Variational Bandwidth Auto-Encoder for Hybrid Recommender Systems

Yaochen Zhu and Zhenzhong Chen, Senior Member, IEEE

Abstract—Hybrid recommendations have recently attracted a lot of attention where user features are utilized as auxiliary information to address the sparsity problem caused by insufficient user-item interactions. However, extracted user features generally contain rich multimodal information, and most of them are irrelevant to the recommendation purpose. Therefore, excessive reliance on these features will make the model overfit on noise and difficult to generalize. In this article, we propose a variational bandwidth auto-encoder (VBAE) for recommendations, aiming to address the sparsity and noise problems simultaneously. VBAE first encodes user collaborative and feature information into Gaussian latent variables via deep neural networks to capture non-linear user similarities. Moreover, by considering the fusion of collaborative and feature variables as a virtual communication channel from an information-theoretic perspective, we introduce a user-dependent channel to dynamically control the information allowed to be accessed from the feature embeddings. A quantum-inspired uncertainty measurement of the hidden rating embeddings is proposed accordingly to infer the channel bandwidth by disentangling the uncertainty information in the ratings from the semantic information. Through this mechanism, VBAE incorporates adequate auxiliary information from user features if collaborative information is insufficient, while avoiding excessive reliance on noisy user features to improve its generalization ability to new users. Extensive experiments conducted on three real-world datasets demonstrate the effectiveness of the proposed method. Codes and datasets are released at https://github.com/yaochenzhu/VBAE.

Index Terms—Auto-encoders, information bottleneck, recommender systems, uncertainty modeling, variational inference

1 INTRODUCTION

In the era of information overload, people have been inundated by large amounts of online content. Therefore, it becomes increasingly difficult for users to discover interesting information. Consequently, recommender systems have played a pivotal role in modern online services due to their ability to help users discover items that they may be interested in from a large collection of candidates. Based on how recommendations are made, existing recommender systems can be categorized into three classes [1]: collaborative-filtering-based methods, content-based methods, and hybrid methods. Collaborative-filtering-based methods [2], [3] predict user preferences by exploiting their past activities, such as clicks or ratings, where the recommendation quality relies heavily on peers with similar behavior patterns. Content-based methods [4], [5], [6], on the other hand, make recommendations based on users or items that share similar features. Hybrid methods [7], [8], [9], [10] combine the advantages of both worlds where collaborative information and user/item features are comprehensively considered to generate more accurate recommendations.

Recent years have witnessed an upsurge of interest in employing auto-encoders [11] to both collaborative and content-based recommender systems, where compact representations of sparse user ratings [12], [13] or high-dimensional user/item features [14], [15] can be learned to more effectively exploit the similarity patterns between users or items for recommendations. As a Bayesian version of auto-encoder where the latent representations are modeled as random variables, variational auto-encoder (VAE) [16] demonstrates superiority compared to other forms of auto-encoders, such as the contractive auto-encoder [17] and the denoising auto-encoder [18]. Among them, collaborative variational auto-encoder (CVAE) [19], which extends the collaborative deep learning (CDL) framework [20], first used a VAE to infer latent item content embeddings from tf-idf textual features. The embeddings are then finetuned with rating information via matrix factorization. Multi-VAE [12], in contrast, used the VAE in the collaborative setting to learn compact user embeddings from discrete user ratings. Recently, Macrid-VAE [21] further extended Multi-VAE, where learned user representations were constrained to disentangle at both the macro and the micro-levels to improve the interpretability of learned embeddings.

Despite the success of VAEs in handling either user rating or feature side information, it is difficult to generalize VAEs to a hybrid recommender system, due to various challenges from both the collaborative and content-based components (Fig. 1). As users tend to vary in their activity levels and tastes, user embeddings learned through collaborative filtering bear different degrees of uncertainty, which hinders good recommendations for users with unreliable collaborative embeddings (e.g. user #4 and #5 in Fig. 1). The uncertainty mainly comes from three aspects: (1) Sparsity: First, for a user with sparser interactions (user #4), her associated embedding is more unreliable due to the information...
We present VBAE, a unified information-driven recommendation framework where the generation of user ratings and features is parameterized via deep Bayesian networks and their fusion is modeled as a personalized virtual communication channel, such that the rating sparsity and feature noise problems can be simultaneously addressed.

- A novel quantum-inspired uncertainty measurement of the hidden rating embedding is proposed accordingly to infer the bandwidth of the user-dependent channel, which enhances the model generalization ability by dynamically controlling the information allowed to be accessed from user features based on the sufficiency level of collaborative information.

- Two channel implementations with different desired properties, i.e., Bernoulli and Beta channels, are thoroughly discussed, with the corresponding objectives derived with posterior approximation and variational inference to make them amenable to stochastic gradient descent optimization.

- The proposed VBAE empirically out-performs state-of-the-art hybrid recommendation baselines. We also discover that the inferred bandwidth of the channel variable can well distinguish users with different sufficiency levels of collaborative information.
2 RELATED WORK

As a special kind of deep neural network, auto-encoders aim to learn compact representations of inputs by self-reconstruction [25]. Since both user ratings and features are high-dimensional sparse vectors where such that directly manipulating them in the original space is difficult, considerable effort has been dedicated to using auto-encoders to learn their low-dimensional compact representations [11]. Generally, based on whether the auto-encoder is used to tackle user or item side information, existing auto-encoder-based recommenders can be categorized into two main classes: user-oriented auto-encoders (UAEs) [12], [18], [26], [27] and item-oriented auto-encoders (IAEs) [11], [28].

2.1 Item-Oriented Auto-Encoders

The advent of IAE predates that of UAE, where item content auto-encoders are built on top of matrix factorization (MF)-based collaborative backbones, such as weighted matrix factorization [29], to incorporate auxiliary item content information into the factorized item collaborative embeddings. Two exemplar methods from this category are CDL [20] and CVAE [19], where an item offset variable is introduced to tightly couple the Bayesian stacked denoising auto-encoder (SDAE) [30] or variational auto-encoder (VAE) [16] with MF to enhance its performance. MF and item content auto-encoder are then trained in an iterative manner. Recently, auto-encoders have also been exploited to model the item collaborative information. Among them, DICER [28] was proposed to capture non-linear item similarity based on their user ratings, and from which disentangles the content information for better generalization. Since in collaborative IAEs, the input dimension and the number of trainable weights is proportional to the number of users, and the number of training samples equals the number of items, these methods generally require a large item-to-user ratio such that a good representation of items can be learned for a satisfactory recommendation performance.

2.2 User-Oriented Auto-Encoders

Compared with IAEs, UAEs have attracted more attention among researchers because they break the long-standing bottleneck of the linear collaborative modeling ability of MF and allow modeling users in a deeper manner [18], [26]. Instead of factorizing the rating matrix into user and item embeddings, UAE-based recommenders take the historical ratings of users as the inputs, embed them into hidden user representations with a deep encoder network, and from which reconstruct the ratings with a deep decoder network. The reconstructed ratings for unrated items are then ranked for recommendations. Since UAE-based recommenders eliminate the need of modeling item latent variables and reconstruct the whole ratings directly from the user latent embedding, another advantage of UAEs over MF is that they are efficient to fold in new users for whom historical ratings have been recorded, since recommendations can be made with a single forward propagation. The first UAE-based recommender system is the collaborative denoise auto-encoder (CDAE) [18], where the input ratings are randomly masked with zeros to simulate the missing ratings. Afterwards, Multi-VAE [12] was proposed where a VAE with multinomial likelihood is used instead of the DAE in CDAE, which demonstrates clear performance improvement. However, one key problem for these collaborative UAES is that if the ratings of certain users are sparse, the recommendation performance could be severely degenerated due to the lack of collaborative information.

2.3 Hybrid Recommendation Techniques

Due to the wide availability of user features and various methods to build user profiles with the feature of items the user has interacted with, incorporating auxiliary user features into UAE to address its sparsity problem has become a new trend. A simple method is the early fusion strategy [31], where user features are concatenated with ratings as inputs for UAE. With this strategy, the first dense layer of UAE can be viewed as calculating a weighted combination of user rating and features. A more sophisticated approach is the conditional VAE (CondVAE) [32], where the user features are exploited to calculate the conditional prior of user latent variables. However, both methods treat the relative importance of collaborative and content information as fixed for all users, which ignores the individual differences in the reliability of extracted user features and the sufficiency levels of historical rating information [23]. This is problematic, since user features contain much irrelevant information and noise, and a good recommender system should avoid unnecessary dependence on these features when collaborative information is sufficient to improve generalization. Recently, attention mechanism has been introduced into recommender systems to dynamically fuse the rating and feature information [9], [33]. However, the attention weights are generally calculated by modality embedding and softmax normalization, which may fail to capture the true relative importance among the modalities. Therefore, it motivates us to design a quantum-inspired uncertainty measurement of collaborative information and a channel with user-specific bandwidth to dynamically fuse the user feature and collaborative information.

