VAE without providing further analysis. The most similar survey to ours may be the one by Zhang et al. [204], who summarized autoencoder-based recommendation methods, including a small number of VAE-based recommendation methods. Given the advantages of VAEs among deep learning-based recommendation methods and given the lack of surveys on VAEs for RSs, we believe it is valuable to survey the applications of VAEs in RSs from the aspect of research and applications.

Below, we summarize several characteristics of VAEs that make a VAE applicable to RSs. Based on these characteristics we analyze current works on applying VAEs to RSs. We also describe several potential future research directions for, and new perspectives on, VAEs for RSs.



(a) (b)

Fig. 1. Distributions of the reviewed papers of VAE-based recommendation methods over years (a) and over venues (b).

1.4 Contributions of this survey

The contributions of this survey can be summarized as follows:

(1) More than 100 research papers on VAEs and especially on applying VAEs for recommendation are collected. In conducting our literature search, we use DBLP and Google Scholar as the primary search engines with the keywords: *variational* + *recommendation*, *variational autorencoder* + *recommendation*, *VAE* + *recommendation*, and *generative* + *recommendation* (noting that recommendation is also replaced with collaborative filtering) in the title and the abstracts, to filter the related papers from the top-tier data mining/machine learning/recommendation system/artificial intelligence conferences and journals including but not limited to: KDD, ICDE, WWW, SIGIR, ICML, NeurPIS, AAAI, IJCAI, WSDM, CIKM, RecSys, TKDE, TNNLS, TOIS and TKDD, ranging from 2003 to 2023 and mainly focus on the most recent 6 years. We then traverse the citation graph of the identified papers and incorporate pertinent studies. In addition to reviewing published papers, we screen preprints on arXiv and identify those with novel and intriguing ideas for a more comprehensive picture. Four characteristics of VAEs used in current RSs are summarized, based on which an in-depth analysis of the surveyed work is given to help researchers understand the specific application and benefits of VAEs in RS;
(2) A taxonomy of the work that we survey to help recognize trends in applications of VAEs in RS; and
(3) Future research directions are pointed out, based on our discussion as well as some other advanced perspectives of VAE in RS, aiming to attract more researchers to engage in this research field.

The remainder of this survey is organized as follows. Preliminaries are introduced in Section 2. Section 3 presents our analysis of traditional recommendation methods (e.g., CF) that apply VAEs and our taxonomy for them, and discusses how VAEs are applied to other topics in the context of RSs. Section 4 describes possible future research directions and new perspectives. Finally, Section 5 concludes the survey.

2 PRELIMINARIES

To aid in the presentation of the surveyed papers, we first define the notation and basic concepts that we use in the paper.

Table 1. Main notation used in the paper.

Symbols	Descriptions
M, N	number of users, items
$\mathbf{r}_i^u, \mathbf{r}_j^v$	user i's, item j's interaction vector
$\mathbf{x}_i^u, \mathbf{x}_j^v$	user i's, item j's side information (features)
$\mathbf{z}_i^u$	latent variable of user i's interaction vector $\mathbf{r}_i^u$
$\mathbf{z}_j^v$	latent variable of item j's interaction vector $\mathbf{r}_j^v$
$\mathbf{h}_i^u$	latent variable of user i's side information $\mathbf{x}_i^u$
$\mathbf{h}_j^v$	latent variable of item j's side information $\mathbf{x}_j^v$
$\mathbf{u}_i, \mathbf{U}$	latent variable of user i, and corresponding matrix
$\mathbf{v}_j, \mathbf{V}$	latent variable of item j, and corresponding matrix
$\mathbf{v}_j^\dagger$	collaborative latent variable of item j
$\mathbf{R}$	rating matrix or interaction matrix
$\mathbf{R}_{ij}$	i, j-th element of feedback matrix $\mathbf{R}$
$\mathbf{g}_i$	social factor of user i
$\mathbf{S}_{ik}$	i, k-th element of social matrix
$\boldsymbol{\phi}, \mathbf{W}_{inf}$	parameters of the encoder of a VAE
$\boldsymbol{\theta}, \mathbf{W}_{gen}$	parameters of the decoder of a VAE
$\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$	mean, variance of Gaussian distribution
$\boldsymbol{\epsilon}$	random variable drawn from $\mathcal{N}(0, I)$
β, α	hyper parameter for regularization in a VAE
$\lambda_u, \lambda_v, \lambda_g, \lambda_q$	weights for corresponding vector regularization
$D_{KL}(\cdot\|\cdot)$	KL divergence between two distributions
$\mathbf{o}^{(t)}$	hidden state in RNN at time t
$\mathbf{v}^{(t)}$	user's interaction item at time t
$\mathbf{z}^{(t)}$	latent variable of $\mathbf{v}^{(t)}$

2.1 Notations

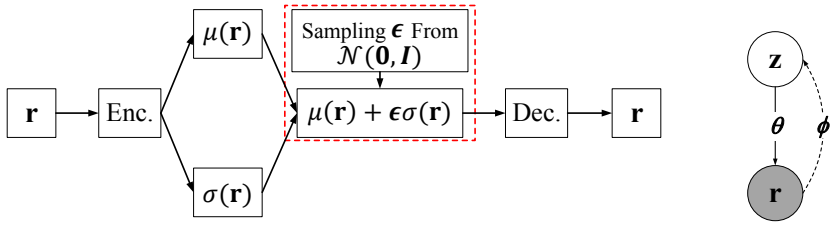
We use M and N to denote the number of users and items, respectively. We use $i \in \{1, 2, \ldots, M\}$ and $j \in \{1, 2, \ldots, N\}$ to denote the i-th user and the j-th item. User-item interaction/rating data is denoted using $\mathbf{R} \in \mathbb{R}^{M \times N}$, where $\mathbf{R}_{ij}$ denotes the preference[1] of user i to item j. For latent factor models, let $\mathbf{U} \in \mathbb{R}^{M \times k}$ and $\mathbf{V} \in \mathbb{R}^{N \times k}$ represent the user latent factor matrix and item latent factor matrix, respectively, where k is the dimension of the latent factors. Side information of users/items is denoted by $\mathbf{X}$, with the row vector $\mathbf{x}$ describing an arbitrary user's/item's side information. Further notation will be introduced when necessary in the corresponding sections; notation used throughout the paper is summarized in Table 1.

2.2 Basics of VAEs

To better understand VAEs and their applications in recommender systems, in this subsection, we first introduce vanilla VAEs and some related concepts based on the recommendation scenario, and then detail the characteristics of VAEs.

2.2.1 The vanilla VAE. The vanilla VAE is a generative model that can generate samples similar to those in the training datasets. It is called an "autoencoder" because it has an encoder (inference module) and a decoder

[1]Preferences can be measured by rating scores for items from users.



(a) (b)

Fig. 2. (a) A vanilla VAE implemented by a feedforward neural network; inside the red dashed rectangle is an illustration of the reparameterization trick when the latent distribution is Gaussian. (b) Graphical model of a vanilla VAE for modeling user-item interaction data.

(generative module). The final training objective consists of an encoder and a decoder, which resembles traditional autoencoders [41]. During training, unlike traditional autoencoders, the encoder of a VAE produces the distribution of the latent variable for the input data, while the decoder reconstructs the input data from the dedicated latent representation sampled from the distribution. An illustration of the workflow of a vanilla VAE can be found in Figure 2(a).

For brevity and without causing confusion, we use $\mathbf{r}$ to represent an arbitrary user's interaction vector $\mathbf{r}_i^u$. We take modeling $\mathbf{r}$ as an example; the graphical model is shown in Figure 2(b); the encoder, decoder, objective and some related concepts are detailed below.

Encoder. The encoder of a VAE encodes the interaction data $\mathbf{r}$ into the approximate posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{r})$ (we refer to it as the "latent distribution" for convenience and as we will explain later, this distribution is used to approximate the true posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{r})$ of the latent variable $\mathbf{z}$). Specifically, taking $\mathbf{r}$ as input, a neural network with parameters $\boldsymbol{\phi}$ is used to parameterize the distribution of latent variable $\mathbf{z}$. For instance, if we use a Gaussian distribution to represent $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{r})$, then the encoder should output the mean and variance of $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{r})$. By using this encoder, for every $\mathbf{r}$, we can learn the dedicated latent distribution for it. Applying *amortized inference* [52] and neural networks to approximate the true distributions of latent variables are the cores of the encoder of a VAE and foundations for efficient Bayesian inference.

Decoder. The decoder, also known as the generative module, is used to generate data that is as close to the input data (of the encoder) as possible, i.e., to reconstruct the input data. With the samples of $\mathbf{z}$ generated from the latent distribution $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{r})$, the decoder can reconstruct the interaction data $\mathbf{r}$. Similar to the encoder, the decoder can also use a neural network with parameters $\boldsymbol{\theta}$ to do this reconstruction. Specifically, the output of the decoder, i.e., the preference probabilities over all items, is used to parameterize the distribution $p_{\boldsymbol{\theta}}(\mathbf{r} \mid \mathbf{z})$, from which the reconstructed $\mathbf{r}$ is sampled. In practice, we use the output probabilities over all the items to generate recommendations.

Objective. As is introduced in the decoder above, the interaction data $\mathbf{r}$ is generated from a latent variable $\mathbf{z}$ using the decoder. Then, modeling the generative process of $\mathbf{r}$ is equivalent to maximizing the following marginal probability of $\mathbf{r}$:

$$p_{\boldsymbol{\theta}}(\mathbf{r}) = \int p_{\boldsymbol{\theta}}(\mathbf{r}, \mathbf{z}) d\mathbf{z} = \int p_{\boldsymbol{\theta}}(\mathbf{r} \mid \mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{z}) d\mathbf{z}. \tag{1}$$

The above integral for computing the marginal likelihood $p_{\boldsymbol{\theta}}(\mathbf{r})$ does not have an analytic solution. Considering the Bayes' theorem $p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{r}) = p_{\boldsymbol{\theta}}(\mathbf{r}, \mathbf{z}) / p_{\boldsymbol{\theta}}(\mathbf{r})$ and the intractability of $p_{\boldsymbol{\theta}}(\mathbf{r})$, the true posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{r})$ is also intractable. To solve the problem, the latent distribution $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{r})$, parameterized by the encoder

neural network, is used to approximate the intractable $p_{\theta}(\mathbf{z} \mid \mathbf{r})$ and the Evidence Lower BOund (ELBO) is derived from Eq. (1) as:

$$\log p_{\theta}(\mathbf{r}) = D_{KL}(q_{\phi}(\mathbf{z} \mid \mathbf{r}) \| p_{\theta}(\mathbf{z} \mid \mathbf{r})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{r}). \tag{2}$$

The first term in the right hand side of Eq. (2) is the KL divergence[2] between the true posterior $p_{\theta}(\mathbf{z} \mid \mathbf{r})$ and $q_{\phi}(\mathbf{z} \mid \mathbf{r})$ and the second term is the ELBO on the marginal likelihood of $\mathbf{r}$. Since the KL divergence is non-negative, we have:

$$\log p_{\theta}(\mathbf{r}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{r}) = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{r})}\left[-\log q_{\phi}(\mathbf{z} \mid \mathbf{r}) + \log p_{\theta}(\mathbf{r}, \mathbf{z})\right], \tag{3}$$

where the ELBO $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{r})$ can be rewritten as:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{r}) = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{r})}[\log p_{\theta}(\mathbf{r} \mid \mathbf{z})] - D_{KL}(q_{\phi}(\mathbf{z} \mid \mathbf{r}) \| p_{\theta}(\mathbf{z})). \tag{4}$$

As in other variational methods, the ELBO is the optimization objective of the vanilla VAE. The complete derivation process of the ELBO is provided in [87]. The first term at the right hand side of Eq. (4) is the expectation of reconstructing $\mathbf{r}$ in terms of the latent distribution $q_{\phi}(\mathbf{z} \mid \mathbf{r})$, often called *negative reconstruction error*. The second term is the KL divergence between $q_{\phi}(\mathbf{z} \mid \mathbf{r})$ and the prior of the latent variable, i.e., $p_{\theta}(\mathbf{z})$.

Reparameterization trick. For the moment, we want to differentiate and optimize the ELBO w.r.t. both the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ in the encoder and decoder, respectively. Since the KL divergence term can often be calculated analytically, only the expectation w.r.t. $q_{\phi}(\mathbf{z} \mid \mathbf{r})$ in Eq. (4) requires estimation by sampling, i.e., the Monte Carlo estimate of this term should be made. However, the sampling of $\mathbf{z}$ from $q_{\phi}(\mathbf{z} \mid \mathbf{r})$ is a non-continuous operation and has no gradient, so the backpropagation can not be applied to update the parameters. To solve this problem, while training a VAE, we use the reparameterization trick to sample the latent variable $\mathbf{z}$ from the latent distribution $q_{\phi}(\mathbf{z} \mid \mathbf{r})$. The reparameterization trick makes the Monte Carlo estimate of the expectation w.r.t. $q_{\phi}(\mathbf{z} \mid \mathbf{r})$ in Eq. (4) differentiable by introducing an extra auxiliary variable. The VAE assumes that the latent distribution takes the form of a Gaussian distribution, whose parameters are learned from an encoder, i.e., $q_{\phi}(\mathbf{z} \mid \mathbf{r}) = \mathcal{N}(\mu(\mathbf{r}), \sigma^2(\mathbf{r}))$, where

$$\mu(\mathbf{r}) = f_1(\mathbf{r}), \ \sigma(\mathbf{r}) = f_2(\mathbf{r}).$$

Here, f_1 and f_2 are two neural networks, e.g., MultiLayer Perceptrons (MLPs). Then, sampling from the latent distribution is equal to first sampling $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, and then generating the sample of $\mathbf{z}$ by computing $\mathbf{z} = \mu(\mathbf{r}) + \boldsymbol{\epsilon}\sigma(\mathbf{r})$. The reparameterization trick allows the VAE to be trained in an end-to-end manner using neural networks, since $\mathbf{z}$ is deterministic and can propagate the error from the decoder to the encoder after exploiting the reparameterization trick. An illustration of the reparameterization trick can be found in Figure 2(a).

SGVB estimator. The Stochastic Gradient Variational Bayes (SGVB) estimator is the result of applying the reparameterization trick and Monte Carlo estimate to the objective, the ELBO. The SGVB estimator is computable and differentiable, so usually it is used to optimize the model, simply by applying Stochastic Gradient Descent (SGD) or Adam [85] to it.

2.2.2 Characteristics of VAEs in recommender systems. Recall the four key characteristics of VAEs that are used in applications in recommender systems: (i) their encoding capability, (ii) their generative nature, (iii) their Bayesian nature, and (iv) their flexible internal structure. Before describing the application of VAEs to recommender systems in details, we summarize the intuitions why these characteristics benefit recommendation performance and provide detailed discussions of the advantages of applying VAEs to recommender systems.

[2]The Kullback-Leibler (KL) divergence is usually used to measure the "distance" between two distributions. The KL divergence can often be calculated analytically.

Encoding capability. The encoding capability refers to the fact that VAEs can learn latent representations of data. The encoder part of a VAE has a strong capability for encoding data, without additional manipulation, e.g., noise corruption in denoising autoencoders [17, 103, 166, 167]. This facilitates that data in recommender systems can be encoded by a VAE. Side information (such as item content, user profiles, etc.) and user-item interaction data are two main types of data that could be encoded by a VAE. Moreover, the data is usually encoded into a distribution of the latent variable, which to some degree retains uncertainty.

Generative nature. The counterpart of the encoding capability of a VAE is its generative nature. This means that a VAE is inherently a generative model, and can simulate how the data are generated in the real world by learning a joint distribution over all variables. The generative nature of VAE lies in two aspects: (i) the learned continuous latent space[3] (distribution), and (ii) the generative network. Different from traditional autoencoders that learn a fixed low-dimensional vector, a VAE learns a continuous latent distribution over the possible latent representations through the encoder. Moreover, the learned distribution is data-dependent, i.e., for each datapoint, a VAE can generate a distribution over some possible values of the latent variable that are likely to reproduce the input data. Apart from reconstructing the input data, with its generative nature, we can also generate other previously unseen data by exploring the learned latent distribution [168]. The generative network can be flexible, i.e., can be chosen to correspond to the encoder or other more expressive deep neural networks.

Bayesian nature. Thanks to its Bayesian nature, a VAE can efficiently use Bayesian inference to optimize the graphical models, which stimulates diverse designs of graphical models for recommender systems. With efficient Bayesian inference, VAE can handle sparse and complex data more appropriately, e.g., user-item interaction data, because uncertainties are considered in Bayesian inference.

Flexible internal structures. The idea of flexible internal structures refers to the internal structures that can be replaced flexibly. Firstly, one can replace the encoder and decoder with other deep learning networks according to the data type. Secondly, exploring more powerful prior and reformulating the ELBO are another two choices for making a VAE more suitable for specific recommendation tasks. To sum up, the flexible internal structures of a VAE are its encoder, decoder, prior, and the ELBO.