3 METHODOLOGY

3.1 Problem Formulation

The focus of this article is on recommendations with implicit feedback [34]. We define the rating matrix as \( R \in \mathbb{R}^{I \times J} \), where each row vector \( r_i \) is the bag-of-words vector denoting whether user \( i \) visited each of the \( J \) items. \( R \) is obtained by keeping track of user activities for a certain amount of time. In addition, the user profiles are represented by matrix \( X \in \mathbb{R}^{I \times F} \), where each row vector \( x_i \) is the extracted feature for the \( i \)th user. \( x_i \) may include inherent user attributes such as her age, location, and self-description, etc., or may be built from the feature of items that users have interacted with when such information is not available. The capital non-boldface letters \( R \) and \( X \) are used to denote the corresponding random variables, respectively.\(^1\) The density of \( r_i \), which is defined as \( \sum_j r_{ij} \), could vary dramatically for different \( i \), and \( x_i \) is generally high-dimensional and noisy. Given the partial observation of the records in \( R \) and the user features \( X \), the focus of this article is on making predictions of the

---

1. The subscript \( i \) will be omitted for simplicity if no ambiguity exists.
remaining ratings in $\mathbf{R}$ so as to recommend new relevant items to users.

### 3.2 Model Overview

The PGM in the left part of Fig. 2 shows the overall generative and inference process of VBAE. The details of VBAE are discussed in the following sections.

### 3.3 Generative Process

#### 3.3.1 Fusion by Channel With User-Dependent Bandwidth

In this article, user collaborative embedding $\mathbf{z}_t$ and user feature embedding $\mathbf{z}_u$ are assumed to lie in $K$-dimensional Gaussian latent spaces. Traditional generative models for hybrid recommendation [19], [20], [35] directly add $\mathbf{z}_t$ and $\mathbf{z}_u$ via an offset variable to form the latent user embedding $\mathbf{v}$, or learn their fixed relative weights by concatenation, and do not consider that the uncertainty of $\mathbf{z}_t$ varies for different users due to both explicit (difference in the sparsity and diversity of ratings) and implicit (difference in the number of overlapped ratings) factors, nor that $\mathbf{z}_t$ contains lots of irrelevant information that may distract the recommendation model. This could be problematic, since the collaborative embeddings $\mathbf{z}_t$ for users who have denser ratings or who rate items that are also rated by many other users are more reliable, and $\mathbf{v}$ can afford to access less information from $\mathbf{z}_t$, such that unnecessary dependence of the model on noisy user features can be avoided.

From an information-theoretic perspective, if we view the fusion of $\mathbf{z}_t$ and $\mathbf{z}_u$ into $\mathbf{v}$ as a virtual communication channel, the issue sources from the assumption that the channel is deterministic and independent of the information already contained in the collaborative embedding $\mathbf{z}_t$. This ignores the individual difference in the sufficiency level of collaborative information. Therefore, to address such a problem, we design a user-dependent channel in VBAE by introducing a latent capacity variable $\alpha$ that determines the bandwidth of the channel from $\mathbf{z}_t$ to $\mathbf{v}$. The bandwidth variable $\alpha$ encodes for each user our belief towards how much extra information is required from the user features given the information already contained in the observed interactions. Through this mechanism, the channel dynamically allocates the amount of information that is allowed to flow from $\mathbf{z}_t$ to $\mathbf{v}$ conditional on $z_b$. Two strategies are explored to implement the user-dependent channel with bandwidth $\alpha$. The first strategy we consider is called the “hard” channel, where the bandwidth $\alpha$ is achieved when $\mathbf{v}$ losslessly accesses $\mathbf{z}_t$ with probability $\alpha$, and accesses no information otherwise [36]. In the generative case, we can introduce an auxiliary channel variable $d$, and draw $d$ from a Bernoulli distribution with probability $\alpha$:

$$d \sim \text{Bernoulli}(\alpha).$$

Although the hard channel conforms more strictly to the definition of bandwidth in information theory, it may result in training instability because the bandwidth $\alpha$ only appears as a statistical property, i.e., an expectation when the feature and collaborative embeddings of a user are repeatedly fused for multiple times. However, for one user, the user latent embedding either access the user feature information or not in one generation step, which is coarse in granularity to distinguish user with different uncertainty levels of collaborative information. Therefore, we consider a second strategy, the “soft” channel, which is a relaxed version of the hard channel and resembles more to the variational attention channel and resembles more to the variational attention approach proposed in [37] than the variational information bandwidth theory [36]. This strategy assumes that the channel variable $d$ is drawn from a Beta distribution and the bandwidth $\alpha$ determines the mean of the Beta:

$$d \sim \text{Beta}(\alpha_1, \alpha_2), \quad \text{where } \alpha = \alpha_1/(\alpha_1 + \alpha_2),$$

and the channel $d$ curtails or amplifies the weights of user feature information based on the bandwidth $\alpha$. Given only $\alpha$, however, the Beta distribution for the channel $d$ is undetermined, as its variance remains to be specified to calculate both $\alpha_1$ and $\alpha_2$. However, since we care primarily about the bandwidth itself and not its uncertainty, we fix the variance of the Beta, which is treated as a nuisance parameter, to a small value $\delta^2_{\text{fixed}}$ for simplicity. Hereafter, we use the mean-variance parameterization of Beta distribution unless otherwise specified, since it explicitly contains the bandwidth as its first parameter. We use VBAE-hard and VBAE-soft to distinguish the two channel implementation strategies.
The detailed comparisons between the soft and hard channels are summarized in Fig. 3 for reference. After drawing $d$, the user latent variable $v$ is deterministically calculated as

$$v = z_o + d \cdot z_i,$$

which defines the fusion process of $z_i$ into $z_o$ via the user-dependent channel of VBAE-hard and VBAE-soft.

### 3.3.2 Neural Network Implementations

To model the non-linear generation process of user features and ratings from the corresponding latent variables, we parameterize the generative distributions as deep neural networks. The user feature $x$ is generated from the user-feature latent variable $z_i$ via a multilayer perceptron (MLP) $Gen(u(z_i))$: If $x$ is binary, we squash the output of $Gen$ by the sigmoid function, and draw $x$ from $Bernoulli(\text{sigmoid}(Gen(u(z_i))))$, or if $x$ is real-value, we take the raw outputs of $Gen$ as the mean of a Gaussian distribution and draw $x$ from $\mathcal{N}(Gen(u(z_i)), \lambda_x^{-1}I_s)$. Finally, we put a multinomial likelihood on the ratings $r$ as [12], and generate $r$ from the latent user variable $v$ via $p(r|v)$ parameterized as $Multi(N, \text{softmax}(Gen(u(v))))$, where $Gen$ is another MLP-based generative neural network, and $N$ is number of interacted items. The generation process of $x, r$ from $z_o, v$ is given as follows:

1. For each layer $l$ of the collaborative and the user-feature modules of the generation network:
   - (a) For each column $n$ of the weight matrices, draw $W^{(l)}_{[u,t]} \sim \mathcal{N}(0, \lambda_w^{-1}I_k)$;
   - (b) Draw the bias vector from $b^{(l)}_{[u,t]} \sim \mathcal{N}(0, \lambda_w^{-1}I_k)$;
   - (c) For $h^{(l)}_{[u,t]}$ of a user $u$, draw $h^{(l)}_{[u,t]} \sim \delta(t_{act}(W^{(l)}_{[u,t]} h^{(l-1)}_{[u,t]} + b^{(l)}_{[u,t]}))$.

2. For user features that are binary, draw $x \sim Bernoulli(\text{sigmoid}(W^{(L+1)} h^{(L)} + b^{(L+1)}))$.

For user-item interactions, draw $r \sim Multi(N, \text{softmax}(W^{(L+1)} h^{(L)} + b^{(L+1)}))$, where $h^{(0)}_{[u,t]} = z_i, h^{(0)} = v, \lambda_w$ is a hyperparameter, $t_{act}$ is the intermediate activation function and $\delta$ is the Dirac Delta function. Step 1.c can be alternatively viewed as putting Gaussian priors on the intermediate activations and setting the precision to infinity.

The generative model of VBAE is described by the joint distribution of all observed and hidden variables:

$$\begin{align*}
p_v(R, X, Z_o, Z_i, D) &= \pi_v(R|Z_o, Z_i, D)\pi_v(X|Z_i) \\
p(Z_o)p(Z_i)p(D),
\end{align*}$$

where the symbol $\theta$ denotes the set of trainable parameters that pertain to the generation network.

### 3.4 Inferential Process

Given Eq. (1), however, it is intractable to calculate the posterior $p_v(Z_o, Z_i, D|R, X)$ exactly, as the non-linearity of generative process precludes us from integrating over the latent space and calculating the marginal distribution of the observed evidence $p_v(R, X)$. Therefore, we resort to the amortized variational inference [38], where we introduce a variational posterior $q_v(Z_o, Z_i, D|R, X)$ parameterized by an inference neural network as an approximation to the true but intractable posterior. Using the conditional independence assumptions implied by VBAE, the joint variational posterior can be decomposed into the compact product of two factors that make up two modules of the inference network: the collaborative module $q_v(Z_o, D|R) = q_v(Z_o|Z_i) \times q_v(D|R)$ and the user feature module $q_v(Z_i|X)$, which infer the user collaborative embeddings, the bandwidth of the user-dependent channel, and user feature embeddings from the user ratings and features, respectively.

#### 3.4.1 Quantum-Inspired Semantic and Uncertainty Measurement of Collaborative Information

The collaborative module infers the user latent collaborative variable and the latent channel variable from the observed user-item interactions. Since the bandwidth of the channel determines the sufficiency level of the collaborative information, an important role that this module should play is to disentangle uncertainty and semantic information from the user ratings. To achieve this objective, we first use an MLP to embed the raw, sparse rating vector into a compact hidden representation $h^{inf}_k$ (the superscript $\text{inf}$ will be omitted for simplicity hereafter) for each user. Inspired by [39], we then use the length (L2-norm) of the hidden representation as the uncertainty measurement of collaborative information to calculate the channel bandwidth and use the direction (L2-normalized hidden embedding) as the representation of semantic information to infer the latent collaborative embedding. The details of the introduced semantic and uncertainty measurement of the hidden rating embeddings are illustrated in Fig. 4, which draws inspiration from theoretic quantum mechanics. To see the connections, we first liken the hidden rating embedding $h_k$ to a quantum superposition state of a physical system, which is represented by a complex vector in $\mathbb{C}^N$. Then, the superposition vector has the property that the norm of the vector is positively correlated with the probability that this superposition is observed when measuring the system (which depicts the fundamental uncertainty of quantum physics), and the direction of the vector distinguishes this superposition state with other states (which carries the state semantic information).
Such a semantic-uncertainty interpretation of a vector’s length and direction also works for hidden rating embedding $\mathbf{h}_b$ in recommendation, where the length and direction can be associated with similar meanings. To gain the intuition, we demonstrate the case where the network used to calculate $\mathbf{h}_b$ has one dense layer (Generalization to the multi-layer case requires linearizing the network locally with network Jacobians). We first decompose the calculation of $\mathbf{h}_b$ from raw rating vector $\mathbf{r}$ into two basic operations, embedding and element-wise sum, as follows:

$$\mathbf{h}_b = \mathbf{W}^{(1)}_b \cdot \mathbf{r} = \sum_{j \neq j_b} \mathbf{w}^{(1)}_{b,j}.$$  

(2)

If we assume that each element in $\mathbf{W}^{(1)}_b$, i.e., $w^{(1)}_{b,j}$ is independent and identically distributed (i.i.d.) Gaussian variable with zero mean and a small variance $\epsilon^2$, the $j$th element in $\mathbf{h}_b$, which is denoted as $h_{b,j}$ is the sum of $N_{\text{int}}$ independent Gaussian variables, where $N_{\text{int}} = \sum_{j \neq j_b} 1$ is the number of items this user has interacted with, i.e., the density of ratings. Therefore, $h_{b,j}$ is also a Gaussian variable that follows $N(0, N_{\text{int}} \epsilon^2)$. According to the probability theory, since the squared L2-norm of $\mathbf{h}_b$, i.e., $||\mathbf{h}_b||^2$, is the sum of squares of $K_h$ Gaussian variables with zero mean and $N_{\text{int}} \epsilon^2$ variance, $||\mathbf{h}_b||^2$ follows the scaled Chi-square distribution $N_{\text{int}} \epsilon^4 \cdot \chi_{K_h}^2$ and is equivalent to Gamma distribution with cumulative distribution function $\Gamma(K_h/2, 2N_{\text{int}} \epsilon^2)$ [40]. Finally, according to the property of Gamma distribution, the expected value of $||\mathbf{h}_b||^2$ can be calculated as $N_{\text{int}} K_h \epsilon^2$. This term is a monotonic increasing function of the number of interacted item $N_{\text{int}}$, i.e., the rating density, which is a main indicator for the sufficiency level of collaborative information.

### 3.4.2 Bandwidth and User Collaborative Embedding

The above property reveals that $||\mathbf{h}_b||^2$ is positively correlated with the sufficiency level of the collaborative information. Moreover, compared to the L2-norm of the sparse rating vector $\mathbf{r}$, the hidden embedding $\mathbf{h}_b$ lies in a latent low dimensional space where user collaborative representations are more compact, which could better reflect the similarity among user rating patterns by eliminating redundant information contained in similar items. Therefore, the L2-norm of $\mathbf{h}_b$ contains important information regarding interaction sparsity and rating similarity, which are suitable for the inference of the bandwidth that we have defined in the previous section. In addition, the L2-normalization of $\mathbf{h}_b$ eliminates the negative influence of differences in the number of items the users have interacted with by scaling the hidden rating embeddings of users with different activity levels into the same sphere space, which makes the direction of $\mathbf{h}_b$ a more suitable representation than the original $\mathbf{h}_b$ to infer user collaborative embeddings. This is in contrast with the previous auto-encoder-based approaches such as Multi-VAE [12], CondVAE [32], where the information regarding the sparsity level of the historical ratings is discarded after the L2-normalization of the input ratings.

In practice, we calculate the bandwidth by a linear transformation of $||\mathbf{h}_b||$ transformed with sigmoid activation function to squash the value between $[0, 1]$:

$$\alpha = \text{sigmoid}(w_B^c ||\mathbf{h}_b|| + b_B^c).$$  

(3)

Since $||\mathbf{h}_b||$ is strictly non-negative, in backward propagation, the weight $w_B^c$ can only be updated in one direction for all samples in one mini-batch, if the sign of the gradients is the same for the users. This is problematic, since the training loop could make $\alpha$ converge to zero or one where the bandwidth for all users is identical, which fails to distinguish users with different sufficiency levels of collaborative information. This is referred to as the mode collapsing problem in variational inference. In this article, this problem is addressed through the batch normalization [41]. We denote a mini-batch of calculated length of user hidden embedding as $B = \{||\mathbf{h}_{b,0}||, ||\mathbf{h}_{b,\text{mini-1}}||, \ldots, ||\mathbf{h}_{b,\text{mini}}||\}$. Before feeding samples in the mini-batch $B$ into Eq. (3) to calculate the bandwidth, we renormalize them by:

$$\bar{||\mathbf{h}_{b,i}||} = \frac{||\mathbf{h}_{b,i}|| - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}},$$  

(4)

where $\mu_B$ and $\sigma_B^2$ are the mean and sample variance of $||\mathbf{h}_b||$ in the mini-batch, and $\epsilon$ is a small value to avoid division by zero error. By batch normalization, we observe that the inferred bandwidth $\alpha$ corresponds more consistently with the sufficiency level of the user collaborative information, and the training procedure becomes more stable as well. In the testing phase, $\mu_B$ and $\sigma_B^2$ are fixed to their running average estimated in the training phase.

### 3.4.3 Neural Network Implementations

The user feature module infers the feature-based latent user embedding from the extracted user features by another MLP, which serves as the auxiliary information source to the collaborative information. The detailed inference process of the latent variables $Z_b, Z_i$, and $D$ through the inference network is described as follows:

1. For each layer $l$ of the collaborative and user-feature module of the inference network:

   a. For each column $n$ of the weight matrices, draw

   $$W^{(l)}_{\{b,l\},n} \sim \mathcal{N}(0, \lambda^{-1}_w I_{K_l});$$

   b. Draw the bias vector from $b^{(l)}_{\{b,l\}} \sim \mathcal{N}(0, \lambda^{-1}_w I_{K_l});$

   c. Draw the user feature embedding, $\mathbf{z}_i \sim \mathcal{N}(0, \lambda^{-1}_z I_{K_i});$

   d. Draw the auxiliary information, $\mathbf{z}_a \sim \mathcal{N}(0, \lambda^{-1}_a I_{K_a});$

   e. Draw the user collaborative embedding, $\mathbf{z}_c \sim \mathcal{N}(0, \lambda^{-1}_c I_{K_c});$

   f. Draw the label, $\mathbf{z}_d \sim \mathcal{N}(0, \lambda^{-1}_d I_{K_d});$

2. For each mini-batch $B = \{||\mathbf{h}_{b,0}||, ||\mathbf{h}_{b,\text{mini-1}}||, \ldots, ||\mathbf{h}_{b,\text{mini}}||\}$, calculate the bandwidth

   $$\alpha = \text{sigmoid}(w_B b ||\mathbf{h}_b|| + b_B^c).$$

   (3)

Since $||\mathbf{h}_b||$ is strictly non-negative, in backward propagation, the weight $w_B^c$ can only be updated in one direction for all samples in one mini-batch, if the sign of the gradients is the same for the users. This is problematic, since the training loop could make $\alpha$ converge to zero or one where the bandwidth for all users is identical, which fails to distinguish users with different sufficiency levels of collaborative information. This is referred to as the mode collapsing problem in variational inference. In this article, this problem is addressed through the batch normalization [41]. We denote a mini-batch of calculated length of user hidden embedding as $B = \{||\mathbf{h}_{b,0}||, ||\mathbf{h}_{b,\text{mini-1}}||, \ldots, ||\mathbf{h}_{b,\text{mini}}||\}$. Before feeding samples in the mini-batch $B$ into Eq. (3) to calculate the bandwidth, we renormalize them by:

$$\bar{||\mathbf{h}_{b,i}||} = \frac{||\mathbf{h}_{b,i}|| - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}},$$  

(4)

where $\mu_B$ and $\sigma_B^2$ are the mean and sample variance of $||\mathbf{h}_b||$ in the mini-batch, and $\epsilon$ is a small value to avoid division by zero error. By batch normalization, we observe that the inferred bandwidth $\alpha$ corresponds more consistently with the sufficiency level of the user collaborative information, and the training procedure becomes more stable as well. In the testing phase, $\mu_B$ and $\sigma_B^2$ are fixed to their running average estimated in the training phase.
(c) For $h_{i,t}^l$ of a user $i$, draw
\[ h_{i,t}^l \sim \delta\left( f_{\text{act}}\left( W_{i,t}^l h_{i,t}^{(l-1)} + b_{i,t}^l \right) \right). \]