Advantages of applying VAEs to recommender systems. VAEs infer embeddings of entities, e.g., embeddings of users and items in the setting of recommender systems, via a variational EM (Expectation-Maximization) algorithm on large datasets. Although they consist of an encoder and a decoder, whose structures are similar to traditional deep learning-based autoencoder algorithms and generative models, they work differently and serve a large range of purposes. It is well-known that VAEs are able to infer representations of entities with disentangled factors. This is because isotropic priors such as Gaussians are adopted on the latent variables in VAEs. Modeling them as isotropic priors enables each dimension of the inferred representations to push themselves as far as possible from each other. In addition, a regularization coefficient can be applied to VAEs to control the influence of the priors over the inferred embeddings [71]. While isotropic priors are sufficient for inferring embeddings of entities in most cases, for specific purposes, one may be interested in using other priors for the embedding inference. For instance, to co-embed embeddings of both attributes and nodes for attributed networks, one may want to define the priors as hyperspherical ones [45]. One more advantage when applying VAEs to applications such as recommender systems is that the priors in VAEs give significant control over how the latent embeddings of entities are modeled. Such flexible control does not exist in many deep learning-based autoencoder algorithms and generative models [32].

[3]In a VAE, we use the terms "latent space" and "latent distribution" interchangeably since the learned latent distribution can be a small (dedicated) part of the latent space.

3 VAE-BASED RECOMMENDATION METHODS

In this section, we introduce VAE-based recommendation approaches. The approaches use either VAEs themselves or optimization strategies of VAEs, so as to perform recommendation tasks. In order to gain better insights into these approaches, they are divided into two categories: *traditional recommendation approaches* and *others*. We refer to VAE-based recommendation methods that exploit traditional recommendation algorithms such as CF as *traditional recommendation approaches*, and to VAE-based recommendation methods that use external positive signals such as linking information among items and other machine learning techniques that are mainly used in other domains such as disentanglement learning [115] as *other approaches*. For instance, VAE-based CF methods [145] fall into the category of *traditional recommendation approaches*, while the VAE-based recommendation methods using cross-domain knowledge to improve the performance of recommendation [4, 126] fall into the category of *other approaches*. We discuss the traditional VAE-based recommendation approaches in Secion 3.1 and other VAE-based methods in Section 3.2.

3.1 Traditional Recommendation with VAE-based methods

In this section, we introduce a number of recommendation systems, where VAEs are used to either model the interaction data or side information in CF methods and hybrid methods (introduced in Section 1). We divide the methods introduced in this section into three categories according to the ways how VAE techniques are used to address challenges in the task of recommendation, as shown in Figure 3:

(1) Directly applying VAEs for recommendation. In this class of methods, VAEs are used to directly generate the recommendation results for users. The specific applications of VAEs in this class of methods are two-fold: to learn user's preference pattern from the user-item interaction history, and to further mine a user's preference with the latent representation of side information.
(2) Extracting side information for recommendation by VAEs. An alternative way of using VAEs for recommendation purposes is to use a VAE or its variants to learn representations of side information, followed by a step to integrate the learned representations into a recommendation model such as PMFs.
(3) Exploiting optimization strategies of VAEs for recommendation. Approaches that fall into this category only exploit optimization strategies of VAEs. These optimization strategies include: the reparameterization trick, the SGVB estimator, applying a neural network to approximate the true posterior distribution of the latent variable, and amortized inference (input-dependent encoder). In contrast to directly exploiting a VAE, these approaches may not be built based on VAEs; these approaches belong to the family of Bayesian models, and VAEs may be adopted to conduct efficient Bayesian inference for them.

There are other taxonomies that can be used to categorize methods, e.g., categorizing them according to whether there are temporal dependencies among items: *static* and *dynamic* recommendation methods. The most prominent characteristic of static methods is that they usually use a matrix, which represents the relationships between users and items, as the main training data. The matrix is called interaction matrix in the *implicit feedback* scenario, or rating matrix in the *explicit feedback* scenario, respectively. The interaction matrix can be constructed as follows: using rows to represent users and columns to represent items, and if a user has interacted with an item, then the corresponding entry of the matrix will be denoted as '1', otherwise '0'. The rating matrix can be similarly constructed by treating ratings of the unrated items as '0' and keeping the ratings of the items unchanged. Some approaches binarize the rating matrix into an interaction matrix by converting the rating that is equal to or more than a threshold to '1' and others to '0' [102, 183]. Besides adopting the interaction (rating) matrix, some methods [36] additionally use side information to improve recommendation performance, e.g., item content [64, 100, 177]. Unlike static methods, in dynamic methods the order of items is usually considered because it will influence the prediction results. In general, temporal dependencies among items are used to describe the

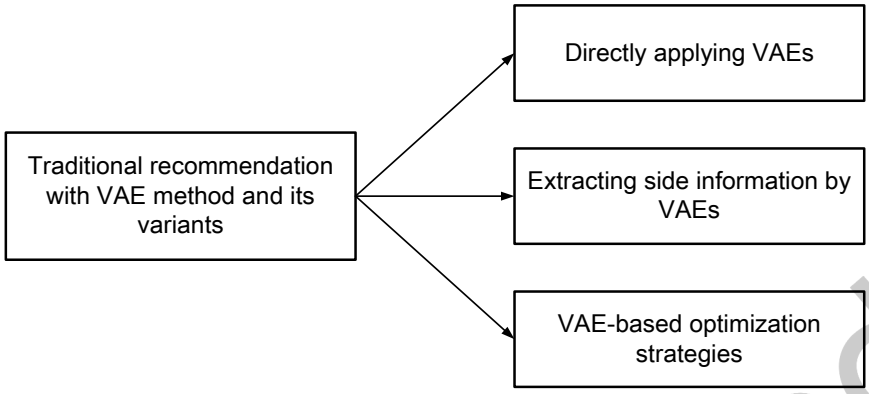


Fig. 3. Taxonomy of VAE-based CF approaches, including those directly applying VAEs [6, 30, 91, 93, 93, 93, 102, 110, 162, 186, 194, 203, 211, 214], those extracting side information by VAEs [28, 33, 57, 75, 127, 196, 197, 217], and VAE-based optimization strategies [7, 34, 74, 80, 83, 110, 114, 135, 149, 161, 162, 183, 201].

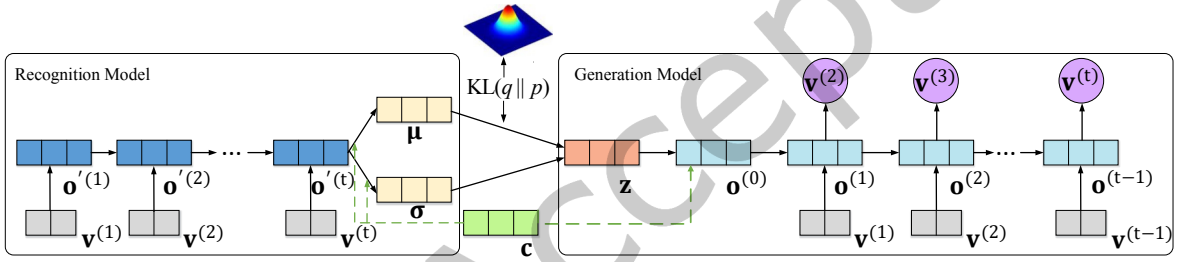


Fig. 4. Example of using a GRU as the encoder and decoder of a VAE. The green dotted line indicates that the condition is optionally added. Image credits: [179].

order of items. Sequential recommendation and sequence-aware session-based recommendation[4] are two types of dynamic method. To model user-item interaction data with temporal dependencies, an RNN or variant (e.g., Long Short Term Memory (LSTM) or Gated Recurrent Unit (GRU)) usually serves as the encoder and decoder of the VAE. Figure 4 shows an example of using GRUs for VAEs [179]. Next, top-n recommendation[5] is usually addressed using dynamic methods. Rating prediction at the next time step for user i and item j is also addressed using dynamic methods [154].

In this section, we organize the presentation using the taxonomy in Figure 3, that is, according to the way in which VAE-based techniques and their variants are used to address the challenges of RSs. However, when appropriate, we distinguish between static and dynamic approaches within the categories of Figure 3.

3.1.1 Directly applying VAEs for recommendation. Firstly, VAEs are mainly used to model the generative process of user-item interaction data, which is similar to Section 2.2.1. The decoder outputs the predicted probability vector over all items, which can be interpreted as a user preference pattern, characterizing user's preference

[4]Although general session-based recommendation methods are not all dynamic, the sequential context is an important source of information for session-based recommendation. To prevent any ambiguity, henceforth in this paper, we will use the term "session-based recommendations" to refer to "sequence-aware session-based recommendations."

[5]When $n = 1$, top-n recommendation is equivalent to the next clicked item prediction task considered in session-based recommendation.

towards all the items. We can rank the unrated items according to their probability values in the predicted probability vector, and recommend the top-n items to the user. The VAE uses amortized inference to learn the parameters in the model. This is in line with the core of CF, which analyzes user preferences by exploiting similar patterns inferred from past experiences [102]. The "collaborative" aspect is reflected by the common encoder and decoder shared by all users. After the model has been learned, all the collaborative information is obtained by the VAE, which can even unveil the preference pattern of a user whose history was not used to train the VAE [102]. Note that this is different from the method [186] that uses denoising autoencoders for recommendation, in which the input user interaction data is corrupted with Gaussian noise. In a VAE, the latent representation of the encoder's output is combined with stochastic noise, and input into the decoder. Moreover, compared with models that use denoising autoencoders, the advantage of applying VAEs for recommendation is that the latent representation is distributed. To some degree, the VAE retains the uncertainty of a user, i.e., the user's preferences are uncertain when the user has few interactions with items, or has interactions with diverse items. Moreover, Bayesian inference can be used to optimize the model, making the recommendation performance robust even when the data is sparse.

Secondly, VAEs are used to improve a user's representation with latent representations of side information, which is mainly conducted by the encoder. The latent representation of side information can be further integrated into the process of modeling the generative process of user-item data, so as to form a hybrid method [91, 93].

The generative nature and Bayesian nature (used to optimize the model's parameters) of a VAE are reflected in these category of methods. The encoding capability of VAEs is reflected in some methods in which a VAE is used to learn the latent representation of side information. Table 2 summarizes VAE-based recommendation methods that work with different internal structures of VAEs. Next, we will introduce methods that directly apply VAEs for recommendation from 6 angles. First, vanilla VAE models for recommendation are introduced. Then enhanced VAE models with side information, reasonable priors, powerful encoders, increasing uncertainty and other strategies are introduced, respectively.

Vanilla VAEs for recommendation. Some methods use a vanilla VAE to directly model the user-item interaction data [93, 102]. Lee et al. [93] propose a model called VAE-CF, which is a basic version of modifying a VAE to accommodate CF for handling implicit feedback without side information. VAE-CF uses only the implicit feedback data for CF. Specifically, a vanilla VAE is used to reconstruct the implicit feedback from which the user's preference pattern can be extracted, which is similar to the role introduced in Section 2.2.1. Liang et al. [102] propose a model called partially regularized VAE with Multinomial likelihood (Mult-VAE), which is another kind of implementation of the VAE-CF introduced in [93]. Firstly, they use *multinomial likelihood* for the distribution of implicit feedback data, which is empirically shown to be more suitable for implicit feedback and a good proxy for the top-n ranking loss. Secondly, a parameter $\beta \in [0, 1]$ is introduced to the KL divergence term in the ELBO to govern regularization, which improves the recommendation performance compared to only maximizing the ELBO, and leads to a reformulated ELBO as follows:

$$\mathcal{L}_{\text{Mult-VAE}} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{r})}[\log p_{\boldsymbol{\theta}}(\mathbf{r}|\mathbf{z})] - \beta \cdot D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{r}) \| p(\mathbf{z})). \tag{5}$$

To efficiently tune the parameter β, an annealing method is used by starting training with $\beta = 0$, and then gradually increasing it to 1. In this process, β that achieves the best performance will be recorded. To tune β, a scaling factor related to the number of observations of a user to serve as β is proposed [149]. The encoder and decoder structures for Mult-VAE are adopted as $600 \rightarrow 200 \rightarrow 600$. In other VAE-based methods, the encoder and decoder structures could be set as a fully-connected layer (e.g., the latent embedding dimension is set to 100 in MacridVAE [115]). In addition to these structural considerations, the other hyperparameters such as learning rate, batch size, L2 regularization, optimizer are tuned with the TPE method [13], based on the dataset at hand.

Mult-VAE indicates that a reformulated ELBO with additional β controlling the regularization, i.e., KL divergence term, can improve the recommendation performance.

Similar to Mult-VAE, several publications use a VAE to directly generate recommendations without side information [6, 30, 110, 162, 182, 194, 203, 211, 214]. The difference between these methods and Mult-VAE is that different or additional manipulation is used, e.g., they may use different likelihood for the user-item interaction data. Zhang et al. [211] propose a VAE-based model called exploitation-exploration motivated VAE (XploVAE) for CF. They extend user-item interaction data to consider higher-order proximities between users and items. Askari et al. [6] propose a Joint Variational Autoencoder (JoVA) model, an ensemble of two VAEs, that jointly learns both user and item representations under uncertainty, and then collectively predicts user preferences for recommendation. By doing so, JoVA is able to encapsulate user-to-user and item-to-item correlations at the same time. Choi et al. [30] propose a local recommendation algorithm, called Local Collaborative Autoencoders (LOCA), that provides a VAE-based generalized architecture for learning a variety of local models by identifying various sub-communities for training and inference of the VAE model.

Enhanced VAEs with side information. To alleviate the data sparsity problem and improve the recommendation performance, side information (e.g., social relationships [196, 197]) is fused into vanilla VAEs for recommendation. Considering the way side information is fused with implicit feedback, augmented VAE models can be divided into *conditional-based* VAE models and *joint-based* VAE models. Next, we introduce the augmented VAE models in each category.

- **Conditional VAE (CVAE-CF) and variants.** Motivated by the Conditional VAE [41], Lee et al. [93] propose CVAE-CF, which models a user's implicit feedback given side information. Based on the traditional Conditional Variational AutoEncoder (CVAE), the hybrid method CVAE-CF introduces an additional latent variable $\mathbf{h}$ into the graphical model to extract the latent representations of side information, potentially leading to better recommendation performance. The graphical model of CVAE-CF is shown in Figure 5 (a). Similar to VAE-CF, the CVAE-CF model also adds regularization in the form of KL divergence to avoid overfitting. It helps the models focus more on implicit feedback than side information, since the goal is to achieve a better recommendation performance. The final ELBO of CVAE-CF is defined in Eq. (6), where α_1 and α_2 are hyper parameters used to govern the KL divergence:

$$\begin{aligned}\mathcal{L}_{\text{CVAE-CF}} = \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\mathbf{r}|\mathbf{z},\mathbf{h})] - D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{r})\|p(\mathbf{z})) \\ - \alpha_1 D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{h}|\mathbf{x})) - \alpha_2 D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})\|q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{r})).\end{aligned} \tag{6}$$

  Compared with the ELBO of vanilla VAEs in Eq. (4), the changes of the ELBO are: (i) An additional latent variable $\mathbf{h}$ is used to reconstruct the implicit feedback in the negative reconstruction error term; (ii) Side information is additionally used to infer the latent variable; (iii) An additional KL divergence term, $D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x})\|q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{r}))$, is added to force the model to uncover the latent representation related to implicit feedback. These changes of the ELBO in $\mathcal{L}_{\text{CVAE-CF}}$, have been proven experimentally to achieve better recommendation performance.
  Some methods [28, 33, 57, 75, 127, 217] augment vanilla VAEs with extra side information, using similar designs as CVAE-CF, but with different implementations and diverse side information. Their graphical models can be seen as variants of that of CVAE-CF. Iqbal et al. [75], Polato et al. [131] and Pang et al. [127] consider adopting CVAEs to fuse side information into VAEs. Different from CVAE-CF, they only used a latent variable, i.e., a common latent variable is used for both implicit feedback data and side information. In particular, Iqbal et al. [75] consider the user profile as the condition of the CVAE. The profile refers to the genre of interest, e.g., Romance, Comedy and Horror in the context of movie recommendation. Polato et al. [131] feed the item's categories as the condition into the encoder to make recommendations that satisfy certain constraints. Also based on CVAEs, Pang et al. [127] use the label of user or item as a condition.

Assuming similar users have similar purchasing preferences, Pang et al. [127] use the labels of users to enable the CVAE to represent users with the same labels with similar representations. While most of the conditions used in a CVAE are static, Kim [84] considers time-varying item features learned by an LSTM as the condition of a CVAE. Zhu and Chen [217] propose a collaborative variational bandwidth auto-encoder (VBAE) for recommendation. VBAE models both the collaborative and user feature embeddings as Gaussian random variables inferred via neural networks to capture non-linear semantic similarities among users using their ratings and side information.