(2) For the user-dependent channel variable:
(a) Calculate the bandwidth from its logits, which is inferred by the L2-norm of the hidden rating embedding $h_b^L$ as follows:
\[ \alpha = \text{sigmoid}\left( w_b^l \| h_b^L \| + b_b^l \right); \]
(b) For VBAE-hard, draw the Bernoulli channel variable:
\[ d \sim \text{Bernoulli}(\alpha); \]
(c) For VBAE-soft, draw the Beta channel variable:
\[ d \sim \text{Beta}(\alpha, \sigma_{\text{fixed}}^2). \]

(3) For the collaborative and user feature latent variable:
(a) Draw the mean and standard deviation:
\[ [\mu_{(b,t)}, \log \sigma_{(b,t)}] \sim \delta\left( W_{b,t}^l h_b^L + b_{b,t}^l \right); \]
(b) Draw the sample of the latent variable:
\[ z_{(b,t)} \sim \mathcal{N}\left( \mu_{(b,t)}, \text{diag}(\sigma_{(b,t)}^2) \right). \]

It is not trivial to draw samples from the Bernoulli or the Beta channel variable such that gradients can be back-propagated to the trainable weights of the inference network, and we defer the discussion of the solution, which is called posterior approximation and reparameterization trick, to Section 3.7. The neural network implementation of VBAE is schematically illustrated in Fig. 2 for reference.

### 3.5 Training Objective

To jointly learn the parameters of the generative network and the inference network, we maximize the evidence lower bound (ELBO), which is an approximation of the marginal log-likelihood of evidence $p_R(R, X)$:
\[ \mathcal{L}(q) = \mathbb{E}_{q_\theta}[\log p_R(R, X, Z_b, Z_i, D)] - \log q_\theta(Z_b, Z_i, D|R, X)], \]
\[ = \mathbb{E}_{q_\theta}[\log p_R(R|Z_b, Z_i, D) + \log p_\phi(X|Z_i)] \]
\[ - \mathbb{KL}[q_\theta(Z_b, D|R)||p(Z_b, D)] - \mathbb{KL}[q_\theta(Z_i|X)||p(Z_i)], \]
\[ \text{(5)} \]
and the value of $\mathcal{L}(q)$, for a fixed $\theta$, achieves the maximum if and only if the discrepancy between the variational approximation $q_\theta$ and the true posterior $p_\phi$ measured by the Kullback-Leibler (KL) divergence is zero (i.e., if $q_\theta = p_\phi$).

### 3.6 Maximum a Posteriori Estimation

Although the collaborative and user feature module of VBAE can be jointly trained as Eq. (5), extra computational and memory consumption is inevitable. In addition, in joint training, the VBAE model may converge to a sub-optima where it relies solely on one main information source for the recommendation. This is undesirable, because the user ratings and features contain complementary information, both of which are important for recommendations. Therefore, we take an EM-like optimization approach, where we iteratively consider only one of the variational distributions in $q_\theta(Z_b, D|R)$ and $q_\theta(Z_i|X)$ and fix the random variables (e.g., to their means or their previous estimates) that concern the other. Consequently, for the collaborative part $q_\theta(Z_b, D|R)$, we fix $Z_i$ to the estimated mean to calculate $\nu$, and the objective becomes:
\[ L_{h,\text{step}}^{\text{MAP}} = \mathbb{E}_{q_\theta(Z_b, D|R)}[\log p_R(R)] - \mathbb{KL}[q_\theta(D|R)||p(D)] \]
\[ - \mathbb{KL}[q_\theta(Z_b|R)||p(Z_b)] - \frac{\lambda_z}{2} \sum \left( \| W_{b,t}^0 \|_F^2 + \| b_{b,t}^0 \|_2^2 \right), \]
\[ \text{(6)} \]
where the prior of $D$ is ${\text{Bernoulli}}(0.5)$ for VBAE-hard and the standard logistic-normal distribution for VBAE-soft. Eq (6) can be viewed alternatively as paying a cost equals to the KL-with-prior term whenever the model accesses the noisy user feature information, and the cost is dynamically decided by the urgency level of introducing the extra user feature information based on the sufficiency level of the collaborative information. This can prevent the model from depending excessively upon the noisy features. After one-step optimization of $L_{h,\text{step}}^{\text{MAP}}$, we then fix $Z_b$ and $D$ to their estimated values to calculate the user latent variable $v$, and maximize the following objective for the user feature part:
\[ L_{i,\text{step}}^{\text{MAP}} = \mathbb{E}_{q_\theta(Z_i|X)}[\log p_R(V)] + \mathbb{E}_{q_\theta(Z_i|X)}[\log p(X|Z_i)] \]
\[ - \mathbb{KL}[q_\theta(Z_i|X)||p(Z_i)] - \frac{\lambda_z}{2} \sum \left( \| W_{i,t}^0 \|_F^2 + \| b_{i,t}^0 \|_2^2 \right), \]
\[ \text{(7)} \]
Intuitively, the objectives consist of two parts. The first part is the expected log-likelihood term, where the hidden Gaussian embeddings and the latent channel variable are encouraged to best explain the extracted user features and the observed historical ratings. The second part is the KL-with-prior terms and the weight decay terms, which act as regularizers to prevent over-fitting. Liang et al. [12] have shown that the KL regularization for $Z_b$ could be too strong, which over-constrains the representation ability of latent collaborative embeddings. As a solution, they introduced a scalar $\beta$ to control the weight of the KL term for the latent collaborative variable in Eq. (6), which has its theoretical foundation in both beta-VAE [42] and variational information bottleneck theory [43], [44]. We anneal the $\beta$ from 0 to 0.2 as in [12] and have confirmed the effectiveness of KL annealing in our experiments. Under such settings, the model learns to encode more information of its user in the initial training stages while gradually regularizing $z_b$ by forcing it close to the prior as the training proceeds [45].

### 3.7 Monte Carlo Gradient Estimator

In this section, we derive the gradients of $L_{h,\text{step}}^{\text{MAP}}$ and $L_{i,\text{step}}^{\text{MAP}}$ w.r.t. the trainable parameters of the generation and inference networks to make them amenable to SGD optimization. For Gaussian and Bernoulli distributions, since their KL divergence with prior can be computed analytically, the minimization of the KL terms in Eqs. (6), (7) w.r.t. the weights of the inference network $\phi$ can be properly calculated. However, since the gradients of the expected log-likelihood terms, which we note as $\text{ELM}_{(b,l)}$, need to be back-propagated through stochastic nodes $Z_b$, $Z_i$, and $D$, it precludes us from calculating an analytic solution. Hence, we introduce Monte Carlo methods to form unbiased estimators of the gradients.
For \( \theta \), as the generative distribution is explicit in an expectation form, its gradient can be estimated by generating samples from the encoder distribution, calculating the gradients, and taking the average [46]. For \( \phi \) that is associated with the inference of user feature and collaborative embeddings, we use the reparameterization trick, where we transform the stochastic nodes into differentiable bivariate functions of their parameters and random noises to allow gradients to pass through the distribution parameters.

### 3.7.1 User Embeddings: Vanilla Reparameterization

Specifically, for the latent Gaussian variables \( Z_{(b,t)} \), with the vanilla reparameterization trick introduced in [16], their samples can be represented as:

\[
z^{(l)}_{(b,t)}(\mu_{(b,t)}, \sigma_{(b,t)}; \epsilon^{(l)}) = \mu_{(b,t)} + \epsilon^{(l)} \odot \sigma_{(b,t)}. \tag{8}
\]

where \( \epsilon^{(l)} \sim \mathcal{N}(0, I) \). Eq (8) could be viewed alternatively as injecting Gaussian noise to the hidden user collaborative and user feature variables, which is the main mechanism that previous auto-encoder-based recommender systems adopt to address the rating and feature noise problems.

### 3.7.2 Hard Channel: Gumbel-Softmax Reparameterization

For the channel in VBAE-hard that follows the Bernoulli distribution, we note that sampling from which is equivalent to sampling a one-hot vector from a two-class Categorical distribution with probability mass \( \alpha = [\alpha_1, 1 - \alpha] \) and discarding the second dimension. Therefore, we resort to the Gumbel-softmax trick [47] and reparameterize samples from the hard channel \( D \) as:

\[
d^{(l)}(\alpha, g^{(l)}) = \frac{\exp\left(\log(\alpha_i) + g^{(l)}_i/\tau\right)}{\sum_{i=1}^{2} \exp\left(\log(\alpha_i) + g^{(l)}_i/\tau\right)} = \text{sigmoid}\left(\log\left(\frac{\alpha_i}{1 - \alpha_i}\right) + g^{(l)}_i - g^{(l)}_2, \tau\right), \tag{9}
\]

where \( g^{(l)}_1 - g^{(l)}_2 \sim \text{Gumbell}(0, 1) \) and \( \tau \) is the temperature of the softmax and the sigmoid. When \( \tau \) approaches zero, the samples \( d^{(l)} \) are proved to be equivalent to samples drawn from the corresponding Bernoulli distribution with the probability \( \alpha \). In practice, \( \tau \) is generally annealed as the training proceeds for a more stable convergence.