- **Joint VAE (JVAE-CF) and variants.** Inspired by the joint modalities VAE (JMVAE) [157], Lee et al. [93] propose Joint Variational AutoEncoder (JVAE)-CF, which models a joint distribution of user's implicit feedback and side information. Similar to CVAE-CF, an additional latent variable $\mathbf{h}$ is introduced into the graphical model of JVAE-CF (as shown in Figure 5(b)), to extract latent representations of side information. Different from CVAE-CF, which only reconstructs implicit feedback, JVAE-CF reconstructs implicit feedback and side information simultaneously. The final ELBO of JVAE-CF is defined in Eq. (7):

$$\begin{aligned}\mathcal{L}_{\text{JVAE-CF}} = \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{h})] + \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\mathbf{r}|\mathbf{z},\mathbf{h})] - D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{r})\|p(\mathbf{z})) \\ - D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x},\mathbf{r})\|p(\mathbf{h})) - \alpha_1 D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x},\mathbf{r})\|p_{\boldsymbol{\theta}}(\mathbf{h}|\mathbf{x})) \\ - \alpha_2 D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{x},\mathbf{r})\|q_{\boldsymbol{\phi}}(\mathbf{h}|\mathbf{r})).\end{aligned} \tag{7}$$

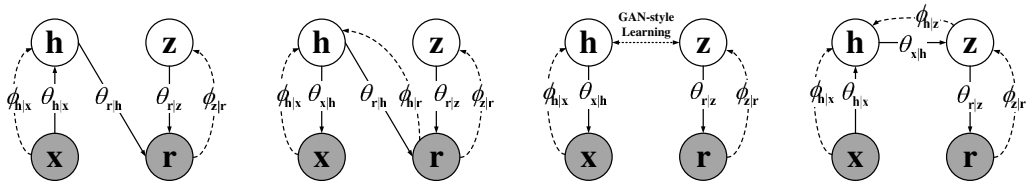
  The model proposed by Chen and de Rijke [28] can be seen as a variant of JVAE-CF. Inspired by the collective SLIM [125], where both the user rating and side information are collectively reproduced by the same coefficient matrix through a linear matrix multiplication, Chen and de Rijke [28] use the same VAE to encode the interaction matrix and side information, and reconstruct them collectively. Since side information will be first put into a VAE to pretrain the network before implicit feedback, a larger hyper parameter β (see Eq. (5)) is used to make the posterior comply more with the prior from side information. Gupta et al. [57] and Cui et al. [33] use two VAEs to encode implicit feedback and side information, respectively. Gupta et al. [57] use one VAE to learn the latent representation of movie reviews given by a user, which is used to reshape implicit feedback data, and the refined implicit feedback data is passed to another VAE to generate recommendations. Cui et al. [33] propose the Variational Collaborative Model (VCM), which uses two VAEs to learn latent representations of implicit feedback and users' reviews with a multinomial distribution, respectively. The two VAEs guide each other to learn synchronously: the model pulls the two posteriors in the two VAEs close to each other in terms of the KL divergence; by doing so, the latent representation in one VAE actually helps the other VAE reconstruct the data.

To improve the structure of modeling $\mathbf{h}$, [93] propose VAE-AR and Conditional Ladder Variational AutoEncoder (CLVAE), which are built on Generative Adversarial Nets (GANs) [54] and Ladder Variational AutoEncoders (LVAEs) [152], respectively. In VAE-AR, the learning trick of a GAN is used to force the latent representation of implicit feedback to be similar to the latent representation of side information, ensuring a better fusion of these two modalities. The graphical model is shown in Figure 5 (c). Given side information, a CLVAE models the conditional distribution of the implicit feedback data with a ladder structured recognition model LVAE, to learn a more expressive latent representations of implicit feedback data and side information. The graphical model is shown in Figure 5 (d). The integration with a GAN and a LVAE show the flexibility of VAEs for incorporating other deep learning methods. Some of the internal structure (e.g., regularization in the ELBO) of the models introduced above can be found in Table 2.

Enhanced VAEs with reasonable priors. From Table 2, it is evident that in most approaches that use a VAE for recommendation, the prior of the latent variable for each user is a standard Gaussian distribution as in a VAE [see, e.g., 102]. This choice may limit the ability of a VAE to model more expressive representations. For instance, when a VAE is employed to encode a user's implicit feedback vector, it is commonly assumed that the encoded vector

Table 2. Comparison of flexible internal structures of different models that directly use VAEs to generate recommendation results. "Regul." is short for "Regularization". We use authors' names to represent models that do not have model names. "R" or "U" in the "ELBO" column indicates whether the ELBO is reformulated or unchanged, respectively, compared to the vanilla VAE. "A" (Adaptive) in the "Regularization" column means that a hyper parameter is used to control the regularization level and "N" otherwise. "Likelihood" refers to the likelihood distribution used for the user-item interaction data.

Model	Year	Encoder	Prior	ELBO	Regularization	Likelihood
VAE-CF [93]	2017	MLP	$\mathcal{N}(0, I)$	U	N	Bernoulli
CVAE-CF [93]	2017	MLP	$\mathcal{N}(0, I)$	R	A	Bernoulli
JVAE-CF [93]	2017	MLP	$\mathcal{N}(0, I)$	R	A	Bernoulli
VAE-AR [93]	2017	MLP	$\mathcal{N}(0, I)$	U	N	Bernoulli
CLVAE [93]	2017	MLP	$\mathcal{N}(0, I)$	R	N	Bernoulli
Mult-VAE [102]	2018	MLP	$\mathcal{N}(0, I)$	R	A	multinomial
Iqbal et al. [75]	2019	MLP	$\mathcal{N}(0, I)$	R	A	multinomial
Zheng et al. [214]	2020	not mentioned	$\mathcal{N}(0, I)$	R	A	Bernoulli
Lobel et al. [110]	2019	MLP	$\mathcal{N}(0, I)$	R	A	multinomial
XploVAE [211]	2020	not mentioned	$\mathcal{N}(0, I)$	U	N	Bernoulli
Zhang et al. [203]	2018	MLP	not mentioned	R	A	multinomial
Tong et al. [162]	2019	MLP	$\mathcal{N}(0, I)$	U	N	not mentioned
Polato et al. [131]	2020	MLP	$\mathcal{N}(0, I)$	R	A	multinomial
Pang et al. [127]	2019	MLP	$\mathcal{N}(0, I)$	R	A	Bernoulli
Kim [84]	2019	not mentioned	not mentioned	R	A	multinomial
Chen and de Rijke [28]	2018	MLP	$\mathcal{N}(0, I)$	R	A	Bernoulli
Gupta et al. [57]	2018	MLP	$\mathcal{N}(0, I)$	U	N	logistic
VCM [33]	2018	MLP	$\mathcal{N}(0, I)$	R	A	multinomial
Karamanolakis et al. [80]	2018	MLP	user-dependent prior	R	A	multinomial
Kim and Suh [83]	2019	hierarchical stochastic units, gated linear units	VampPrior	R	N	multinomial
RecVAE [149]	2020	densely collected layer	composite prior	R	A	multinomial
VAEGAN [201]	2019	implicit inference model	not mentioned	R	A	not mentioned
p-VAE [114]	2018	partial inference network	$\mathcal{N}(0, I)$	U	N	not mentioned
Q-VAE [183]	2019	not mentioned	$\mathcal{N}(0, I)$	R	N	not mentioned
ACVAE [194]	2020	MLP	$\mathcal{N}(0, I)$	R	A	multinomial
VBAE [217]	2021	MLP	$\mathcal{N}(0, I)$	R	A	Bernoulli
JoVA [6]	2021	MLP	$\mathcal{N}(0, I)$	R	A	Bernoulli
LOCA [30]	2021	MLP	$\mathcal{N}(0, I)$	R	A	multinomial



(a) CVAE-CF (b) JVAE-CF (c) VAE-AR (d) CLVAE-CF

Fig. 5. Graphical models of (a) CVAE-CF, (b) JVAE-CF, (c) VAE-AR and (d) CLVAE. The dashed line refers to the inference process and the solid line refers to the generative process. Image credits: [93].

adheres to a uniform Gaussian prior for all users. However, this uniform Gaussian prior can be overly simplistic and may limit the model's ability to represent the diverse preferences of users, particularly active users with a wide range of interests. Consequently, the latent representations learned by the VAE may not align well with the intricate and multifaceted preferences of these users. Ideally, the latent variable of each user has a dedicated prior, which could model the user's preferences better. As explained in Section 2.2.2, the prior could be replaced to achieve better recommendations. Karamanolakis et al. [80], Kim and Suh [83] replace the widely used standard Gaussian distribution with more reasonable priors, to improve the recommendation performance. Karamanolakis et al. [80] propose a user-dependent Gaussian prior, whose parameters (i.e., mean, variance) are obtained by encoding the user's explicit text reviews with word embeddings or topic models [15]. The user-dependent prior can avoid additional KL annealing strategies used in Mult-VAE [102]. Kim and Suh [83] use the flexible prior called variational mixture of posterior prior [VampPrior, 161] to replace the original standard Gaussian prior. Shenbin et al. [149] adopt a composite prior, i.e., a convex combination of a standard normal distribution and an approximate posterior with fixed parameters from the previous training epoch.

Enhanced VAEs with powerful encoders. Table 2 presents several kinds of encoders for VAE-based recommendation methods. Increasingly complex interaction data and side information require a powerful encoder to extract more knowledge from this type of data, so as to achieve better recommendation performance. Kim and Suh [83] use the hierarchical stochastic unit [161] and gated linear units [34] to change the structure of the encoder of the original VAE for enhancing the modeling capacity. In their Recommender VAE (RecVAE), Shenbin et al. [149] present a more complex inference network, which is similar to the dense connected layers of dense CNNs [74], and borrow ideas from *swish activation functions* [135] and layer normalization [7]. Combining GANs [54] and VAEs, Yu et al. [201] propose VAEGAN, which reformulates the KL divergence term of the ELBO and derives an arbitrarily flexible implicit inference model to approximate the true distribution of the latent variable.

Enhanced VAEs with increasing uncertainty. Uncertainty is important in RSs, since the user-item interaction data is sparse and unbalanced. For instance, some users may have interacted with many different kinds of items, but some users have only interacted with a limited number of items. Enhancing the uncertainty could, to some degree, improve the recommendation performance. Ma et al. [114] propose Partial VAE (p-VAE), which only exploits the observed ratings for inferring the user latent representation and adopts a partial inference network called *permutation invariant set function encoding* to replace the inference network (encoder) of the original VAE. They do not regard the missing entries in the rating matrix as '0', which can capture the uncertainties in missing data and slightly improve the recommendation performance. Wu et al. [183] propose a model called Queryable-VAE (Q-VAE) to reformulate the ELBO to support arbitrary conditional queries over observed user interactions, i.e., to maximize the joint probability of two arbitrarily selected partitions of the observed data. This appropriately increases uncertainty in cases where a large number of user preferences may lead to an ambiguous

Table 3. Performance improvements after changing the internal structures of VAEs used in the recommendation models. We use the authors' names to represent models that do not have model names.

	ML-20M		Netflix	
Model	NDCG	Recall	NDCG	Recall
Mult-VAE [102]	0.426	0.395	0.386	0.351
RecVAE [149]	0.442	0.414	0.394	0.361
Kim and Suh [83]	0.445	0.413	0.409	0.377

user representation and degrade the recommendation performance. The recommendation performance is also improved accordingly. These methods show that changing the internal structures (encoder and ELBO) can help handle the uncertainty problems in VAEs.

Enhanced VAEs with other strategies. Other strategies, such as GANs, are applied to VAEs to help better model user-item interaction data. Like the VAE-AR [93], several methods use a GAN to improve the recommendation performance. Zhang et al. [203] use a GAN to improve the latent representations of user interaction data, so as to improve the recommendation performance. Tong et al. [162] combine a VAE with a GAN to improve the recommendation performance. Specifically, a VAE is used to model the generative process of the user interaction vector and a GAN is used to conduct adversarial training to improve the quality of the generation process in order to improve the recommendation performance. Lobel et al. [110] use a VAE as the *actor* in the proposed RaCT model, which is an actor-critic reinforcement learning [129]; a *critic* network is trained to approximate ranking-based metrics to further improve the recommendation performance. By adding these strategies to VAEs, VAEs are more adaptive to the recommendation scenarios, consequently improving the recommendation performance in each scenario.

Discussion. Though some methods [see, e.g., 83, 102, 149] only use user-item interactions, changing the internal structure of VAE can help to improve recommendation performance. Table 3 shows an example of the influence on performance when changing the internal structure of a VAE. The performance is evaluated on two datasets, ML-20M [62] and Netflix Prize [12], and the evaluation metrics are NDCG@100 and Recall@20. Compared with Mult-VAE, RecVAE and the model proposed by Kim and Suh [83] can improve the recommendation performance significantly due to the changes of the internal structures of VAEs. Further, the model proposed by Kim and Suh [83] achieves a better performance than RecVAE (on some metrics). The reason appears to be that the gated linear units proposed by Kim and Suh allows information to propagate better in the network. Some methods, e.g., those in [28, 33, 57, 93], use a VAE to encode side information into their latent representation, which is further used to improve the recommendation performance. From Table 2, we can see that an MLP is most frequently used, which indicates that the field still lacks explorations of other types of side information such as, e.g., image data. From the viewpoint of encoding capability and internal structures, the advantage of a VAE's flexibility makes it adapt to multimodal data, with MLP, the gated linear units, and GAN etc. From the perspective of generative and Bayesian nature, the advantage of VAE helps it to capture the uncertainty of the user latent representation. Meanwhile, due to the limitation of the sparse user-item interaction data, most of VAEs still use relatively shallow encoders and decoders [174], unlike the deep networks applied in CV and NLP areas. It brings disadvantages and also potentiality for improvement to the encoding capability and generative nature.

Next, we consider the application of VAEs in dynamic settings where they are used to directly produce the prediction, e.g., next clicked item, next interacted session, or next interacted top-n items. This category of methods belongs to the family of generative models, and Bayesian inference is used to optimize them; the generative and Bayesian nature of VAEs are reflected in these methods. The ELBOs of these methods are similar to Eq. (4), and a

Table 4. Encoders of VAEs and their likelihood used to model the user-item interaction data.

Model	Encoder	Likelihood
VRM [179]	GRUs	not mentioned
VASER [215]	GRUs, normalizing flows	multinomial
Sachdeva et al. [142]	RNN	multinomial

multinomial distribution is usually used to model user-item interaction data. However, the encoders of these methods models are different; see Table 4.

Session-based recommendation [76]. Current classical session-based recommendation methods usually model the generation of item sequences in a short session. The adoption of VAEs in session-based recommendation allows us to model the generation of the session from a probabilistic perspective, which helps to understand the randomness and uncertainties of the session generation. Wang et al. [179] propose a session-based recommendation model, called Variational Recurrent Model (VRM), which unifies knowledge of frequent click patterns as the distribution of a stochastic latent variable. The authors assume that the sequence of interactions in a session are generated from the latent variable, and the next predicted item is obtained from the generated sequence. The VAE in this approach serves as a generative model to simulate the generative process of session sequences of a user. The encoder and decoder of the VAE are GRUs, for modeling sequential data (see the overall model in Figure 4). The next clicked item is predicted using a probability distribution on each item, produced at each step of the generating process. A model similar to VRM has been proposed by Zhong et al. [215], who present a generative session-based recommendation framework, VAriational SEssion-based Recommendation (VASER). VASER uses a multinomial distribution to model user-item interaction data, and adopts a normalizing flow [138] to approximate the posterior of the latent variable. Another difference between VRM and VASER is that, in VASER, a deterministic attention mechanism [195] is used to enhance the GRU generative network, by dynamically selecting and linearly combining different parts of the input sequence in the inference network.

Sequential recommendation. As with session-based recommendations, for sequential recommendations VAEs are used to model the consumption sequence of each user. Sachdeva et al. [142] introduce a recurrent version of a VAE, recurrent VAEs, to perform CF for sequential recommendation, using a subset of user consumption with temporal dependencies. Multiple architectures of recurrent VAEs are proposed, using different latent dependencies of the variables. These architectures reveal the flexibility of combining VAEs and RNNs. Compared with Mult-VAE [102], the methods in this work additionally consider temporal dependencies among items, so as to notably improve the recommendation performance. Xie et al. [194] propose a novel sequential recommendation model to enhance the encoder in VAE, where contrastive learning is employed with a VAE by minimizing the contrastive loss so as to achieve better generalization ability of the model. Zhao et al. [212] propose a novel variational self-attention network (VSAN) for sequential recommendation. VSAN introduces variational inference to self-attention networks to handle the uncertainty of user preferences by employing the VAE paradigm; two self-attention networks serve as the encoder and decoder of the VAE framework. Deffayet et al. [35] consider a slate recommendation scenario, where a *list* of items is recommended at each interaction turn; to ensure tractability, the authors encode slates in a continuous, low-dimensional latent space learned by a variational auto-encoder. Then, a reinforcement learning agent selects continuous actions in this latent space, which are ultimately decoded into the corresponding slates.