### 3.7.3 Soft Channel: Logistic-Normal Reparameterization

Similarly, we draw the Beta channel of VBAE-soft by keeping the first dimension of a sample from the corresponding two-class Dirichlet distribution. However, unlike Gaussian and Categorical distributions, there is no consensus regarding how to reparameterize a Dirichlet variable [37], [48], [49]. In this article, we eschew the commonly used reparameterization strategies that transform a uniformly distributed vector by the inverse of the Dirichlet cumulative distribution function as the bivariate transformation. In contrast, we derive its reparameterization with logistic-normal approximation instead [49]. The reason is that, the logistic-normal distribution converts the original parameters \( [\alpha_1, \alpha_2] \) of the Dirichlet (the values that should be predicted by the inference network) to the mean and standard deviation of a Gaussian distribution, such that the convergence to a low-entropy area is smoother. Otherwise, to reach a low variance area of the Dirichlet requires large values of \( [\alpha_1, \alpha_2] \), which is difficult to learn by the network and results in unstable training dynamics [37]. The relationship between the parameters of logistic-normal and the corresponding Dirichlet is formulated as follows:

\[
\begin{align*}
\mu_1 &= -\mu_2 = \frac{1}{2} \left( \log \alpha_1 - \log \alpha_2 \right) \\
\sigma_1 &= \sigma_2 = \frac{1}{4} \left( \frac{1}{\alpha_1} + \frac{1}{\alpha_2} \right) = \delta_{\text{fixed}}. \tag{10}
\end{align*}
\]

We fix the logistic-normal to a small value in Eq. (10), as here only the mean of the Beta channel variable (i.e., the bandwidth) is important, and a small value of the variance prevents the Dirichlet distribution from stuck into a low-entropy area. This is also a common trick used in most regression task, where the outputs are assumed to be the mean Gaussian and the variance is trivial to model. The sample \( d^{(l)} \) from the logistic-normal is drawn according to

\[
d^{(l)}(\alpha, \epsilon^{(l)}) = \frac{\exp\left(\left(\mu_i + \sigma_i \cdot \epsilon^{(l)}_i\right)\right)}{\sum_{i=1}^{2} \exp\left(\left(\mu_i + \sigma_i \cdot \epsilon^{(l)}_i\right)\right)} = \text{sigmoid}\left(2 \cdot \mu_i + \sigma_i \cdot \left(\epsilon^{(l)}_i - \epsilon^{(l)}_2\right)\right), \tag{11}
\]

where \( \epsilon^{(l)}_i \sim \mathcal{N}(0, 1) \). A close look at Eq. (11) shows that it bears great similarity to Eq. (9), since we can view \( 2 \cdot \mu_i = \log(\frac{\alpha_i}{1 - \alpha_i}) = \log(\frac{x^{(l)}_i}{1 - x^{(l)}_i}) \) as pseudo logits of the bandwidth, and both Eqs. add and subtract two i.i.d. random variables to the logits of bandwidth before squashing it into \( (0, 1) \) with the sigmoid function. The major difference between these two equations is that for Eq. (9), a small temperature of the sigmoid pushes the value of \( d \) to 0 or 1, i.e., the two extremities, such that for each user the channel is either open or closed in one iteration. However, in Eq. (9), the temperature is set to one and therefore \( d \) can take any value between \([0, 1]\), which avoids sudden swerve of gradient direction in training and smooths the convergence.

### 3.7.4 Unbiased Gradient Estimators

With the stochastic user latent embedding variable and the channel variable reparameterized with the strategies we introduce above, the unbiased gradient estimator of the objective ELM\(_{(b,t)}\) step w.r.t. \( \phi \) can be formulated as:

\[
\nabla_\phi E_{\text{U-step}} \simeq \frac{1}{L} \sum_i \left( \nabla_{\phi Z_i^{(l)}} \log p(x | \phi z^{(l)}_i) \cdot \nabla_{\phi} z^{(l)}_i, \ n^{(l)}_i \right);
\]

\[
\nabla_\phi E_{\text{B-step}} \simeq \frac{1}{L} \sum_i \left( \nabla_{\phi Z_i^{(l)}} \log p(r | \phi z^{(l)}_i) \cdot \nabla_{\phi} z^{(l)}_i \right)
\]

\[
+ \frac{1}{L} \sum_i \left( \nabla_{\phi} V_i \log p(x | \phi z^{(l)}_i) \cdot \nabla_{\phi} z^{(l)}_i \right), \tag{12}
\]

where \( \simeq \) is used to denote that the RHS, is an unbiased estimator of the LHS. Since the variance of the gradient estimated by the reparameterization trick is low, previous work has
Algorithm 1. SGD for VBAE-soft and VBAE-hard.

Input: \( D = \{(r_i, x_i)\} \), a dataset of collected user ratings and features, where \( r_i \in \{0, 1\}^I \), \( i \in \{1, 2, \ldots, I\} \); \( \Theta_t, \Theta_b \), the randomly initialized weights of the collaborative and feature sub-networks; \( lr \), the learning rate.

while metrics on validation users improve, do

\[ L_{MAP}^{t,\text{step}} = 0; \]

for all \((r_i, x_i) \in D\), do

Infer \( \mu_t \) and \( \sigma_t \) and sample \( z_t \sim \mathcal{N}(\mu_t, \sigma_t) \);

Calculate \( x_t \) the reconstruction of \( x_i \) from \( z_t \);

Calculate \( L_{MAP}^{t,\text{step}} \) from \( x_t \) and \( x_i \), as Eq. (7).

\[ L_{MAP}^{t,\text{step}} = L_{MAP}^{t,\text{step}} + \]

end

Update \( \Theta_t = \Theta_t - lr \cdot \nabla_{\Theta_t}(-L_{MAP}^{t,\text{step}})/I \)

forall \((r_i, x_i) \in D\), do

Infer \( \mu_o \) and \( \sigma_o \) and sample \( z_o \sim \mathcal{N}(\mu_o, \sigma_o) \);

Infer \( \alpha \) from \( r_i \) and \( x_i \);

For VBAE-hard, Sample \( d \sim \text{Bernoulli}(\alpha) \);

For VBAE-soft, Sample \( d \sim \text{LogisticNormal}(\alpha, \theta_{\text{fixed}}) \);

Calculate \( v = z_o + d \cdot z_i \);

Calculate \( r_t \) the multinomial probability of \( r_i \);

Calculate \( L_{MAP}^{t,\text{step}} \) from \( r_t \) and \( r_i \), as Eq. (6).

\[ L_{MAP}^{t,\text{step}} = L_{MAP}^{t,\text{step}} + \]

end

Update \( \Theta_o = \Theta_o - lr \cdot \nabla_{\Theta_o}(-L_{MAP}^{t,\text{step}})/I \)

end

return \( \Theta_t, \Theta_o \), the trained weights of the network.

3.8 Prediction for New Users

After the weights of the generative and inference networks of the VBAE model are learned, our discussion shifts towards how to predict new relevant items for users given their observed ratings \( r_i \) and noisy features \( x_i \). For a user, we first calculate the mean of the collaborative embedding \( \mu_o \), the bandwidth \( \alpha \) from the ratings via the collaborative inference network, and the mean of the feature embedding \( \mu_t \) from the user features via the feature inference network. The user latent variable \( v \) can then be approximated as:

\[
E[V|r_o, x] \approx E[Z_t|r_o] + d \cdot E[Z_t|x] = \mu_b + d \cdot \mu_t.
\]

To avoid randomness of the channel in testing, for VBAE-hard, we set \( d \) to a fixed sample from \( \text{Bernoulli}(\alpha) \) to determine whether or not information in user feature embeddings \( \mu_o \) are necessary to be introduced to support the recommendation. For VBAE-soft, we use the mean of the Beta channel variable (approximated by logistic-normal), \( \alpha \), as \( d \). Finally, we calculate the multinomial probabilities of the remaining items from \( v \) via \( \text{gen}_b \), as:

\[
E[R|r_o, x] = E[\text{gen}_b(V)|r_o, x] \approx \text{gen}_b(E[V|r_o, x]),
\]

where \( \approx \) is due to the non-linearity of \( \text{gen}_b \) and the estimated logits of probabilities of unobserved items are sorted to get final ranked list of items for recommendation.

4 Empirical Study

In this section, we present and analyze extensive experiments we conducted on three real-world datasets to demonstrate the effectiveness of the proposed VBAE model for hybrid recommender systems.

4.1 Datasets

We use three real-world datasets to evaluate the model performance. Two of the datasets, citeulike-a [51] and citeulike-t [52] are from CiteULike, where scholars can add academic articles they are interested in to their libraries such that new relevant articles can be automatically recommended. The third dataset, toys & games, is collected by [53] from Amazon. In preprocessing, we randomly split the users by the ratio of 8:1:1 for training, validation, and testing. For each user, 80% of the interactions are selected as the observed interactions to learn the user collaborative embedding and the bandwidth of the channel, and the remaining 20% are hold-out for testing. The user profiles are built from the features of their interacted items. We represent each article in the citeulike datasets by the concatenation of its title and abstract, and each item in toys & games by combining all of its reviews. We then select discriminative words according to the tf-idf values and normalize the word counts of each item over the maximum occurrences of each word in all items. Finally, we calculate the element-wise maximum of the normalized word counts of the observed items for each user as the user features. Table 1 summarizes the details of the datasets after preprocessing. Fig. 5 illustrates the distributions of interaction density for different users. From Fig. 5 we can find that the interaction density distribution clearly demonstrates a long-tail characteristic, which reflects the uneven distribution of sufficiency level of collaborative information for users in all three datasets.