Discussion. The success of modeling sequential data by a VAE and RNNs (GRUs) in these methods indicates that a VAE is a flexible model, i.e., other deep learning methods can be incorporated into the VAE framework. By using a VAE to model the generative process of sequential data in RSs, Bayesian inference can be performed. And

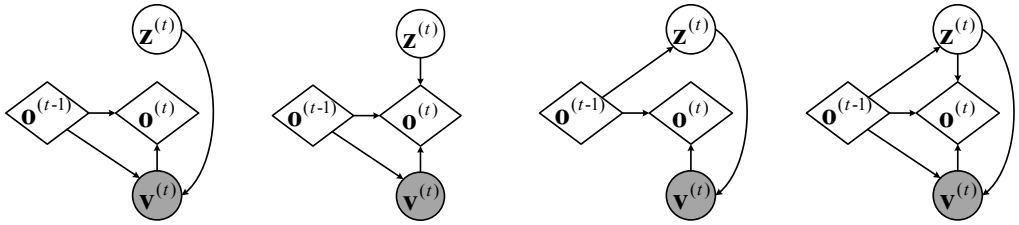


Fig. 6. Examples of different graphical models of recurrent VAEs. $\mathbf{z}^{(t)}$ denotes the latent variable at time t and $\mathbf{v}^{(t)}$ denotes the user's interaction item at time t. $\mathbf{o}^{(t)}$ refers to the hidden state of RNN at time t. (a) user consumption history (history for short) and a single latent variable determine the next click item; (b) history and multiple latent variables determine the next clicked item; (c) history determines a single latent variable and the latent variable determines the next clicked history; (d) history and multiple latent variables determine the latent variable that is used to determine the next clicked item. Image credits: [142]

the uncertainties of user-item interactions can be preserved, which is beneficial for dealing with data sparsity and improving the model's performance. Moreover, by modeling temporal dependencies among items, these methods outperform methods that ignore temporal information.

3.1.2 Extracting side information for recommendation with VAEs. The VAE-based approaches discussed in Section 3.1.1 use VAEs to directly generate recommendations. That is, they use a VAE to model user-item interaction data and produce top-n recommendations by predicting preference scores for items that a user has not interacted with. An alternative way of using VAEs for recommendation purposes is to use a VAE to learn latent representations of side information, followed by a step to integrate the learned representations into a recommendation model, such as PMFs. Table 5 lists different recommendation models that VAEs have been combined with. To integrate side information and rating information in a unified model, the combined approaches listed in Table 5 modify the ELBO used in vanilla VAEs in Eq. (4) to include additional latent variables to represent side information. Often, these combined methods are Bayesian generative latent variable models, in which efficient Bayesian inference of the VAE can be conducted.

The difference between the hybrid models in Section 3.1.2 and Section 3.1.1 is that VAEs in the hybrid models in Section 3.1.2 are not used to generate recommendations. Instead, other recommendation models are used to model the interactions between users and items and generate predictions of unrated items. Figure 7 illustrates the main difference between the hybrid collaborative methods in Section 3.1.1 and Section 3.1.2.

Below, we will introduce models that combine VAEs with Probabilistic Matrix Factorization (PMF) and other recommendation models, e.g., Neural Collaborative Filtering (NCF) [67].

Combining side information of items with PMF. Li and She [100] only use side information of items (item content). The graphical model is shown in Figure 8. Besides exploiting PMF to factorize the interaction matrix, the authors use a VAE to simultaneously learn the latent representation of item content in the proposed generative latent variable model. They assume that the item content is generated from a content latent variable $\mathbf{h}_j^v$.[6] Additionally, for the item, they also use a collaborative latent variable $\mathbf{v}_j^\dagger$ to represent the interaction information between user and item.[7] In other approaches, authors directly use a bias to replace the collaborative latent variable [37, 64, 190].

[6]We use 'content latent variable' here instead of 'latent content variable' as in [100] for better alignment with naming schemes for other latent variables.

[7]The collaborative latent variable of an item refers to the latent variable of the item's interaction vector with users.

Table 5. Recommendation methods that combine other models (the third column) with VAEs.

Paper	Year	Other model	Integration method	Side information	Dataset/Domain
CMVAE [9]	2019	PMF	sharing latent variable	title, abstract, link relationship	CiteULike
CVDL [37]	2019	PMF	linear combination	user profile, the description of item, item tags	CiteULike, Healthcare Dataset
CAVAE [64]	2019	PMF	setting as a bias	item contetn,tag	CiteULike
CVAE [100]	2017	PMF	linear combination	title, abstract	CiteULike
CML [123]	2019	PMF	setting as an offset vector	title, description, category	Amazon
DCBVN [170]	2020	PMF	nonlinear transformation	user profile	Training course
BDCMF [190]	2019	PMF	setting as bias and latent offset vector	social interaction, contents of item	LastFM, Delicious
NVMF [191, 192]	2019	PMF	sharing latent variable	user attributes, item title category	MovieLens, Bookcrossing
VDCMF [193]	2019	PMF	setting as an offset vector	user social, item content	LastFM, Epinions
NeuHash-CF [60]	2020	MF	Hamming distance	review	Yelp, Amazon
DSHRM [25]	2017	LFM	setting as additional latent variables	review	Amazon
NVCF [38]	2019	MLP, GMF	setting as latent variable directly	demographics, genres, categories, social relations	MovieLens, Yelp
VAE-based CF [65]	2019	MLP/ NCF	setting as latent variable directly	user attributes, item title category	MovieLens
NVHCF [189]	2018	MLP	conditional prior	demographics, description	MovieLens,LastfM
BiVAE [163]	2021	MLP	dot product	review	MovieLens, Amazon
DAVE [199]	2021	MLP	adversarial learning	None	MovieLens,Yelp, Digital Music, Pinterest
LVSM [29]	2020	Item-based CF	setting as latent variable directly	review	Amazon
CVRank [77]	2018	Pairwise ranking-based CF	setting as latent variable directly	title, abstract	CiteULike
RRGAN [26]	2019	GAN-based model	adversarial learning	review	Amazon
ACVAE [194]	2021	Contrastive learning model	adversarial learning	None	MovieLens, Yelp
FLVAE [165]	2021	Augmentation model	Hadamard product	None	MovieLens, Netflix

In the end, the item latent variable $\mathbf{v}_j$ (take item j as an example) consists of the content latent variable $\mathbf{h}_j^v$ and

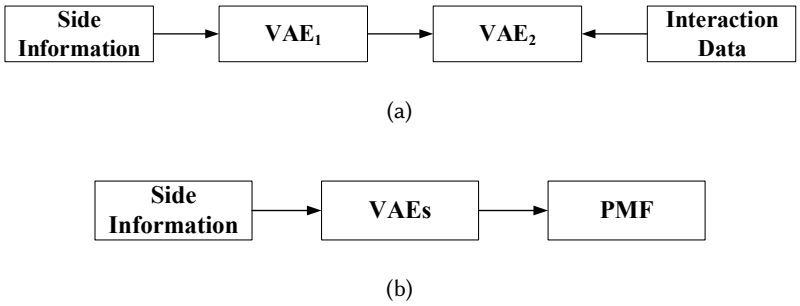


Fig. 7. Illustration of the different schemes for incorporating side information in Section 3.1.1 and 3.1.2: (a) uses a VAE to learn the latent representation of side information followed by incorporating it into another VAE to generate recommendations. (b) incorporates the learned representation into other recommendation models (e.g., PMF) to generate recommendations instead.

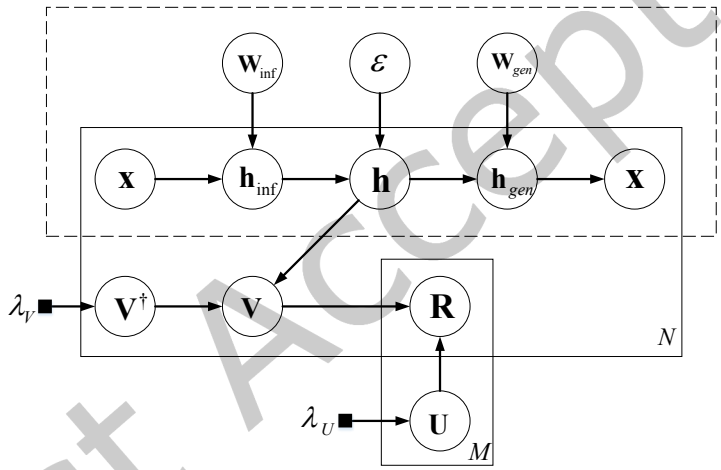


Fig. 8. Graphical model by Li and She [100]. Inside the dashed line box is a VAE used to learn the latent representation of item side information. $\mathbf{W}_{\text{inf}}$ and $\mathbf{W}_{\text{gen}}$ are parameters of the network. λ_V and λ_U are hyper parameters.

the collaborative latent variable $\mathbf{v}_j^{\dagger}$, as below:

$$\mathbf{v}_j = \mathbf{v}_j^{\dagger} + \mathbf{h}_j^{v}. \tag{8}$$

With user i's latent variable $\mathbf{u}_i$ predefined, user-item interaction is thus generated by the item latent representation and the user latent representation, as follows:

$$\mathbf{r}_{ij} \sim \mathcal{N}(\mathbf{u}_i^T \mathbf{v}_j, C_{ij}^{-1}), \tag{9}$$

where C_{ij} is the confidence for $\mathbf{r}_{ij}$, similar to that in Collaborative Topic Regression [169] and $\mathcal{N}(\cdot, \cdot)$ is a Gaussian distribution. Note that $\mathbf{v}_j^{\dagger}$ and $\mathbf{h}_j^{v}$ follows the standard Gaussian distribution. Without presenting the derivation

process, we directly show the ELBO:

$$\begin{aligned}\mathcal{L} = &- D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{h}_j^v|\mathbf{x}_j^v)\|p(\mathbf{h}_j^v)) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{h}_j^v|\mathbf{x}_j^v)} \log p_{\boldsymbol{\theta}}(\mathbf{x}_j^v|\mathbf{z}_j^v) \\ &+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{h}_j^v|\mathbf{x}_j^v)} \log p(\mathbf{v}_j|\mathbf{h}_j^v) + const,\end{aligned} \tag{10}$$

where *const* refers to terms that are not related to $\mathbf{h}$. Compared with the ELBO of a vanilla VAE in Eq. (4), the ELBO above actually has a similar form, but with an additional term $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{h}_j^v|\mathbf{x}_j^v)} \log p(\mathbf{v}_j|\mathbf{h}_j^v)$, which means the final item latent variable $\mathbf{v}_j$ is formed partially by the item content latent variable $\mathbf{h}_j^v$). Notably, due to the flexibility of the ELBO, the condition c can be also added to derive the ELBO of conditional VAE, e.g., altering $q_{\boldsymbol{\phi}}(\mathbf{h}_j^v|\mathbf{x}_j^v)$, $p(\mathbf{h}_j^v)$ and $p_{\boldsymbol{\theta}}(\mathbf{x}_j^v|\mathbf{h}_j^v)$ to $q_{\boldsymbol{\phi}}(\mathbf{h}_j^v|\mathbf{x}_j^v, c)$, $p(\mathbf{h}_j^v|c)$ and $p_{\boldsymbol{\theta}}(\mathbf{x}_j^v|\mathbf{h}_j^v, c)$ respectively. It means that we can incorporate other information (e.g., tags) to learn more robust latent representations. Following Li and She [100] who combine a VAE and PMF, other approaches enrich the item content for a better representation learning of the item. Bai and Ban [9] use multiple VAEs to learn multiple types of side information of an item. These VAEs share one latent variable for these side information. Besides item content, He et al. [64] also use item tags to assist VAEs to learn more robust item representations, with a similar setting of JMVAE [157].

Integrating side information of users into PMF. Integrating side information of items can alleviate the item cold-start problem, but it does not aid to combat the user cold-start problem. Using side information of an item can enhance the expressivity of the item latent representation, the user's latent representation cannot be improved simultaneously. These issues stimulate researchers to incorporate side information of user to PMF. Xiao et al. [190, 193] propose graphical models (see the example in Figure 10(a)) to additionally take social information of users into consideration, where VAE is used to learn the representation of side information of items. Wang et al. [151] employ a VAE with autoencoding variational inference to extract interpretable latent representations of employees' competencies from their skill profiles. Nguyen and Ishigaki [123] use two VAEs to learn latent representations of the textual and categorical information of users, respectively. Unlike Nguyen and Ishigaki [123], Xiao and Shen [191, 192] harness two VAEs to learn latent representations of users and items, and establish a connection between the two VAEs and PMF using the user and item latent variables. Apart from the user and item latent variables, four additional latent variables are used in the model: a user feature latent variable, an item feature latent variable, a user interaction latent variable, and an item interaction latent variable. The user feature and item feature latent variables are used to represent the user features and item features,[8] respectively. The user interaction and item interaction latent variables are used to represent the users' interaction vectors over all items and the items' interaction vector over all users. Figure 9 shows a graphical model of this approaches [191, 192]. Treating patients as users and doctors as items, Deng and Huangfu [37] use a standard VAE to learn a user content latent representation, and another VAE, the same as that proposed in [64], to learn an item content latent representation with tags, to generate healthcare recommendations.

Integrating VAEs and other models. The methods listed above mainly consider combining VAEs and PMF. Xiao et al. [189], Deng et al. [38], He et al. [65] and Yi et al. [199] use an MLP or NCF [67] to calculate the preferences between users and items; these approaches consider the non-linear relationships between users and items, and combines them with the latent representation learned from side information by VAEs. Chen et al. [29] learn latent representations of the item contents with a VAE (specifically, item features extracted from the reviews), and then use them for item-based CF (specifically, feature-based similarity models) so as to generate recommendations. Chen et al. [25] use one VAE to extract user profiles and item representations from reviews, to ensure both of them are in a consistent latent semantic space, followed by fusing the learned latent representations with the Latent Factor Model (LFM) [89] to generate recommendations. Ji et al. [77] propose CVRank, which uses a VAE to learn the latent representations of items and integrates the latent representations into pairwise ranking-based

[8]User features and item features can be regarded as the content of users and items.

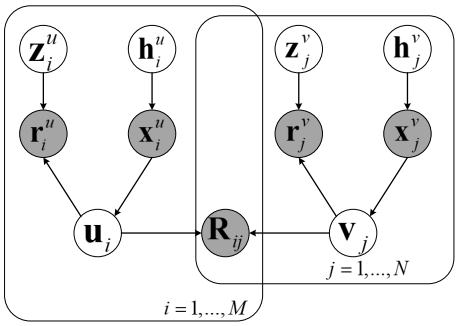


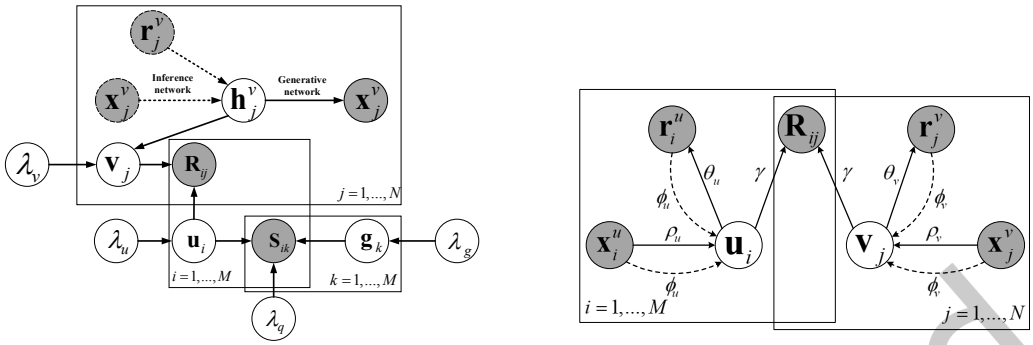
Fig. 9. A graphical model due to Xiao and Shen [191], in which four latent variables are used for inference. The shaded nodes are observed variables while the transparent nodes are latent. Note that $\mathbf{r}_i^u$, $\mathbf{x}_i^u$, $\mathbf{r}_j^v$, $\mathbf{x}_j^v$ are the user's interaction vector, user's feature vector, item's interaction vector, and the item's feature vector, respectively. The corresponding latent variables are $\mathbf{z}_i^u$, $\mathbf{h}_i^u$, $\mathbf{z}_j^v$, $\mathbf{h}_j^v$, respectively. $\mathbf{R}_{ij}$ is the rating or relevance between user i and item j.

CF for recommendation. Chen et al. [26] use a shared VAE to learn latent representations from reviews posted by users and received by items, respectively. Then the learned latent representations of reviews, together with the user and item latent representations, are sent to a GAN-based model to produce recommendations. Hansen et al. [60] introduce an approach known as NeuHash-CF. Here, a VAE-like model is used to generate the hash codes of users and items from learned embeddings and item contents. Hamming distance is used to estimate user-item relevance with the hash codes of users and items. Truong et al. [163] propose a bilateral variational autoencoder for CF (BiVAE), a flexible model that can use a MLP, dot product or any other differentiable function for estimating user-item relevance with the representations of user and item learned by VAEs.

Discussion. There are novel inference methods in some of the graphical models [e.g., 187, 189–191]: the interaction vector of user/item is sometimes used to infer the latent variable of side information. Xiao et al. [190] use an item interaction vector to infer a latent variable of the item side information (see the example in Figure 10(a)). Xiao and Shen [191, 192] and Xiao et al. [189] use the user and item interaction vector to infer the latent variable of side information of user/item, respectively (see the examples in Figure 9 and Figure 10(b)). Compared with the graphical model in [100] (Figure 8), the advantage of adding additional interaction information for inference is that side information and interaction information can be coupled for better recommendation performance. This novel inference method reveals that there are connections between side information and interaction information. To enhance the effectiveness of encoded features fusion, mutual dependency between latent variables learned from side information and interaction information, and contrastive learning could be explored further upon VAEs [180].