4.2 Evaluation Metrics

Two ranking-based metrics are used to evaluate the recommendation performance: Recall@M and truncated normalized discounted cumulative gain (NDCG@M). We do not
use the precision metric, since rating matrices in all three datasets record implicit feedbacks where a zero entry does not necessarily imply that the user lacks interests, but it may also indicate that the user is not aware of the item’s existence [34]. For a user \( i \), we first obtain the rank of the hold-out items by sorting their multinomial probabilities. If we denote the item at rank \( r \) by \( j(r) \) and the set of hold-out items for the user by \( \mathcal{J}_i \), Recall@\( M \) is calculated as:

\[
\text{Recall@} M(i) = \frac{\sum_{r=1}^{M} \mathbb{1}[j(r) \in \mathcal{J}_i]}{\min(M, |\mathcal{J}_i|)},
\]

where \( \mathbb{1} \) in the numerator is the indicator function, and the denominator is the minimum of \( M \) and the number of hold-out items. Recall@\( M \) has a maximum of 1, which is achieved when all relevant items are ranked among the top \( M \) positions. Truncated discounted cumulative gain (DCG@\( M \)) is computed as

\[
\text{DCG@} M(i) = \sum_{r=1}^{M} \frac{2^{[j(r) \in \mathcal{J}_i]} - 1}{\log(r + 1)},
\]

which, instead of uniformly weighting all positions, introduces a logarithm discount function over the ranks where larger weights are applied to recommended items that appear at higher ranks [54]. NDCG@\( M \) is calculated by normalizing the DCG@\( M \) to \([0, 1]\) by the ideal DCG@\( M \) where all relevant items are ranked at the top.

### 4.3 Implementation Details

Since the datasets we consider vary both in their scale and scope, we select the structure and the hyperparameters of VBAE based on evaluation metrics on validation users through grid search.\(^3\) In VBAE, sigmoid is used as both intermediate and output activations. The weights of the inference network are tied to the generation network the same way as [19] to more effectively learn representations of user features. Specifically, to avoid the component collapsing problem where the inferred bandwidth for all users are identical, batch normalization [41] is applied to the L2-norm of the latent feature representations such that they have zero mean and unit variance before the inference of the bandwidth; in addition, a larger decay rate is applied to the weights of the dense layer for bandwidth inference for regularization. We first layerwise pretrain the user feature network as the initial starting point for VBAE, and then iteratively train the collaborative network (\( b \_\text{step} \)) and the user feature network (\( t \_\text{step} \)) for 100 epochs. Adam is used as the optimizer with a batch size of 500 users. We randomly split the datasets into ten train/val/test splits as described in Section 4.1. For each split, we keep the model with the best NDCG@100 on the validation users and report the test metrics averaged on ten splits of the datasets.

### 4.4 Baselines

In this section, we compare the proposed VBAE with the following state-of-the-art collaborative and hybrid recommendation baselines to demonstrate its effectiveness:

- **FM** (Factorization Machine) is a widely employed algorithm for hybrid recommendation with sparse inputs [55]. We use Bayesian parameter search as suggested in [56] to find optimal hyperparameters and the loss function on the validation users.
- **CTR** [51] learns the topics of item content via latent Dirichlet allocation (LDA) and couples it with probabilistic matrix factorization (PMF) for collaborative filtering. We find the optimal hyperparameters \( a, b, \lambda_u, \lambda_i \), and latent dimension \( K \) through grid search.
- **CDL** [20] replaces the LDA in CTR with a stacked Bayesian denoising auto-encoder (SDAE) [30] to learn the item content embeddings in an end-to-end manner. We set the mask rate of SDAE to 0.3 and search its architecture the same as VBAE.
- **CVAE** [19] further improves over the CDL by utilizing a VAE in place of the Bayesian SDAE, where a self-adaptive Gaussian noise is introduced to corrupt the latent item embeddings instead of corrupting the input features with zero masks.
- **Multi-VAE** [12] breaks the linear collaborative modeling ability of PMF by using a VAE with multinomial likelihood to capture the user collaborative information in ratings for recommendations.
- **CoVAE** [57] utilizes the non-linear Multi-VAE as the collaborative backbone and incorporates item feature information by treating their co-occurrences as pseudo training samples to collectively train the Multi-VAE with the item features.
- **CondVAE** [32] builds a user conditional VAE where the user features are used as the conditions. We extend the original CondVAE by replacing the categorical

---

\(^3\) Due to space limit, please refer to the JSON files we release with the codes for the searched optimal hyperparameters and model architecture for each of the three datasets.
user features with the ones we build from the interacted items, which we find have a better performance on all the datasets.

- **DICER** [28] is an item-oriented auto-encoder (IAE)-based recommender system where the item content information is utilized to learn disentangled item embeddings from their user ratings to achieve more robust recommendations.

- **RecVAE** [58] improves over the Multi-VAE by designing a new encoder architecture with a composite prior for user collaborative latent variables, which leads to a more stable training procedure.

Table 2 summarizes the comparison results between the two VBAE models and the selected baselines. As it can be seen, Table 2 comprises of three parts. The middle part shows four hybrid baselines with linear collaborative filtering module, i.e., matrix factorization (MF). Generally, the performance improves with the increase of the representational ability of the utilized item embedding model. Specifically, CVAE, which uses VAE to encode the item content information into Gaussian variables, performs consistently better than CDL and CTR on all three datasets. However, we also observe that simple methods such as FM can outperform some of the deep learning-based baselines (e.g., CDL on citeulike-a datasets) when their parameters are systematically searched with a Bayesian optimizer [56].

### 4.5 Comparison Analysis

The bottom part shows baselines that utilize deep neural networks (DNNs) as the collaborative module. Multi-VAE, RecVAE can capture non-linear similarities among users, so they improve almost consistently over the linear hybrid baselines when the datasets are comparatively dense (e.g., the citeulike-a dataset), even if they do not use any user or item side information. When the datasets get sparser, however, they cannot perform on par with the PMF-based hybrid recommenders that augments with item side information due to lack of sufficient collaborative information. Moreover, we find that although augmenting user ratings with item feature concurrences as extra pseudo training samples, CoVAE does not consistently outperform Multi-VAE on all three datasets, which could suggest that the item feature occurrences do not necessarily imply user co-purchases. Treating user feature embeddings as the condition for the user collaborative embeddings, CondVAE achieves the best performance among all the non-linear UAE-based baselines on two of the denser citeulike datasets and performs on par with CVAE on the sparser Amazon toys & games dataset. DICER, which is an IAE-based recommender that we include for comparisons, shows clear merits when the dataset has a large item-to-user ratio (e.g. the citeulike-a and citeulike-t datasets). The reason may be that for IAE-based recommenders, the number of training samples is proportional to the number of items, whereas the number of trainable weights is proportional to the number of users, so a large item-to-user ratio ensures sufficient training samples and a reasonable amount of trainable weights to guarantee a good model generalization ability.

Simultaneously addressing the uncertainty of user ratings and noise in user features, VBAE-soft and VBAE-hard outperforms all baselines on all three datasets. Although the Bayesian SDAE in CDL, VAE in CVAE, RecVAE, or CondVAE also have the denoising ability in that they corrupt the item features, latent item embeddings, or latent user embeddings via masked noise or self-adaptive Gaussian noise, the noise they address is not recommendation-oriented and is therefore inevitably low-level. However, high-level and personalized noise (information that is not relevant to the recommendation purpose) exists pervasively in recommendation tasks, which cannot be addressed by these models. In contrast, through the introduction of a user-dependent channel variable, VBAE actively decides how much information should be accessed from the user features based on information already contained in the ratings through a quantum-inspired collaborative uncertainty measurement mechanism. This ensures the quality of personalized recommendations.
when the ratings are sparse by incorporating sufficient user feature information while improving the model generalization ability by avoiding unnecessary dependence on the noisy user features when the collaborative information is sufficient.

4.6 Ablation Study on the User-Dependent Channel

In this section, we further demonstrate the effectiveness of the established information regulation mechanism in VBAE by answering the following two research questions:

RQ1: How do VBAE-hard and VBAE-soft perform compared to VBAE-like models, which, instead of explicitly considering the personal difference in the sufficiency level of collaborative information and the noise level of user features, treat the fusion of user feature information into user variables as a fixed procedure for all the users.

RQ2: How does the dynamic channel perform compared with the vanilla attention mechanism, and how well does the inferred bandwidth correspond to the deficiency of collaborative information? The answer to this question shows the effectiveness of the proposed quantum-inspired collaborative uncertainty measurement to distinguish users with varied sufficiency levels of collaborative information.