Similar to the static methods above, some dynamic methods just use a VAE to learn latent representations of side information instead of generating the recommendations. The encoding capability of VAEs is reflected in these methods.

Lin et al. [104] use a VAE to learn latent representations of image data (i.e., a cover image of each song's corresponding album) of a sequence of songs, which is fed into an RNN for next song recommendation. Sun and Qian [156] use VAEs to learn latent representations of category sequences in their proposed Tripled Seq2seq Translation Model (TSTM). Learned latent representations are used to infer the next recommended item. Song et al. [153] present an enhanced approach where a VAE is integrated into an RNN at each timestep. This improved RNN uses variational inference to capture the user's latent factor variables through a timestep-wise variational lower bound. It can capture the complex and hidden causal relationships in the current session so as to learn latent



(a) (b)

Fig. 10. Examples of graphical model using interaction vector to infer the latent variable. (a) uses only the item interaction vector [190] while (b) uses both of the user and item interaction vector [189]. In (a), $\mathbf{S}_{ik}$ is the ik-th element of the social matrix and $\mathbf{g}_k$ is social latent vector. In (b), $\mathbf{x}_i^u$ and $\mathbf{x}_j^v$ are item feature vector and user feature vector of side information. $\mathbf{r}_i^u$ and $\mathbf{r}_j^v$ denote the user's and item's interaction vector. The solid line refers to the generative process while the dashed line refers to the inference process. The shaded nodes refer to the observed variables.

representation of the session, which are further combined with short-term and long-term interest representations to predict the next top-n clicked items.

Discussion. When used for extracting side information, VAEs actually play the same role in dynamic models as in static models. In dynamic methods, the models that VAEs combine with are mainly RNNs and their variants.

3.1.3 Exploiting optimization strategies of VAEs for recommendation. While the methods in Section 3.1.1 and 3.1.2 directly use VAEs to learn the user-item interaction data or side information, here we discuss the approaches that only exploit optimization strategies of VAEs. Specifically, the optimization strategies of a VAE include: the reparameterization trick, the SGVB estimator, applying a neural network to approximate the true posterior distribution of the latent variable, and amortized inference (input-dependent encoder). In contrast to directly exploiting a VAE, these approaches may not be built based on VAEs. These approaches belong to the family of Bayesian models, and VAEs may be adopted to conduct efficient Bayesian inference for them.

To fuse side information and user-item interaction information, Shen et al. [148] propose a model named Deep Variational Matrix Factorization (DVMF), which consists of three sub-models, to integrate any types of side information of user/item, together with implicit feedback for recommendation. They first embed all the information into a user/item knowledge pool, then use two neural networks to encode user and item knowledge into latent distributions, and lastly sample the latent representations of users/items from the distributions to reconstruct the rating matrix for recommendation. When sampling latent representations of users/items, the reparameterization trick known from VAEs is used. Also, the SGVB estimator is derived to optimize the variational lower bound. Using a very similar model architecture as DVMF, Jin et al. [79] and Zhang et al. [209] use two neural networks to encode user-centric ratings and item-centric ratings into latent distributions of the user latent and item latent variables, respectively. The reparameterization trick is used to sample user/item latent representations from the latent distributions.

[9]http://2016.recsyschallenge.com/
[10]https://ijcai-15.org/index.php/repeat-buyers-prediction-competition

Table 6. Recommendation methods that exploit optimization strategies of VAEs. "Recommendation" is abbreviated as "Rec."

Paper	Year	Task description	Optimization strategies of VAEs	Dataset/domain
DVMF [148]	2019	Rec. by fusing knowledge embedding with U-I interaction	Reparameterization trick, SGVB	MovieLens, Douban
VAE-BMF [79]	2020	General Rec., the user's embeddings inference dependent on both interactions and the rated items embeddings	Reparameterization trick	Movielens, Amazon
DVMF [209]	2019	Rec. on large scale sparse dataset	Reparameterization trick	MovieLens-10M, Book-Crossing, Job
DGLGM [105]	2020	Rec. for users with diverse preference	Reparameterization trick, SGVB	MovieLens, Netflix, Epinions, Yelp
LRMM [43]	2019	Rec. with Side Informationl	Reparameterization trick, SGVB, Bayesian Inference	MovieLens, Netflix, Million Song dataset (MSD)
FAWMF [23]	2020	Rec. with Implicit Feedback	Variational posterior, ELBO	MovieLens, Amazon, Douban
LVM [141]	2019	Session-based Rec.	ELBO	YooChoose
HNVM [188]	2019	Sequential Recommendation with capturing long-term preference	Dedicated ELBO, SGVB	RecSys Challenge 2015,[9] IJCAI-15 Competition datasets[10]
VRNN-BPR [31]	2017	Session-based Rec.	Reparameterization trick, Bayesian inference	RecSys Challenge 2015
CVRCF [154]	2019	Steaming Rec.	Reparameterization trick, ELBO	MovieTweetings, MovieLens, Netflix

Liu et al. [105] propose a Deep Global and Local Generative Model (DGLGM) to characterize both the global and local structure among users. Specifically, under the *Wasserstein auto-encoder* framework, the *Beta-Bernoulli* distribution is introduced to model user-item interaction data, and a *Mixture Gaussian* distribution serves as the prior of the latent variable. The reparameterization trick of VAEs is used in this model to sample the implicit feedback data from the *Beta* distribution, and sample the latent representations from the variational Gaussian distributions parameterized by neural network for model optimization. Elahi et al. [43] propose a low rank multinomial model, which is similar to that in Mult-VAE [102], but with the difference that a linear operation, instead of a deep neural network, is used to reconstruct the interaction matrix. Likewise, the reparameterization trick is used to derive the SGVB estimator of the ELBO of the model, so that the model can be optimized using Bayesian inference.

While the above methods mainly use the reparameterization trick to derive a differentiable SGVB estimator from the ELBO, Chen et al. [23] apply amortized inference with a neural network to learn adaptive weights for weighted matrix factorization, overcoming the inefficiency of exposure-based matrix factorization [101].

Discussion. As evidenced by the methods introduced above, the reparameterization trick is widely used to optimize Bayesian models. Though the reparameterization trick is used to sample the latent representations for the neural network (decoder) in vanilla VAEs, in practice it can also be used in linear models [43]. Interestingly, the reparameterization trick is also applicable when sampling from other distributions than a Gaussian distribution.

The reparameterization trick ensures efficient Bayesian inference for these models, which facilitates handling sparse data.

Similarly, some dynamic methods do not directly use VAEs to model the generative process of item sequences with temporary dependencies. Instead, they resort to the optimization strategies of VAEs to optimize the proposed generative probabilistic model for item sequence generation. In what follows, these methods are introduced.

Rohde and Bonner [141] propose a session-based latent factor recommendation model that can model the evolution of the latent representation of users, as the user interacts with more items. The interaction sequence can be generated from a session-level latent variable, by multiplying an item embedding matrix and then adding bias. In this work, a VAE is used to optimize the model with the reparameterization trick, as well as limiting the number of parameters in the model.

Noticing that most session-based recommendation methods only consider the short session, for capturing user's short-term preference, but overlook long-term preferences, Xiao et al. [188] propose a hierarchical neural variational model to simultaneously capture the general, long-term and short-term preferences of the user, so as to address the next session items sequence prediction given the past set of sessions. Following a VAE, they propose a dedicated ELBO for their proposed model, and the reparameterization trick is applied to derive the SGVB estimator, so that SGD or Adam can be used to optimize the model (see Section 2.2.1). Christodoulou et al. [31] propose the Variational Recurrent Neural Network for session-based recommendation using Bayesian Personalized Ranking (VRNN-BPR) for session-based recommendation. They combine RNNs and pairwise ranking to formulate their model with Bayesian inference; the reparameterization trick is used to make the objective function of VRNN-BPR differentiable.

Song et al. [154] address the streaming recommendation problem with deep Bayesian learning. They propose a model called Coupled Variational Recurrent Collaborative Filtering (CVRCF), which is updated after setting time intervals to handle data dynamics, i.e., the evolution of user preferences and item popularity. This work considers sequential recommendation; CF is achieved by popular factorization-based approaches, and temporal dependencies are encoded by the proposed *coupled variational gated recurrent network*. Following a VAE, the framework uses a deep neural network to approximate the posterior of the latent variable flexibly, and a variant ELBO of the model is derived, which is transformed to a differentiable SGVB estimator through the reparameterization trick.

Discussion. The reparameterization trick is widely used in dynamic recommendation methods to help optimize the designed generative models, though the data are sequential data. In addition, as with vanilla VAEs, a deep neural network is applied to parameterize the latent distribution of the latent variable in sequential data.

Combining the application of VAEs in static methods and dynamic recommendation methods, we can conclude that VAEs are applicable to different types of data in a range of recommendation scenarios. Moreover, VAEs facilitate the combination of deep learning and graphical models, enhancing the model effectiveness and our understanding of inference uncertainties [154].

3.2 Other VAE-based Recommendation Methods

In this section, we introduce other types of VAE-based recommendation methods in terms of the characteristics of VAE that we summarized in Table 2. Note that these methods still use VAEs but focus on other types of recommendations or facets of RSs, e.g., (un)fairness problems.

3.2.1 Recommendations based on the encoding capabilities of VAEs.

Multi-criteria recommendation. Multi-criteria recommendation considers using multi-criteria ratings, a vector of ratings provided by users on several criteria, e.g., item attributes, to make recommendations [2]. Current multi-criteria recommendation methods mainly explicitly collect the multi-criteria ratings for recommendation. But they face several problems: (i) the collected multi-criteria ratings are predefined as low-dimensional criteria,

which limits the ability to express the multiplicity of user experiences; and (ii) some multi-criteria ratings might be missing. To tackle these problems, Li and Tuzhilin [99] use a VAE to project user reviews into a latent continuous space, followed by using embedding compression techniques to compress the obtained embeddings from VAEs into latent multi-criteria ratings. The latent multi-criteria ratings can overcome the aforementioned problems. The use of VAEs for encoding the user reviews into latent representations is a key step of this work.

Item link prediction. In real-world recommendation scenarios, we should not only consider interactions between users and items, but also the relationships between items. Rakesh et al. [134] address the link prediction task between items. Specifically, they propose a model called Linked Variational Autoencoder (LVA), including two VAEs and a connector neural network, taking the reviews of two items as input, to capture the substitute and supplementary relationships between the input items. VAEs are used to learn the latent representations of the item reviews, based on which the connector neural network can make prediction of the relationships between two items.

Query-based recommendation. Altaf et al. [5] deal with query-based dataset recommendation: given the query papers that describe the research interest of user, the RS should recommend to its user datasets that are semantically relevant to the query papers. Unlike baseline methods that recommend datasets directly linked to the query papers, they propose an extended VAE, Heterogeneous Variational Graph AutoEncoder (HVGAE) (see Figure 11), to learn the representations of the papers and datasets. Query-based recommendations are produced based on the learned representations. The advantage of applying the representations learned by a VAE to datasets recommendation is that the semantics of datasets can be considered. Wang et al. [172] identify and study the problem of gradient item recommendation and retrieval given an input query from a user. They define the problem as recommending a sequence of items with a gradual change on a certain attribute, given an input query item and a modification text. To address the problem, they propose a VAE-based weakly-supervised method that can learn a disentangled item representation from user-item interaction data and ground the semantic meaning of attributes to dimensions of the item representation for recommending items given the query, i.e., the input item and a modification of the item.

Disentanglement learning. VAEs are also used to learn disentangled representations from a user-item interaction matrix, to bring enhanced robustness, interpretability, and controllability [115, 147]. Here, disentanglements includes macro and micro disentanglements. Macro disentanglement aims to infer a user's intent, e.g., whether the user wants to buy shoes or clothes. Micro disentanglement aims to infer a user's preference towards aspects of items, e.g., shape and color. Macro disentanglement is achieved by carefully designing the latent representations of users, i.e., using a matrix consisted of preference vectors towards different intents as the latent representation. Micro disentanglement is realized by penalizing the KL divergence terms derived from the original KL divergence term of the variational lower-bound. Surprisingly, disentanglement learning also improves the recommendation performance at the same time. Ma et al. [115] use VAEs to learn the latent representation from the user-item interaction data, and as a generative model to learn the user's preference pattern. Also, reformulating the ELBO improves the micro disentanglement.

Discussion. The advantage of applying VAEs to the tasks listed above lies in the robust representations, it can learn and the fact that they can handle multiple types of data [90, 95, 111, 200]. For instance, VAEs are used to learn representations of reviews/item content in [8, 99, 134], to deal with multi-criteria recommendation and item link prediction, and to learn non-Euclidean distance representations of graphs in [5, 96–98] to address query-based recommendation [172].

3.2.2 Recommendations based on the generative nature of VAEs.

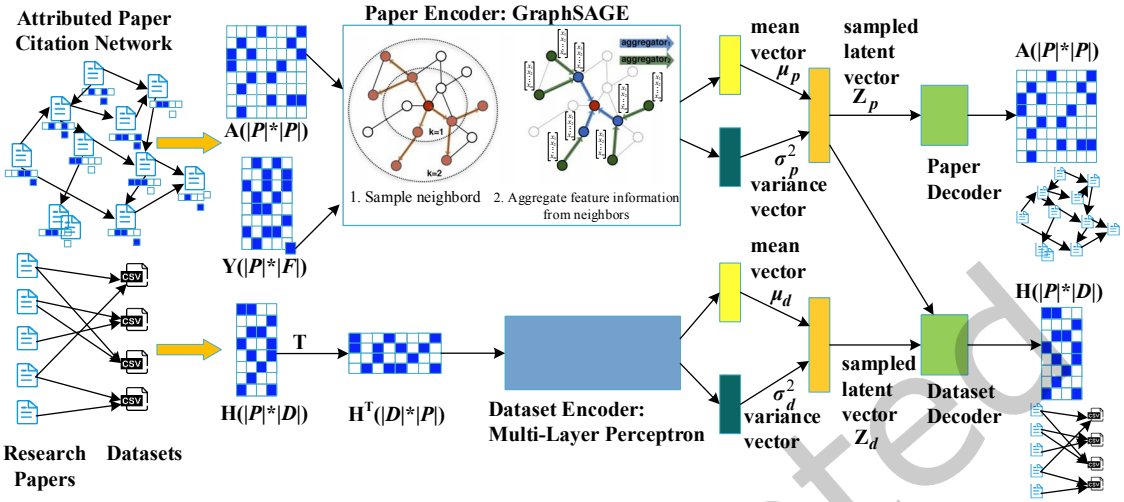


Fig. 11. Illustration of HVGAE [5]. There are two VAEs and on the top is an extended VAE with GraphSAGE [59], a type of graph neural network (GNN) as encoder. $|P|$, $|D|$ and $|F|$ are the number of papers, datasets and features, respectively. **A**, **Y** and **H** are adjacency matrix of paper citation graph, paper content matrix with $|F|$ features and matrix of paper-dataset bipartite network, respectively.

Cross-domain recommendation methods. These methods use knowledge from different domains to deal with data sparsity and cold start problems. The key to cross-domain recommendation is knowledge transfer from a source domain to a target domain. Nguyen and Ishigaki [122] use two VAEs to model the generative processes of implicit feedback data in the source domain and target domain, respectively, to simultaneously capture the homogeneous and varying features from source domain and target domain, and construct a bi-directional mapping between them. The VAEs share the weights of the last few layers of the encoders, and the first few layers of the two decoders. Weight sharing allows one to manipulate the generative process. Unlike [122], Ahangama and Poo [4] assume that the source domain and target domain are asymmetric, i.e., knowledge from the source domain is contributed to the target domain to obtain a better recommendation in the target domain (but not the other way around). Two VAEs are used to model the generative process of implicit feedback data in the source domain and target domain. Knowledge transfer from the source domain to the target domain is achieved by pulling the latent representations in the two VAEs as close as possible.

For the approach in [151], VAEs serve a different role. Shi and Wang [151] propose a Cross-Domain Variational AutoEncoder (CDVAE) for cross-domain recommendation. Matrix factorization is used to factorize the user-item rating matrix into a user latent matrix and item latent matrix in both domains. A VAE acts as a mapping from users' latent representations in the source domain to the target domain. This is where the generative nature of VAEs is used, as it generates the latent representations in the target domain. Similarly, Qian et al. [132] use VAEs to generate preference embeddings from attribute embeddings (related to the attributes of an item) to address the cold-start problem. The advantage of applying VAEs to cross-domain recommendation is that knowledge transfer can naturally be easily implemented as a generative process.

Slate recommendation. Exploiting the generative nature of VAEs, Jiang et al. [78] address the slate recommendation problem. Here, subsets of items, each constituting an ordered list of items that is meant to best serve a user's

preferences, should be recommended. Different subsets of items may cover different perceived purposes or intents of the user. The authors propose a generative model called List Conditional Variational AutoEncoders (List-CVAE) to directly generate optimal slates for user, instead of ranking the items. By directly generating slates, the model can reduce the computational complexity. The authors use a CVAE to model the distributions of all items in the same slate, conditioned on the user responses. The positional, contextual information is encoded into the latent space, and the optimal slate is generated by sending the combination of the condition (user's responses) and a sample from the latent space to the learned generative model (decoder).

Generating content for recommendation. Traditional recommendation methods recommend items that are available in an item pool to users. Some organizations, such as mobile-phone manufacturers and online magazines, are eager to receive the suggestions by RSs on what new items should be created to meet diverse preferences of users. Vo and Soh [168] use a VAE to simultaneously project user-item rating data and item features to a common latent space, where a greedy weighted maximum coverage method is adopted to select the latent representations to be used to generate new items with high predicted probability. In addition to generating new items, a VAE is also used to generate interactive text or explanations for recommended items. Zhang et al. [207] use VAEs to act as a generator to generate text for text-based interactive recommendation. Luo et al. [113] use a VAE framework to generate keyphrase-based explanations of recommendations; they allow users to critique the generated explanations to refine their personalized recommendations.

Discussion. By exploiting the generative nature of VAEs, novel recommendation tasks can be addressed, such as the slate recommendation and item generation. The key is to add different manipulations to the generative process of the data, so that VAEs can be adapted to different tasks.

3.2.3 Recommendations based on other characteristics of VAE. RSs usually provide users with a list of items that is ranked by the predicted user preferences [130]. However, many recommendation methods encounter a ranking fairness issue [42, 133, 202], where "ranking unfairness" refers to a situation in which items of similar or nearly identical relevance might be placed in varying positions within the ranking, potentially receiving vastly different degrees of attention. This promotes unfair treatment and makes items with similar scores have unequal opportunities of being presented to users. Borges and Stefanidis [16] address this unfairness problem using a VAE-based recommendation approach, i.e., Mult-VAE [102]. The authors incorporate noise in the VAE during testing, specifically when using the reparameterization trick to generate samples of the latent variable. This enables the model to vary the output scores even when having the same data as input, so that some items will not be always given the high indexes in sequential rounds of ranking. Borges and Stefanidis [17] aim at addressing the popularity bias problem [1] that promotes unfair recommendation results. To address the problem, they define a metric for evaluating popularity bias in recommendation results that reflect the presence of popular products ranked within the top-n positions, and propose a recommendation method that is able to mitigate the popularity bias in RSs based on VAEs, specifically, by modifying the ELBO of a VAE (one of the flexible internal structure). Gupta et al. [58] follow an alternative strategy and introduce an inverse propensity scoring (IPS) based unbiased training method for VAEs from implicit feedback data, VAE-IPS, which is provably unbiased w.r.t. selection bias; their experimental results show that the proposed VAE-IPS model reaches significantly higher performance than existing baselines. Adversarial training is further incorporated into VAEs, to remedy unfairness and privacy concerns by removing demographic biases and specific protected information of users from the learned interaction representations [48].

Meng et al. [118] propose the Variational Bayesian Context-Aware Representation (VBCAR) model to learn user and item latent representations by using basket context information from past user-item interactions to deal with grocery recommendation. A VAE is used to optimize the proposed model, i.e., the reparameterization trick is used

to make the ELBO differentiable. Compared with traditional methods that deal with grocery recommendation, such a Bayesian model can learn more expressive latent representations.

Discussion. Though some of the recommendation methods [5, 17, 115, 118] listed above do not explicitly work with VAEs, they do modify the ELBO, use the reparameterization trick, and adopt a deep neural network to approximate the intractable posterior of latent variables, which form the core of a VAE. The deep neural network can be replaced depending on the data, which confirms the flexibility of VAEs.

Despite the merits of VAEs listed above, there are some disadvantages and limitations of VAEs in VAEs: (i) Higher order interactions cannot be explicitly propagated as a graph neural network. To learn more complicated relationship, some researchers turn to combinations of VAEs and GNNs [109]. (ii) Tuning β is challenging for VAEs; β is used to make trade-offs between posterior collapse and the hole problem, i.e., the mismatch between the aggregated posterior distribution and the prior distribution. To solve this problem, some researchers [206] proposed a novel regularization method. (iii) Setting priors is critical. To break the limitations of fixed priors, some researchers have proposed non-parametric methods [107]. (iv) The number of parameters and the computational costs of VAEs increase linearly with the large item space. To reduce the complexity of training, some researchers employ a field-aware model [44], dynamic hash tables, or an inner-product-based softmax function [22] to improve the efficiency when facing high-dimensional data.

4 FUTURE DIRECTIONS AND NEW PERSPECTIVES

In this section, future directions and new perspectives of applying VAEs in RSs are given, to encourage further research into using VAEs to solve problems in RSs. Specifically, we first point out future directions based on the characteristics of VAEs in RSs. Additionally, we provide novel perspectives on applying VAEs in explainable and reliable RSs.

4.1 Future directions based on the characteristics of VAEs

Extending encoding capability for heterogeneous data. In real world scenarios, items and users are usually related to different types of information available on the web. Thus, in future work, we can consider encoding diverse types data, other than textual data in VAE-based recommendation models, e.g., images. How to take advantage of the encoding capability to encode heterogeneous data, and fuse them seamlessly either in observed space or latent space using a VAE so as to improve the recommendation performance, is a challenging but promising direction.

Exploiting the generative nature for diversity and explainability. VAEs provide a natural way for extending current RSs. The generative nature of VAEs could be used to generate some previously unseen items, to increase diversity of recommendations. In addition to generating interaction data, in future work, the generative nature could help to generate personalized explanations [94] of recommendations in the form of text for each user, increasing trust in the RS. We will detail this in next subsection.

Exploring the Bayesian nature with sophisticated dependencies among variables. One can explore more dependencies between different variables, since the architecture of the graphical model will influence the recommendation performance as stated by Lee et al. [93]. We suggest to consider adding more latent variables to the graphical model, to improve the recommendation performance. Prior work [116, 136, 144, 185] has added more variables to improve the model. After designing the model's causal structure, devising the most appropriate implementation for each type of graphical model is challenging, at the same time an exciting research direction.

Adjusting the flexible internal structure adaptively. Although the use of neural networks for approximating the intractable posterior of latent variables has seen great improvements, in reducing the gap between the true

posterior distribution and the latent distribution, there is always work to do. A reasonable encoder can improve the quality of the learned representations, which is crucial for improving the recommendation performance. Instead of simply deepening the encoder neural network, it would be better to determine the depth according to the number of user's interactions. To alleviate the so-called "posterior collapse" problem and learn better latent representations, some methods listed in Section 3.1.1 propose to use more reasonable priors. One can also resort to other priors such as autoregressive priors [27], Gaussian mixture priors [40], a nested Chinese Restaurant Process prior [55], a stick-breaking prior [121], hierarchical priors [88], or the prior aggregated by posteriors [120] like that in [83]. We suggest to add more user-specific information into the priors, to make the learned latent distributions of different users diverse.

4.2 Explainable and reliable RSs with VAEs

Based on the surveyed papers, we give some perspectives on how VAE can be used in explainable RSs.

Disentanglement for explainable RSs. In the surveyed papers, there is a work [115] that explores using VAEs to learn disentangled representations for recommendation, including the macro and micro disentanglements mentioned in Section 3.2.1. There is significant room for research into adopting disentangled learning in RSs, to make the recommendation results explainable, whereas only the interaction behavior data is used in [115]. With various side information (e.g., text, image, social relations, statistical histograms), VAEs could be extended to extract features from the multimodal data, disentangle the multimodal features and align the disentangled features cross different modalities. To disentangle the latent representation, i.e., to make latent representation meaningful for explaining a user's preference, intent and behavior are crucial for supporting users' understanding of, and trust in, RSs. Explainability is another important factor of the RSs. Providing interpretable output empowers RSs to elucidate their suggestions and enhance algorithmic transparency. Yet, there are relatively few publications that use disentanglement to make RSs explainable, though there is some work [e.g., 18, 19, 24, 72, 117] that adopts VAEs for disentangled representation learning in other research areas, e.g., computer vision. Thus, we believe that the use of VAEs to learn disentangled representations to make RSs explainable is a promising future direction.

Explainable textual suggestions. In RSs, it is a wise choice to generate textual suggestions to explain to the user why the system makes such recommendations [159]. Cui et al. [33] generate bi-directional predictions (predicting implicit feedback given review text or predicting review text given implicit feedback), which provides us with a way of achieving an explainable RS with suggestions. Specifically, using the generative nature of VAEs, we can train an additional VAE to generate suggestions for users, so as to make the RSs more convincing. We also hypothesize that they may be used to generate counterfactual explanations, which can help users understand not only why they received certain recommendations, but also how these recommendations can be changed [cf., e.g., 112] if we modify user's preferences or traits (e.g., demographics).

Denoising for reliable RSs. Modern RSs mainly rely on implicit feedback, which usually includes noisy interactions. Specifically, it is a long-standing challenge for VAEs to encode user representation from the sparse and flawed data. Though there is work revisiting the reliability of interactions from interactions from positive and negative samples simultaneously [108], it leaves much space to explore from other perspectives, such as cross-modal validation [178], addressing clickbait issue [173], self-guided learning [50], and so on.

5 CONCLUSIONS

In this survey, we provide a timely and systematical review of the research efforts on VAE-based recommendation. Specifically, we investigated a large number of related papers and summarized existing research in terms of the four key characteristics: their encoding capability, their generative nature, their Bayesian nature, and their

flexible internal structure. We have grouped the related algorithms using our proposed taxonomy (Figure 3), introducing how, specifically, VAEs are used in each of these recommendation methods. We have considered two main recommendation scenarios, a static scenario and a dynamic scenario, and introduced the corresponding VAE-based methods. In both scenario, we have summarized the ways in which VAEs are used in the respective methods, i.e., directly applying VAEs to generate recommendation results, extracting side information using VAEs, and exploiting optimization strategies that come with a VAE. In addition, we covered several recommendation scenarios that have only recently been introduced and to which VAE-based methods have been applied. Lastly and importantly, we have also pointed out promising directions for future research in the use of VAEs for recommender systems.

We hope that this survey has helped to explain how VAEs can be used in recommender systems, and we encourage future research aimed at using VAEs to tackle an even wider range of problems in recommender systems.

ACKNOWLEDGMENTS

We are very grateful to our reviewers for the constructive feedback, suggestions, and patience.

This work was supported by the National Natural Science Foundation of China (U2001211, U22B2060,62276279), Research Foundation of Science and Technology Plan Project of Guangzhou City (2023B01J0001, 2024B01W0004), Zhuhai City Industry-University-Research Project (2320004002797, 2220004002549), the Technology Innovation Center for Collaborative Applications of Natural Resources Data in GBA, MNR (No. 2024NRDZ02) and Pazhou Lab. This research was (partially) funded by the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl, by project LESSEN with project number NWA.1389.20.183 of the research program NWA ORC 2020/21, which is (partly) financed by the Dutch Research Council (NWO), by project ROBUST with project number KICH3.LTP.20.006, which is (partly) financed by the Dutch Research Council (NWO), DPG Media, RTL, and the Dutch Ministry of Economic Affairs and Climate Policy (EZK) under the program LTP KIC 2020-2023, and by the FINDHR (Fairness and Intersectional Non-Discrimination in Human Recommendation) project that received funding from the European Union's Horizon Europe research and innovation program under grant agreement No. 101070212.

All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning-to-rank Recommendation. In *RecSys'17*. 42–46.

[2] Gediminas Adomavicius, Nikos Manouselis, and YoungOk Kwon. 2011. Multi-criteria Recommender Systems. In *Recommender Systems Handbook*. Springer, 769–803.

[3] M Mehdi Afsar, Trafford Crump, and Behrouz Far. 2021. Reinforcement Learning Based Recommender Systems: A Survey. *arXiv preprint arXiv:2101.06286* (2021).

[4] Sapumal Ahangama and Danny Chiang-Choon Poo. 2019. Latent User Linking for Collaborative Cross Domain Recommendation. *arXiv preprint arXiv:1908.06583* (2019).

[5] Basmah Altaf, Uchenna Akujuobi, Lu Yu, and Xiangliang Zhang. 2019. Dataset Recommendation via Variational Graph Autoencoder. In *ICDM'19*. IEEE, 11–20.

[6] Bahare Askari, Jaroslaw Szlichta, and Amirali Salehi-Abari. 2021. Variational Autoencoders for Top-k Recommendation with Implicit Feedback. In *SIGIR'21*. 2061–2065.

[7] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer Normalization. *arXiv preprint arXiv:1607.06450* (2016).

[8] Haoli Bai, Zhuangbin Chen, Michael R Lyu, Irwin King, and Zenglin Xu. 2018. Neural Relational Topic Models for Scientific Article Analysis. In *CIKM'18*. 27–36.

[9] Jinxin Bai and Zhijie Ban. 2019. Collaborative Multi-Auxiliary Information Variational Autoencoder for Recommender Systems. In *ICMLC'19*. 501–505.
[10] Zeynep Batmaz, Ali Yurekli, Alper Bilge, and Cihan Kaleli. 2019. A Review on Deep Learning for Recommender Systems: Challenges and Remedies. *Artificial Intelligence Review* 52, 1 (2019), 1–37.
[11] Vito Bellini, Tommaso Di Noia, Eugenio Di Sciascio, and Angelo Schiavone. 2019. Semantics-Aware Autoencoder. *IEEE Access* 7 (2019), 166122–166137.
[12] James Bennett and Stan Lanning. 2007. The Netflix Prize. In *Proceedings of KDD Cup and Workshop*. ACM, 35.
[13] James S. Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for Hyper-parameter Optimization. In *NeurIPS*. 2546– 2554.
[14] Basiliyos Tilahun Betru, Charles Awono Onana, and Bernabe Batchakui. 2017. Deep Learning Methods on Recommender System: A Survey of State-of-the-Art. *International Journal of Computer Applications* 162, 10 (2017), 17–22.
[15] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3, Jan (2003), 993–1022.
[16] Rodrigo Borges and Kostas Stefanidis. 2019. Enhancing Long Term Fairness in Recommendations with Variational Autoencoders. In *MEDES'19*. 95–102.
[17] Rodrigo Borges and Kostas Stefanidis. 2021. On Mitigating Popularity Bias in Recommendations via Variational Autoencoders. In *SAC'21*. 1383–1389.
[18] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. 2018. Multi-level Variational Autoencoder: Learning Disentangled Representations from Grouped Observations. In *AAAI'18*.
[19] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding Disentangling in β-VAE. *arXiv preprint arXiv:1804.03599* (2018).
[20] Robin Burke. 2002. Hybrid Recommender Systems: Survey and Experiments. *User modeling and user-adapted interaction* 12, 4 (2002), 331–370.
[21] Junyang Chen, Ziyi Chen, Mengzhu Wang, Ge Fan, Guo Zhong, Ou Liu, Wenfeng Du, Zhenghua Xu, and Zhiguo Gong. 2023. A Neural Inference of User Social Interest for Item Recommendation. *Data Sci. Eng.* 8, 3 (2023), 223–233.
[22] Jin Chen, Defu Lian, Binbin Jin, Xu Huang, Kai Zheng, and Enhong Chen. 2022. Fast Variational AutoEncoder with Inverted Multi-Index for Collaborative Filtering. In *WWW*. 1944–1954.
[23] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Jingbang Chen, Yan Feng, and Chun Chen. 2020. Fast Adaptively Weighted Matrix Factorization for Recommendation with Implicit Feedback. *arXiv preprint arXiv:2003.01892* (2020).
[24] Tian Qi Chen, Xuechen Li, Roger B. Grosse, and David K. Duvenaud. 2018. Isolating Sources of Disentanglement in Variational Autoencoders. In *NeurIPS'18*. 2610–2620.
[25] Wang Chen, Hai-Tao Zheng, and Xiao-Xi Mao. 2017. Extracting Deep Semantic Information for Intelligent Recommendation. In *NeurIPS'17*. Springer, 134–144.
[26] Wang Chen, Hai-Tao Zheng, Yang Wang, Wei Wang, and Rui Zhang. 2019. Utilizing Generative Adversarial Networks for Recommendation based on Ratings and Reviews. In *IJCNN'19*. IEEE, 1–8.
[27] Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Variational Lossy Autoencoder. *arXiv preprint arXiv:1611.02731* (2016).
[28] Yifan Chen and Maarten de Rijke. 2018. A Collective Variational Autoencoder for Top-n Recommendation with Side Information. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*. 3–9.
[29] Yifan Chen, Yang Wang, Xiang Zhao, Hongzhi Yin, Ilya Markov, and Maarten de Rijke. 2020. Local Variational Feature-based Similarity Models for Recommending Top-N New Items. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–33.
[30] Minjin Choi, Yoonki Jeong, Joonseok Lee, and Jongwuk Lee. 2021. Local Collaborative Autoencoders. In *WSDM'21*. 734–742.
[31] Panayiotis Christodoulou, Sotirios P. Chatzis, and Andreas S. Andreou. 2017. A Variational Recurrent Neural Network for Session-Based Recommendations using Bayesian Personalized Ranking. In *ISD*.
[32] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. 2015. A Recurrent Latent Variable Model for Sequential Data. In *Advances in Neural Information Processing Systems*, Vol. 28. 2980–2988.
[33] Kenan Cui, Xu Chen, Jiangchao Yao, and Ya Zhang. 2018. Variational Collaborative Learning for User Probabilistic Representation. *arXiv preprint arXiv:1809.08400* (2018).
[34] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language Modeling with Gated Convolutional Networks. In *ICML'17*. PMLR, 933–941.
[35] Romain Deffayet, Thibaut Thonet, Jean-Michel Renders, and Maarten de Rijke. 2023. Generative Slate Recommendation with Reinforcement Learning. In *WSDM 2023: The Sixteenth International Conference on Web Search and Data Mining*. ACM, 580–588.
[36] Yashar Deldjoo, Markus Schedl, Paolo Cremonesi, and Gabriella Pasi. 2021. Recommender Systems Leveraging Multimedia Content. *ACM Comput. Surv.* 53, 5 (2021), 106:1–106:38.