4.6.1 Comparisons With Ablation Baselines

To answer RQ1, we design the following five baseline models as ablation studies:

- **DBAE-pass** uses an “all-pass” channel to connect the user collaborative and feature networks, where all the information in user feature embeddings are losslessly transferred to the corresponding user latent variables irrespective of the individual difference in the sufficiency level of the collaborative information;
- **DBAE-stop** uses a “stop” channel where the user feature information is indiscriminately blocked, and only the collaborative information is exploited to calculate the user latent variables. The difference between DBAE-stop and Multi-VAE is that Multi-VAE imposes the L2-normalization on the input ratings.
- **VAE-concat** concatenates the user ratings and features as the inputs to the Multi-VAE to reconstruct the ratings, instead of viewing their fusion from an information-theoretic perspective. The fusion can be viewed as learning a fixed weighted combination between user features and ratings for all users.
- **VAE-attn** uses the vanilla attention mechanism to fuse the user collaborative and feature embeddings. The attention weights for the rating and feature embeddings are calculated by $\alpha_{(k,l)} = \frac{e^{w_{(k,l)} h_{(k,l)}}}{\sum_{(k,l)} e^{w_{(k,l)} h_{(k,l)}}}$, where $w_{(k,l)}$ are trainable embedding vectors.
- **VBAE-nn** uses a simple neural network to calculate the bandwidth $a$ from the user collaborative embeddings $h_u$. The structure of the network is determined by grid search. This model is used to further verify the effectiveness of the proposed quantum-inspired semantic-uncertainty inference strategy.

The comparison results are listed in Table 3. From Table 3 we can find that, among the models that we draw comparisons with, DBAE-stop performs the worst on all the datasets.

### Table 3

| Model | Recall@20 | NDCG@100 | Bandwidth | PCC  |
|-------|-----------|-----------|-----------|------|
| VBAE-soft | 0.299 | 0.296 | 0.543 ± 0.054 | -0.898 |
| VBAE-hard | 0.293 | 0.294 | 0.812 ± 0.048 | -0.901 |
| VBAE-nn (soft) | 0.292 | 0.291 | 0.610 ± 0.046 | -0.871 |
| VBAE-nn (hard) | 0.289 | 0.287 | 0.925 ± 0.031 | -0.883 |
| DBAE-stop | 0.263 | 0.269 | 0.000 ± 0.000 | N/A |
| DBAE-pass | 0.287 | 0.285 | 1.000 ± 0.000 | N/A |
| VAE-attn | 0.283 | 0.286 | N/A | N/A |
| VAE-concat | 0.274 | 0.280 | N/A | N/A |
| VBAE-hard | 0.227 | 0.190 | 0.546 ± 0.050 | -0.887 |
| VBAE-hard | 0.223 | 0.193 | 0.805 ± 0.035 | -0.910 |
| VBAE-nn (soft) | 0.221 | 0.186 | 0.615 ± 0.043 | -0.882 |
| VBAE-nn (hard) | 0.218 | 0.184 | 0.886 ± 0.037 | -0.903 |
| DBAE-stop | 0.170 | 0.142 | 0.000 ± 0.000 | N/A |
| DBAE-pass | 0.212 | 0.178 | 1.000 ± 0.000 | N/A |
| VAE-attn | 0.219 | 0.177 | N/A | N/A |
| VAE-concat | 0.215 | 0.172 | N/A | N/A |
| VBAE-soft | 0.145 | 0.107 | 0.560 ± 0.057 | -0.803 |
| VBAE-hard | 0.144 | 0.105 | 0.829 ± 0.031 | -0.825 |
| VBAE-nn (soft) | 0.139 | 0.103 | 0.643 ± 0.052 | -0.814 |
| VBAE-nn (hard) | 0.137 | 0.099 | 0.901 ± 0.036 | -0.828 |
| DBAE-stop | 0.119 | 0.088 | 0.000 ± 0.000 | N/A |
| DBAE-pass | 0.135 | 0.094 | 1.000 ± 0.000 | N/A |
| VAE-attn | 0.136 | 0.097 | N/A | N/A |
| VAE-concat | 0.132 | 0.095 | N/A | N/A |

The ± in the bandwidth column denotes the std of users’ bandwidth averaged over ten splits of the datasets.

Since DBAE-stop can be viewed as an altered version of Multi-VAE [19] where the L2-normalization is applied on the hidden representations rather than the input ratings, this confirms the previous finding that hybrid recommendation methods augmented with feature information usually perform better than collaborative-filtering-based methods when the ratings are sparse [19], [20]. Comparatively, DBAE-pass is more difficult to beat, since the deficiency of collaborative information for users with sparse interaction makes the auxiliary user feature information valuable even if they are noisy. Still, two VBAE-based methods achieve better performance on all three datasets, which demonstrates that dynamically regulating the feature information allowed to be accessed by user latent variables can indeed improve model generalization. VAE-concat uses a dense layer to learn a weighted combination of user features and ratings, which may be over-parameterized and prone to overfitting when datasets are sparse. Moreover, the weights are the same for all users, which fails to distinguish users with different sufficiency levels of collaborative information. The superiority of VBAE with user-dependent bandwidth to VBAE-pass and VAE-concat indicates that for users with more informative interactions (i.e., denser and overlapped with the...
interactions of other users), the collaborative information is per se very reliable for recommendations, and the noise introduced by the fusion of user features may outweigh the useful information and degenerate the recommendation performance. Finally, the favorable comparisons of VBAE-soft and VBAE-hard with VAE-attn and two VBAE-nn models demonstrate the effectiveness of the proposed quantum-inspired semantic-uncertainty measurement of the collaborative information, where the inferred uncertainty of collaborative information via the length of hidden collaborative embedding can better reflect the relative importance between the user ratings and features.

The explanation for the general superiority of VBAE-soft over VBAE-hard could be that VBAE-soft uses a Beta channel variable with its variance fixed to a small value, where the feature embeddings are stably and smoothly discounted based on the bandwidth inferred from user ratings. In contrast, the Bernoulli channel in VBAE-hard determines whether or not to access user features with the inferred bandwidth as the access probability, which may be coarse in granularity and makes the training process less stable than the Beta channel in VBAE-soft. However, we also observe an exception that VBAE-hard has better NDCG@100 than VBAE-soft on citeulike-t dataset. Based on the attributes of the three datasets summarized in Table 1, we speculate the reason could be that since the features of citeulike-t are established by calculating the normalized word counts for words with top 20,000 tf-idf values instead of 8,000 (for both citeulike-a and toys & games datasets), the features in citeulike-t dataset are nosier, which makes completely discarding the noisy feature information in VBAE-hard more advantageous compared with discounting it through the soft channel.

4.6.2 NDCG Breakdown for Users in Different Active Levels

To further investigate the effectiveness of the user-dependent channel to fuse the collaborative and feature information for users with different activity levels, we divide the test users into quartiles and report the NDCG@100 on each group in Fig. 6. When comparing with DBAE-stop, we mainly focus on users with low activity levels, since for these users, DBAE-stop accesses no information from user features while VBAE-hard and VBAE-soft infer a large channel bandwidth that allows more information to be accessed from user features. The leftmost bar group in Fig. 6 shows that VBAE-hard and VBAE-soft significantly outperform DBAE-stop on all three datasets. The result confirms that incorporating auxiliary feature information can alleviate the uncertainty of user collaborative embeddings and improve recommendation performance when the ratings are extremely sparse, even if user features are generally noisy. When comparing with DBAE-pass, on the other hand, we focus on users with high activity levels. Although for these users, VBAE-hard and VBAE-soft access less information from user features than DBAE-pass, the rightmost bar group of Fig. 6 shows that NDCG@100 improves consistently for these users. This indicates that for users with dense interactions, the collaborative information in the ratings is per se very reliable for recommendations, and the noise introduced by the fusion of user features may outweigh the useful information and lower the recommendation performance. The improvement is more significant on citeulike-t and toys & games datasets. Recall that Table 1 shows that users in these two datasets span a wider spectrum in their activity levels, and therefore the reliability of collaborative embeddings varies drastically for these users. In such a case, the channel can better distinguish these users and allocate for each user a suitable budget for user feature information when calculating the user latent variables for recommendations.

4.6.3 Statistical Analysis of the Bandwidth

To answer RQ2, we calculate several statistics of the inferred bandwidth for all test users: its averaged value, its user variability, and its Pearson correlation coefficient (PCC) with the interaction density, and report the results in Table 3. Table 3 shows that the bandwidth inferred through the proposed quantum-inspired collaborative uncertainty measurement tends to vary across users with different rating sparsity levels. Moreover, the bandwidth has an over -0.8 PCC with the density of user interactions on all three datasets. Such results indicate that the channel in VBAE-hard and VBAE-soft can well distinguish users with different amounts of collaborative information in their ratings and dynamically control the extra amount of information that needs to be accessed from the user features based on the inferred bandwidth, which more convincingly demonstrates the effectiveness of the user-dependent channel in VBAE-hard and VBAE-soft. In addition, the average bandwidth of VBAE-hard is significantly larger than that of VBAE-soft on all three datasets. The reason could be that a large bandwidth for VBAE-hard is conducive to maintaining the stability of the Bernoulli channel in the training phase.
4.7 Discussions of Broader Impact of VBAE

Although we demonstrate the effectiveness of VBAE by its application in recommender systems, VBAE is a general framework that is applicable to any heterogeneous information system where one information source is comparatively reliable but could be missing, whereas another information source is abundant but is susceptible to noise. One typical example is the "audio-assisted action recognition in the dark" task [59], which aims to detect actions in under-illuminated videos. In the task, the visual information is more reliable for action prediction but could be missing due to bad illumination, whereas the audio track always accompanies the video but may contain lots of irrelevant information (e.g., background music) for the action recognition purpose.