[37] Xiaoyi Deng and Feifei Huangfu. 2019. Collaborative Variational Deep Learning for Healthcare Recommendation. *IEEE Access* 7 (2019), 55679–55688.

[38] Xiaoyi Deng, Fuzhen Zhuang, and Zhiguo Zhu. 2019. Neural Variational Collaborative Filtering with Side Information for Top-K Recommendation. *International Journal of Machine Learning and Cybernetics* 10, 11 (2019), 3273–3284.

[39] Utkarsh Desai, Sambaran Bandyopadhyay, and Srikanth Tamilselvam. 2021. Graph Neural Network to Dilute Outliers for Refactoring Monolith Application. In *AAAI'21*. 72–80.

[40] Nat Dilokthanakul, Pedro A.M. Mediano, Marta Garnelo, Matthew C.H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. 2016. Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders. *arXiv preprint arXiv:1611.02648* (2016).

[41] Carl Doersch. 2016. Tutorial on Variational Autoencoders. *arXiv preprint arXiv:1606.05908* (2016).

[42] Michael D. Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. 2022. Fairness in Information Access Systems. *Found. Trends Inf. Retr.* 16, 1-2 (2022), 1–177.

[43] Ehtsham Elahi, Wei Wang, Dave Ray, Aish Fenton, and Tony Jebara. 2019. Variational Low Rank Multinomials for Collaborative Filtering with Side-information. In *RecSys'19*. 340–347.

[44] Ge Fan, Chaoyun Zhang, Junyang Chen, Baopu Li, Zenglin Xu, Yingjie Li, Luyu Peng, and Zhiguo Gong. 2022. Field-aware Variational Autoencoders for Billion-scale User Representation Learning. In *ICDE*. 3413–3425.

[45] Jinyuan Fang, Shangsong Liang, Zaiqiao Meng, and Maarten de Rijke. 2021. Hyperspherical Variational Co-embedding for Attributed Networks. *ACM Transactions on Information Systems (TOIS)* 41, 2 (2021), 1–36.

[46] Weite Feng, Tong Li, Haiyang Yu, and Zhen Yang. 2021. A Hybrid Music Recommendation Algorithm Based on Attention Mechanism. In *MMM'21*. Springer, 328–339.

[47] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain Recommender Systems: A Survey of the State of the Art. In *CERI'12*. sn, 1–12.

[48] Christian Ganhor, David Penz, Navid Rekabsaz, Oleg Lesota, and Markus Schedl. 2022. Unlearning Protected User Attributes in Recommendations with Adversarial Training. In *SIGIR*. 2142–2147.

[49] Jianliang Gao, Xiaoting Ying, Cong Xu, Jianxin Wang, Shichao Zhang, and Zhao Li. 2021. Graph-Based Stock Recommendation by Time-Aware Relational Attention Network. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16, 1 (2021), 1–21.

[50] Yunjun Gao, Yuntao Du, Yujia Hu, Lu Chen, Xinjun Zhu, Ziquan Fang, and Baihua Zheng. 2022. Self-Guided Learning to Denoise for Robust Recommendation. In *SIGIR*. 1412–1422.

[51] Kostadin Georgiev and Preslav Nakov. 2013. A Non-iid Framework for Collaborative Filtering with Restricted Boltzmann Machines. In *ICML'13*. 1148–1156.

[52] Samuel Gershman and Noah Goodman. 2014. Amortized Inference in Probabilistic Reasoning. In *36th Annual Meeting of the Cognitive Science Society (CogSci 2014)*, Vol. 36. 517–522.

[53] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT press.

[54] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NeurIPS'14*. 2672–2680.

[55] Prasoon Goyal, Zhiting Hu, Xiaodan Liang, Chenyu Wang, and Eric P. Xing. 2017. Nonparametric Variational Auto-encoders for Hierarchical Representation Learning. In *ICCV'17*. 5094–5102.

[56] Sven Gronauer and Klaus Diepold. 2022. Multi-agent Deep Reinforcement Learning: A Survey. *Artificial Intelligence Review* 55, 2 (2022), 895–943.

[57] Kilol Gupta, Mukund Yelahanka Raghuprasad, and Pankhuri Kumar. 2018. A Hybrid Variational Autoencoder for Collaborative Filtering. *arXiv preprint arXiv:1808.01006* (2018).

[58] Shashank Gupta, Harrie Oosterhuis, and Maarten de Rijke. 2022. VAE-IPS: A Deep Generative Recommendation Method for Unbiased Learning From Implicit Feedback. In *CONSEQUENCES+REVEAL Workshop at RecSys '22*. ACM.

[59] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS'17*. 1024–1034.

[60] Casper Hansen, Christian Hansen, Jakob Grue Simonsen, Stephen Alstrup, and Christina Lioma. 2020. Content-aware Neural Hashing for Cold-start Recommendation. *arXiv preprint arXiv:2006.00617* (2020).

[61] Junmei Hao, Yujie Dun, Guoshuai Zhao, Yuxia Wu, and Xueming Qian. 2022. Annular-graph Attention Model for Personalized Sequential Recommendation. *IEEE Transactions on Multimedia* 24 (2022), 3381–3391.

[62] F. Maxwell Harper and Joseph A. Konstan. 2015. The Movielens Datasets: History and Context. *TiiS'15* 5, 4 (2015), 1–19.

[63] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked Autoencoders are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16000–16009.

[64] Ming He, Qian Meng, and Shaozong Zhang. 2019. Collaborative Additional Variational Autoencoder for Top-N Recommender Systems. *IEEE Access* 7 (2019), 5707–5713.

[65] Siyuan He, Tao Li, Yuxin Duan, Zhenning Yang, and Feixiang Li. 2019. VAE Based-NCF for Recommendation of Implicit Feedback. In *ITAIC'19*. IEEE, 512–516.

[66] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. ACM, 639–648.
[67] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW'17*. 173–182.
[68] Robert Hecht-Nielsen. 1992. Theory of the Backpropagation Neural Network. In *Neural Networks for Perception*. Elsevier, 65–93.
[69] Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. 2021. Explainability in Deep Reinforcement Learning. *Knowledge-Based Systems* 214 (2021), 1–14.
[70] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based Recommendations with Recurrent Neural Networks. *arXiv preprint arXiv:1511.06939* (2015).
[71] Irina Higgins, Loic Matthey, Xavier Glorot, Arka Pal, Benigno Uria, Charles Blundell, Shakir Mohamed, and Alexander Lerchner. 2016. Early Visual Concept Learning with Unsupervised Deep Learning. *arXiv preprint arXiv:1606.05579* (2016).
[72] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *Iclr* 2, 5 (2017), 6.
[73] Geoffrey E Hinton. 2009. Deep Belief Networks. *Scholarpedia* 4, 5 (2009), 5947.
[74] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *CVPR'17*. 4700–4708.
[75] Murium Iqbal, Kamelia Aryafar, and Timothy Anderton. 2019. Style Conditioned Recommendations. In *RecSys'19*. 128–136.
[76] Dietmar Jannach, Massimo Quadrana, and Paolo Cremonesi. 2020. Recommender Systems Leveraging Multimedia Content. *Comput. Surveys* 53, 5 (2020), Article No: 106.
[77] Li Ji, Guangyan Lin, and Huobin Tan. 2018. Neural Collaborative Filtering: Hybrid Recommendation Algorithm with Content Information and Implicit Feedback. In *IDEAL'18*. Springer, 679–688.
[78] Ray Jiang, Sven Gowal, Timothy A Mann, and Danilo J Rezende. 2018. Beyond Greedy Ranking: Slate Optimization via list-CVAE. *arXiv preprint arXiv:1803.01682* (2018).
[79] Yuan Jin, He Zhao, Ming Liu, Lan Du, Yunfeng Li, Ruohua Xu, and Longxiang Gao. 2020. Leveraging Cross Feedback of User and Item Embeddings for Variational Autoencoder based Collaborative Filtering. *arXiv preprint arXiv:2002.09145* (2020).
[80] Giannis Karamanolakis, Kevin Raji Cherian, Ananth Ravi Narayan, Jie Yuan, Da Tang, and Tony Jebara. 2018. Item Recommendation with Variational Autoencoders and Heterogeneous Priors. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*. 10–14.
[81] Muhammad Murad Khan, Roliana Ibrahim, and Imran Ghani. 2017. Cross Domain Recommender Systems: A Systematic Literature Review. *ACM Computing Surveys (CSUR)* 50, 3 (2017), 1–34.
[82] Zahid Younas Khan, Zhendong Niu, Sulis Sandiwarno, and Rukundo Prince. 2021. Deep Learning Techniques for Rating Prediction: A Survey of the State-of-the-art. *Artificial Intelligence Review* (2021), 95–135.
[83] Daeryong Kim and Bongwon Suh. 2019. Enhancing VAEs For Collaborative Filtering: Flexible Priors & Gating Mechanisms. In *RecSys'19*. 403–407.
[84] Jeeyung Kim. 2019. Time-varying Item Feature Conditional Variational Autoencoder for Collaborative Filtering. In *Big Data'19*. IEEE, 2309–2316.
[85] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).
[86] Diederik P. Kingma and Max Welling. 2013. Auto-encoding Variational Bayes. *arXiv preprint arXiv:1312.6114* (2013).
[87] Diederik P. Kingma and Max Welling. 2019. An Introduction to Variational Autoencoders. *Foundations and Trends in Machine Learning* 12, 4 (2019), 307–392.
[88] Alexej Klushyn, Nutan Chen, Richard Kurle, Botond Cseke, and Patrick van der Smagt. 2019. Learning Hierarchical Priors in VAEs. In *NeurIPS'19*. 2866–2875.
[89] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.
[90] Adit Krishnan, Mahashweta Das, Mangesh Bendre, Hao Yang, and Hari Sundaram. 2020. Transfer Learning via Contextual Invariants for One-to-Many Cross-Domain Recommendation. *arXiv preprint arXiv:2005.10473* (2020).
[91] R. Lavanya, Utkarsh Singh, and Vibhor Tyagi. 2021. A Comprehensive Survey on Movie Recommendation Systems. In *ICAIS'21*. 532–536.
[92] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep Learning. *Nature* 521, 7553 (2015), 436–444.
[93] Wonsung Lee, Kyungwoo Song, and Il-Chul Moon. 2017. Augmented Variational Autoencoders for Collaborative Filtering with Auxiliary Information. In *CIKM'17*. 1139–1148.
[94] Chunying Li, Bingyang Zhou, Weijie Lin, Zhikang Tang, Yong Tang, Yanchun Zhang, and Jinli Cao. 2023. A Personalized Explainable Learner Implicit Friend Recommendation Method. *Data Sci. Eng.* 8, 1 (2023), 23–35.
[95] Jiyong Li, Dilshod Azizov, LI Yang, and Shangsong Liang. 2024. Contrastive Continual Learning with Importance Sampling and Prototype-Instance Relation Distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 13554–13562.

[96] Jingci Li, Guangquan Lu, and Jiecheng LI. 2022. A Self-supervised Graph Autoencoder with Barlow Twins. In *PRICAI 2022: Trends in Artificial Intelligence*, Sankalp Khanna, Jian Cao, Quan Bai, and Guandong Xu (Eds.). Springer Nature Switzerland, 501–512.

[97] Jingci Li, Guangquan Lu, and Zhengtian Wu. 2022. Multi-View Graph Autoencoder for Unsupervised Graph Representation Learning. In *2022 26th International Conference on Pattern Recognition (ICPR)*. 2213–2218.

[98] Jingci Li, Guangquan Lu, Zhengtian Wu, and Fuqing Ling. 2023. Multi-view Representation Model Based on Graph Autoencoder. *Information Sciences* 632 (2023), 439–453. https://doi.org/10.1016/j.ins.2023.02.092

[99] Pan Li and Alexander Tuzhilin. 2019. Latent Multi-criteria Ratings for Recommendations. In *RecSys'19*. 428–431.

[100] Xiaopeng Li and James She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *SIGKDD'17*. 305–314.

[101] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling User Exposure in Recommendation. In *WWW'16*. 951–961.

[102] Dawen Liang, Rahul G Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *WWW'18*. 689–698.

[103] Jason Liang and Keith Kelly. 2021. Training Stacked Denoising Autoencoders for Representation Learning. In *arXiv preprint arXiv:2102.08012*. 1–16.

[104] Qika Lin, Yaoqiang Niu, Yifan Zhu, Hao Lu, Keith Zvikomborero Mushonga, and Zhendong Niu. 2018. Heterogeneous Knowledge-based Attentive Neural Networks for Short-term Music Recommendations. *IEEE Access* 6 (2018), 58990–59000.

[105] Huafeng Liu, Liping Jing, Jingxuan Wen, Zhicheng Wu, Xiaoyi Sun, Jiaqi Wang, Lin Xiao, and Jian Yu. 2020. Deep Global and Local Generative Model for Recommendation. In *WWW'20*. 551–561.

[106] Juntao Liu and Caihua Wu. 2017. Deep Learning based Recommendation: A Survey. In *ICISA'17*. Springer, 451–458.

[107] Wei Liu, Shangsong Liang, Huaijie Zhu, Leonghou U, Jianxing Yu, Xiang Li, and Jian Yin. 2024. Variational Kernel Density Estimation Recommendation Algorithm for Users with Diverse Activity Levels. In *DASFAA*.

[108] Wei Liu, Leong Hou U, Shangsong Liang, Huaijie Zhu, Jianxing Yu, Yubao Liu, and Jian Yin. 2023. Revisiting Positive and Negative Samples in Variational Autoencoders for Top-N Recommendation. In *DASFAA*. Springer, 563–573.

[109] Yang Liu, Qianzhen Rao, Weike Pan, and Zhong Ming. 2023. Variational Collective Graph AutoEncoder for Multi-behavior Recommendation. In *IEEE International Conference on Data Mining, ICDM 2023, Shanghai, China, December 1-4, 2023*. IEEE, 438–447.

[110] Sam Lobel, Chunyuan Li, Jianfeng Gao, and Lawrence Carin. 2019. Towards Amortized Ranking-Critical Training for Collaborative Filtering. *arXiv preprint arXiv:1906.04281* (2019).

[111] Guangquan Lu, Xishun Zhao, Jian Yin, Weiwei Yang, and Bo Li. 2020. Multi-task Learning Using Variational Auto-encoder for Sentiment Classification. *Pattern Recognition Letters* 132 (2020), 115–122.

[112] Ana Lucic, Harrie Oosterhuis, Hinda Haned, and Maarten de Rijke. 2022. FOCUS: Flexible Optimizable Counterfactual Explanations for Tree Ensembles. In *AAAI'22*. AAAI.

[113] Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. 2020. Deep Critiquing for VAE-Based Recommender Systems. In *SIGIR'20*. 1269–1278.

[114] Chao Ma, Wenbo Gong, José Miguel Hernández-Lobato, Noam Koenigstein, Sebastian Nowozin, and Cheng Zhang. 2018. Partial VAE for Hybrid Recommender System. In *NIPS Workshop on Bayesian Deep Learning*.

[115] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *NeurIPS'19*. 5712–5723.

[116] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. 2016. Auxiliary Deep Generative Models. *arXiv preprint arXiv:1602.05473* (2016).

[117] Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. 2018. Disentangling Disentanglement in Variational Autoencoders. *arXiv preprint arXiv:1812.02833* (2018).

[118] Zaiqiao Meng, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2019. Variational Bayesian Context-aware Representation for Grocery Recommendation. *arXiv preprint arXiv:1909.07705* (2019).

[119] Andriy Mnih and Russ R. Salakhutdinov. 2008. Probabilistic Matrix Factorization. In *NeurIPS'08*. 1257–1264.

[120] Dmitry Molchanov, Valery Kharitonov, Artem Sobolev, and Dmitry Vetrov. 2018. Doubly Semi-implicit Variational Inference. *arXiv preprint arXiv:1810.02789* (2018).

[121] Eric Nalisnick and Padhraic Smyth. 2016. Stick-breaking Variational Autoencoders. *arXiv preprint arXiv:1605.06197* (2016).

[122] Linh Nguyen and Tsukasa Ishigaki. 2018. Domain-to-Domain Translation Model for Recommender System. *arXiv preprint arXiv:1812.06229* (2018).

[123] Linh Nguyen and Tsukasa Ishigaki. 2019. Collaborative Multi-key Learning with an Anonymization Dataset for a Recommender System. In *IJCNN'19*. IEEE, 1–9.

[124] Juan Ni, Zhenhua Huang, Chang Yu, Dongdong Lv, and Cheng Wang. 2021. Comparative Convolutional Dynamic Multi-Attention Recommendation Model. *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[125] Xia Ning and George Karypis. 2012. Sparse Linear Methods with Side Information for Top-n Recommendations. In *RecSys'12*. 155–162.

[126] Yi Ouyang, Bin Guo, Xing Tang, Xiuqiang He, Jian Xiong, and Zhiwen Yu. 2021. Mobile App Cross-Domain Recommendation with Multi-Graph Neural Network. *ACM Trans. Knowl. Discov. Data*, Article 55 (2021), 21 pages.

[127] Bo Pang, Min Yang, and Chongjun Wang. 2019. A Novel Top-N Recommendation Approach Based on Conditional Variational Auto-Encoder. In *PAKDD'19*. Springer, 357–368.
[128] Saavan Patel, Philip Canoza, and Sayeef Salahuddin. 2022. Logically Synthesized and Hardware-accelerated Restricted Boltzmann Machines for Combinatorial Optimization and Integer Factorization. *Nature Electronics* 5, 2 (2022), 92–101.
[129] Jan Peters and Stefan Schaal. 2008. Natural Actor-critic. *Neurocomputing* 71, 7-9 (2008), 1180–1190.
[130] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. 2021. Fairness in Rankings and Recommendations: An Overview. *arXiv preprint arXiv:2104.05994* (2021).
[131] Mirko Polato, Tommaso Carraro, and Fabio Aiolli. 2020. Conditioned Variational Autoencoder for top-N item recommendation. *arXiv preprint arXiv:2004.11141* (2020).
[132] Tieyun Qian, Yile Liang, and Qing Li. 2019. Solving Cold Start Problem in Recommendation with Attribute Graph Neural Networks. *arXiv preprint arXiv:1912.12398* (2019).
[133] Amifa Raj and Michael D. Ekstrand. 2022. Measuring Fairness in Ranked Results: An Analytical and Empirical Comparison. In *SIGIR*. 726–736.
[134] Vineeth Rakesh, Suhang Wang, Kai Shu, and Huan Liu. 2019. Linked Variational Autoencoders for Inferring Substitutable and Supplementary Items. In *WSDM'19*. 438–446.
[135] Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2017. Searching for Activation Functions. *arXiv preprint arXiv:1710.05941* (2017).
[136] Rajesh Ranganath, Dustin Tran, and David Blei. 2016. Hierarchical Variational Models. In *ICML'16*. 324–333.
[137] Shamima Rashid, Suresh Sundaram, and Chee Keong Kwoh. 2022. Empirical Study of Protein Feature Representation on Deep Belief Networks Trained with Small Data for Secondary Structure Prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2022).
[138] Danilo Jimenez Rezende and Shakir Mohamed. 2015. Variational Inference with Normalizing Flows. *arXiv preprint arXiv:1505.05770* (2015).
[139] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *arXiv preprint arXiv:1401.4082* (2014).
[140] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to Recommender Systems Handbook. In *Recommender Systems Handbook*. Springer, 1–35.
[141] David Rohde and Stephen Bonner. 2019. Latent Variable Session-Based Recommendation. *arXiv preprint arXiv:1904.10784* (2019).
[142] Noveen Sachdeva, Giuseppe Manco, Ettore Ritacco, and Vikram Pudi. 2019. Sequential Variational Autoencoders for Collaborative Filtering. In *WSDM'19*. 600–608.
[143] Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Deep Boltzmann Machines. In *AISTATS'09*. 448–455.
[144] Tim Salimans, Diederik Kingma, and Max Welling. 2015. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. In *ICML'15*. 1218–1226.
[145] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative Filtering Recommender Systems. In *The Adaptive Web*. Springer, 291–324.
[146] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders Meet Collaborative Filtering. In *WWW'15*. 111–112.
[147] Anna Sepliarskaia, Julia Kiseleva, and Maarten de Rijke. 2021. How Not to Measure Disentanglement. In *ICML Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI*.
[148] Xiaoxuan Shen, Baolin Yi, Hai Liu, Wei Zhang, Zhaoli Zhang, Sannyuya Liu, and Naixue Xiong. 2019. Deep Variational Matrix Factorization with Knowledge Embedding for Recommendation System. *IEEE Transactions on Knowledge and Data Engineering* (2019).
[149] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I. Nikolenko. 2020. RecVAE: A New Variational Autoencoder for Top-N Recommendations with Implicit Feedback. In *WSDM'20*. 528–536.
[150] Heng-Shiou Sheu, Zhixuan Chu, Daiqing Qi, and Sheng Li. 2021. Knowledge-Guided Article Embedding Refinement for Session-Based News Recommendation. *IEEE Transactions on Neural Networks and Learning Systems* 33, 12 (2021), 7921–7927.
[151] Jiarui Shi and Quanmin Wang. 2019. Cross-Domain Variational Autoencoder for Recommender Systems. In *ICAIT'19*. IEEE, 67–72.
[152] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. 2016. Ladder Variational Autoencoders. In *NeurIPS'16*. 3738–3746.
[153] Jing Song, Hong Shen, Zijing Ou, Junyi Zhang, Teng Xiao, and Shangsong Liang. 2019. ISLF: Interest Shift and Latent Factors Combination Model for Session-based Recommendation. In *IJCAI'19*. 5765–5771.
[154] Qingquan Song, Shiyu Chang, and Xia Hu. 2019. Coupled Variational Recurrent Collaborative Filtering. In *SIGKDD'19*. 335–343.
[155] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence* 2009 (2009), Article ID 421425.
[156] Ke Sun and Tieyun Qian. 2019. Seq2seq Translation Model for Sequential Recommendation. *arXiv preprint arXiv:1912.07274* (2019).
[157] Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. 2016. Joint Multimodal Learning with Deep Generative Models. *arXiv preprint arXiv:1611.01891* (2016).

[158] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-based Recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. 17–22.
[159] Nava Tintarev and Judith Masthoff. 2015. Explaining Recommendations: Design and Evaluation. In *Recommender Systems Handbook*. Springer, 353–382.
[160] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. 2017. Wasserstein Auto-encoders. *arXiv preprint arXiv:1711.01558* (2017).
[161] Jakub M Tomczak and Max Welling. 2018. VAE with a VampPrior. In *AISTATS'18*. PMLR, 1214–1223.
[162] Yuzhen Tong, Yadan Luo, Zheng Zhang, Shazia Sadiq, and Peng Cui. 2019. Collaborative Generative Adversarial Network for Recommendation Systems. In *ICDEW'19*. IEEE, 161–168.
[163] Quoc-Tuan Truong, Aghiles Salah, and Hady W Lauw. 2021. Bilateral Variational Autoencoder for Collaborative Filtering. In *WSDM'21*. 292–300.
[164] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep Content-based Music Recommendation. In *NeurIPS'13*. 2643–2651.
[165] Vojtěch Vančura and Pavel Kordík. 2021. Deep Variational Autoencoder with Shallow Parallel Path for Top-N Recommendation (VASP). *arXiv preprint arXiv:2102.05774* (2021).
[166] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and Composing Robust Features with Denoising Autoencoders. In *ICML'08*. 1096–1103.
[167] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of machine learning research* 11, Dec (2010), 3371–3408.
[168] Thanh Vinh Vo and Harold Soh. 2018. Generation Meets Recommendation: Proposing Novel Items for Groups of Users. In *RecSys'18*. 145–153.
[169] Chong Wang and David M. Blei. 2011. Collaborative Topic Modeling for Recommending Scientific Articles. In *SIGKDD'11*. 448–456.
[170] Chao Wang, Hengshu Zhu, Chen Zhu, Xi Zhang, Enhong Chen, and Hui Xiong. 2020. Personalized Employee Training Course Recommendation with Career Development Awareness. In *WWW'20*. 1648–1659.
[171] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative Deep Learning for Recommender Systems. In *SIGKDD'15*. 1235–1244.
[172] Haonan Wang, Chang Zhou, Carl Yang, Hongxia Yang, and Jingrui He. 2021. Controllable Gradient Item Retrieval. In *WWW'21*. 1–10.
[173] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue. In *SIGIR*. 1288–1297.
[174] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023. Diffusion Recommender Model. *CoRR abs/2304.04971* (2023).
[175] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*. ACM, 165–174.
[176] Xinxi Wang and Ye Wang. 2014. Improving Content-based and Hybrid Music Recommendation Using Deep Learning. In *MM'14*. 627–636.
[177] Yang Wang and Lixin Han. 2021. Adaptive Time Series Prediction and Recommendation. *Information Processing & Management* 58 (2021). Issue 3.
[178] Yu Wang, Xin Xin, Zaiqiao Meng, Joemon M. Jose, Fuli Feng, and Xiangnan He. 2022. Learning Robust Recommenders through Cross-Model Agreement. In *WWW*. 2015–2025.
[179] Zhitao Wang, Chengyao Chen, Ke Zhang, Yu Lei, and Wenjie Li. 2018. Variational Recurrent Model for Session-based Recommendation. In *CIKM'18*. 1839–1842.
[180] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive Learning for Cold-Start Recommendation. In *Multimedia*. ACM, 5382–5390.
[181] Jess Whittlestone, Kai Arulkumaran, and Matthew Crosby. 2021. The Societal Implications of Deep Reinforcement Learning. *Journal of Artificial Intelligence Research* 70 (2021), 1003–1030.
[182] Bin Wu, Yuehong Wu, and Shangsong Liang. 2021. Data-Hungry Issue in Personalized Product Search. In *International Conference on Parallel and Distributed Computing: Applications and Technologies*. Springer, 485–494.
[183] Ga Wu, Mohamed Reda Bouadjenek, and Scott Sanner. 2019. One-Class Collaborative Filtering with the Queryable Variational Autoencoder. In *SIGIR'19*. 921–924.
[184] Junzhuang Wu, Yujing Zhang, Yuhua Li, Yixiong Zou, Ruixuan Li, and Zhenyu Zhang. 2023. SSTP: Social and Spatial-Temporal Aware Next Point-of-Interest Recommendation. *Data Sci. Eng.* 8, 4 (2023), 329–343.
[185] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2021. A Survey on Neural Recommendation: From Collaborative Filtering to Content and Context Enriched Recommendation. *arXiv preprint arXiv:2104.13030* (2021).
[186] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-encoders for Top-n recommender Systems. In *WSDM'16*. 153–162.

[187] Yuehong Wu, Bowen Lu, Lin Tian, and Shangsong Liang. 2022. Learning to Co-Embed Queries and Documents. *Electronics* 11, 22 (2022), 3694.
[188] Teng Xiao, Shangsong Liang, and Zaiqiao Meng. 2019. Hierarchical Neural Variational Model for Personalized Sequential Recommendation. In *WWW'19*. 3377–3383.
[189] Teng Xiao, Shangsong Liang, Hong Shen, and Zaiqiao Meng. 2018. Neural Variational Hybrid Collaborative Filtering. *arXiv preprint arXiv:1810.05376* (2018).
[190] Teng Xiao, Shangsong Liang, Weizhou Shen, and Zaiqiao Meng. 2019. Bayesian Deep Collaborative Matrix Factorization. In *AAAI'19*, Vol. 33. 5474–5481.
[191] Teng Xiao and Hong Shen. 2019. Neural Variational Matrix Factorization for Collaborative Filtering in Recommendation Systems. *Applied Intelligence* 49, 10 (2019), 3558–3569.
[192] Teng Xiao and Hong Shen. 2019. Neural Variational Matrix Factorization with Side Information for Collaborative Filtering. In *PAKDD'19*. Springer, 414–425.
[193] Teng Xiao, Hui Tian, and Hong Shen. 2019. Variational Deep Collaborative Matrix Factorization for Social Recommendation. In *PAKDD'19*. Springer, 426–437.
[194] Zhe Xie, Chengxuan Liu, Yichi Zhang, Hongtao Lu, Dong Wang, and Yue Ding. 2021. Adversarial and Contrastive Variational Autoencoder for Sequential Recommendation. In *arXiv preprint arXiv:2103.10693*. 532–536.
[195] Guangxia Xu and Xinting Hu. 2022. Multi-Dimensional Attention Based Spatial-Temporal Networks for Traffic Forecasting. *Wireless Communications and Mobile Computing* 2022 (2022).
[196] Guangxia Xu, Weifeng Li, and Jun Liu. 2020. A Social Emotion Classification Approach Using Multi-model Fusion. *Future Generation Computer Systems* 102 (2020), 347–356.
[197] Guangxia Xu, Xinkai Wu, Jun Liu, and Yanbing Liu. 2020. A Community Detection Method Based on Local Optimization in Social Networks. *IEEE Network* 34, 4 (2020), 42–48.
[198] Tianjun Yao, Qing Li, Shangsong Liang, and Yadong Zhu. 2020. BotSpot: A Hybrid Learning Framework to Uncover Bot Install Fraud in Mobile Advertising. In *CIKM'20*. 2901–2908.
[199] Qiaomin Yi, Ning Yang, and Philip Yu. 2023. Dual Adversarial Variational Embedding for Robust Recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35 (2023), 1421–1433.
[200] Jiachen Yu, Yuehong Wu, and Shangsong Liang. 2023. Wasserstein Topology Transfer for Joint Distilling Embeddings of Knowledge Graph Entities and Relations. In *2023 6th International Conference on Algorithms, Computing and Artificial Intelligence*. 176–182.
[201] Xianwen Yu, Xiaoning Zhang, Yang Cao, and Min Xia. 2019. VAEGAN: A Collaborative Filtering Framework Based on Adversarial Variational Autoencoders. In *IJCAI'19*. AAAI Press, 4206–4212.
[202] Meike Zehlike, Ke Yang, and Julia Stoyanovich. 2023. Fairness in Ranking, Part II: Learning-to-Rank and Recommender Systems. *ACM Comput. Surv.* 55, 6 (2023), 117:1–117:41.
[203] Guijuan Zhang, Yang Liu, and Xiaoning Jin. 2018. Adversarial Variational Autoencoder for Top-N Recommender Systems. In *ICSESS'18*. IEEE, 853–856.
[204] Guijuan Zhang, Yang Liu, and Xiaoning Jin. 2020. A Survey of Autoencoder-based Recommender Systems. *Frontiers of Computer Science* (2020), 1–21.
[205] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander Smola. 2022. ResNeSt: Split-Attention Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2736–2746.
[206] Jianfei Zhang, Jun Bai, Chenghua Lin, Yanmeng Wang, and Wenge Rong. 2022. Improving Variational Autoencoders with Density Gap-based Regularization. In *NeurIPS*. 35: 19470–19483.
[207] Ruiyi Zhang, Tong Yu, Yilin Shen, Hongxia Jin, Changyou Chen, and Lawrence Carin. 2020. Reward Constrained Interactive Recommendation with Natural Language Feedback. *arXiv preprint arXiv:2005.01618* (2020).
[208] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.
[209] Weina Zhang, Xingming Zhang, Haoxiang Wang, and Dongpei Chen. 2019. A Deep Variational Matrix Factorization Method for Recommendation on Large Scale Sparse Dataset. *Neurocomputing* 334 (2019), 206–218.
[210] Xiaofeng Zhang, Jingbin Zhong, and Kai Liu. 2021. Wasserstein Autoencoders for Collaborative Filtering. *Neural Computing and Applications* (2021), 2793–2802.
[211] Yizi Zhang, Hongxia Yang, and Meimei Liu. 2020. Variational Auto-encoder for Recommender Systems with Exploration-Exploitation. *arXiv preprint arXiv:2006.03573* (2020).
[212] Jing Zhao, Pengpeng Zhao, Lei Zhao, Yanchi Liu, Victor S. Sheng, and Xiaofang Zhou. 2021. Variational Self-attention Network for Sequential Recommendation. In *ICDE'21*. IEEE, 1559–1570.
[213] Xiangyu Zhao, Changsheng Gu, Haoshenglun Zhang, Xiwang Yang, Xiaobing Liu, Hui Liu, and Jiliang Tang. 2021. DEAR: Deep Reinforcement Learning for Online Advertising Impression in Recommender Systems. In *AAAI'21*, Vol. 35. 750–758.

[214] Kai Zheng, Xianjun Yang, Yilei Wang, Yingjie Wu, and Xianghan Zheng. 2020. Collaborative Filtering Recommendation Algorithm based on Variational Inference. *International Journal of Crowd Science* 4, 1 (2020), 31–44.
[215] Ting Zhong, Zijing Wen, Fan Zhou, Goce Trajcevski, and Kunpeng Zhang. 2020. Session-based Recommendation via Flow-based Deep Generative Networks and Bayesian Inference. *Neurocomputing* 391 (2020), 129–141.
[216] Yu Zhou, Haixia Zheng, Xin Huang, Shufeng Hao, Dengao Li, and Jumin Zhao. 2022. Graph Neural Networks: Taxonomy, Advances, and Trends. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 1 (2022), 1–54.
[217] Yaochen Zhu and Zhenzhong Chen. 2021. Collaborative Variational Bandwidth Auto-encoder for Recommender Systems. *arXiv preprint arXiv:2105.07597* (2021).
[218] Yaochen Zhu and Zhenzhong Chen. 2022. Mutually-Regularized Dual Collaborative Variational Auto-encoder for Recommendation Systems. In *WWW*. ACM, 2379–2387.

Received 30 June 2022; revised 9 April 2024; accepted 22 April 2024