To apply VBAE to these new tasks, the only mandatory change required is to design a suitable per data point uncertainty measurement of the primary information source to dynamically decide the information allowed to be accessed from the second source, so that the model will not overfit on the noise in the second auxiliary modality.

Moreover, VBAE can be readily generalized to other multimodal information systems where more than two modalities exist. One strategy is to identify one modality that is most reliable but is also most susceptible to missing data problems as the primary modality (such a modality probably exists because high reliability of an information source usually means it is difficult or expensive to obtain) and the remaining modalities as the auxiliary modalities. The embeddings from auxiliary modalities can be fused with common techniques such as concatenation or product-of-experts principle. Then, a dynamic channel similar to the one used VBAE can be established between the primary and auxiliary modalities to dynamically fuse the multimodal information based on the information sufficiency of the primary modality. Therefore, we speculate that VBAE could have a broader impact on areas of data mining and heterogeneous information systems other than the recommendation tasks that we have discussed in this article.

5 Conclusions

In this article, we develop an information-driven generative model, collaborative variational bandwidth auto-encoder (VBAE), to address uncertainty and noise problems associated with two heterogeneous sources, i.e., ratings and user features in recommender systems. In VBAE, we establish an information regulation mechanism to fuse the collaborative and feature information, where a user-dependent channel variable is introduced to dynamically control how much information should be accessed from the user features given the information already contained in the collaborative embedding. The channel alleviates the uncertainty problem when the ratings are sparse while improving the model generalization ability with respect to noisy user features. The effectiveness of VBAE is demonstrated by extensive experiments conducted on three real-world datasets.

References

[1] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” ACM Comput. Surv., vol. 52, no. 1, pp. 1–38, 2019.
[2] H. Wang and W.-J. Li, “Relational collaborative topic regression for recommender systems,” IEEE Trans. Knowl. Data Eng., vol. 27, no. 5, pp. 1343–1355, May 2015.
[3] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, “Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach,” Proc. AAAI Conf. Artif. Intell., vol. 34, no. 01, pp. 27–34, 2020.
[4] B. Xu, J. Bu, C. Chen, C. Wang, D. Cai, and X. He, “EMR: A scalable graph-based ranking model for content-based image retrieval,” IEEE Trans. Knowl. Data Eng., vol. 27, no. 1, pp. 102–114, Jan. 2015.
[5] J. Yi, Y. Zhu, J. Xie, and Z. Chen, “Cross-modal variational autoencoder for content-based micro-video background music recommendation,” IEEE Trans. Multimedia, early access, Nov. 16, 2021, doi: 10.1109/TPAMI.2021.3128254.
[6] J. Dong et al., “Dual encoding for video retrieval by text,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Feb. 15, 2021, doi: 10.1109/TPAMI.2021.3059295.
[7] P. Sun, L. Wu, and M. Wang, “Attentive recurrent social recommendation,” in Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2018, pp. 185–194.
[8] J. Xie et al., “A multimodal variational encoder-decoder framework for micro-video popularity prediction,” in Proc. Web Conf., 2020, pp. 2542–2548.
[9] X. Chen, D. Liu, Z. Xiong, and Z.-J. Zha, “Learning and fusing multiple user interest representations for micro-video and movie recommendations,” IEEE Trans. Multimedia, vol. 23, pp. 484–496, 2020.
[10] Y. Xu, L. Zhu, Z. Cheng, J. Li, Z. Zhang, and H. Zhang, “Multi-modal discrete collaborative filtering for efficient cold-start recommendation,” IEEE Trans. Knowl. Data Eng., early access, May 12, 2021, doi: 10.1109/TKDE.2021.3079581.
[11] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, “AutoRec: Autoencoders meet collaborative filtering,” in Proc. 24th Int. Conf. World Wide Web, 2015, pp. 111–112.
[12] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, “Variational autoencoders for collaborative filtering,” in Proc. World Wide Web Conf., 2018, pp. 689–698.
[13] S. Li, J. Kawale, and D. Fu, “Deep collaborative filtering via marginalized denoising auto-encoder,” in Proc. 24th ACM Int. Conf. Inf. Knowl. Manage., 2015, pp. 811–820.
[14] H. Wang, X. Shi, and D.-Y. Yeung, “Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 415–423.
[15] Z. Zhu, J. Wang, and J. Caverlee, “Improving top-k recommendation via joint collaborative autoencoders,” in Proc. World Wide Web Conf., 2019, pp. 3483–3482.
[16] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” in Proc. Int. Conf. Learn. Representations, 2014.
[17] S. Zhang, L. Yao, and X. Xu, “AutoSVD++: An efficient hybrid collaborative filtering model via contractive auto-encoders,” in Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2017, pp. 957–960.
[18] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, “Collaborative denoising auto-encoders for top-N recommender systems,” in Proc. 9th ACM Int. Conf. Web Search Data Mining, 2016, pp. 153–162.
[20] X. Li and J. She, “Collaborative variational autoencoder for recommender systems,” in Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2017, pp. 305–314.
[21] H. Wang, N. Wang, and D.-Y. Yeung, “Collaborative deep learning for recommender systems,” in Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2015, pp. 1235–1244.
[22] J. Ma, C. Zhou, P. Cui, H. Yang, and W. Zhu, “Learning disentangled representations for recommendation,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 5711–5722.
[23] H. Hou, N. Yang, and P. Yu, “Dual adversarial variational autoencoder: Recommend while learning to fill in the blanks,” in Proc. 9th ACM Conf. Inf. Knowl. Manage., 2017, pp. 1139–1148.
27] N. Sachdeva, G. Manco, E. Ritacco, and V. Pudi, “Sequential variational autoencoders for collaborative filtering,” in Proc. 12th ACM Int. Conf. Web Search Data Mining, 2019, pp. 600–608.

28] Y. Zhang, Z. Zhu, Y. He, and J. Caverlee, “Content-collaborative disentanglement representation learning for enhanced recommendation,” in Proc. 10th ACM Conf. Recommender Syst., 2020, pp. 43–52.

29] A. Mnih and R. Salakhutdinov, “Probabilistic matrix factorization,” in Proc. Adv. Neural Inf. Process. Syst., 2007, pp. 1257–1264.

30] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” J. Mach. Learn. Res., vol. 11, no. Dec. 3371–3408, 2010.

31] Y. Koren, “Factorization meets the neighborhood: A multifaceted collaborative filtering model,” in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2008, pp. 426–434.

32] B. Pang, M. Yang, and C. Wang, “A novel top-N recommendation approach based on conditional variational auto-encoder,” in Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining, 2019, pp. 357–368.

33] X. Li et al., “Adversarial multimodal representation learning for click-through rate prediction,” in Proc. Web Conf., 2020, pp. 827–836.

34] C. Wang and D. M. Blei, “Collaborative topic regression,” in Proc. 25th Int. Conf. World Wide Web, 2016, pp. 507–517.

35] Y. Wang, L. Wang, Y. Li, D. He, W. Chen, and T.-Y. Liu, “A theoretical analysis of NDCG ranking measures,” in Proc. 26th Annu. Conf. Learn. Theory, 2013, pp. 1–30.

36] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, “DeepFM: A factorization-machine based neural network for ctr prediction,” in Proc. Int. Joint Conf. Artif. Intell., 2017, pp. 1725–1731.

37] M. F. Dacrema, P. Cremonesi, and D. Jannach, “Are we really making much progress? reasoning analysis of recent top-N recommendation approaches,” in Proc. 13th ACM Recom. Syst. Conf., 2019, pp. 101–109.

38] Y. Chen and M. de Rijke, “A collective variational autoencoder for top-N recommendation with side information,” in Proc. 3rd Workshop Deep Learn. Recommender Syst., 2018, pp. 3–9.

39] I. Sheshin, A. Alekseev, E. Tutubalina, V. Malykh, and S. I. Nikolenko, “RecVAE: A new variational autoencoder for top-N recommendations with implicit feedback,” in Proc. 13th Int. Conf. Web Search Data Mining, 2020, pp. 528–536.

40] Y. Xu, J. Yang, H. Cao, J. Yin, and S. See, “Arid: A new dataset for recognizing action in the dark,” Proc. Int. Joint Conf. Artif. Intell. Workshop, vol. 1370, p. 70, 2021.

Yaochen Zhu received the BE degree from Central South University, Changsha, China, in 2019. He is currently working toward the master’s degree in pattern recognition and intelligent systems with Wuhan University, Wuhan, China. His research interests include deep probabilistic and causal reasoning, especially in user modeling, recommender systems, and multimedia content analysis.

Zhenzhong Chen (Senior Member, IEEE) received the BEng degree in electrical engineering from the Huazhong University of Science and Technology and the PhD degree in electrical engineering from the Chinese University of Hong Kong. He is currently a professor with Wuhan University. His current research interests include image and video processing and analysis, computational vision and human-computer interaction, multimedia data mining, photogrammetry, and remote sensing. He is a VQEG board member and Immersive Media Working Group co-chair, a selection committee member of ITU Young Innovators Challenges, an associate editor for the IEEE Transactions on Image Processing (TIP), IEEE Transactions on Circuits and Systems for Video Technology (TCST), Journal of the Association for Information Science and Technology (JASIST), Journal of Visual Communication and Image Representation, and the editor of IEEE IoT Newsletters. He was the recipient of CUHK Young Scholars Dissertation Award, CUHK Faculty of Engineering Outstanding Ph.D. Thesis Award, and Microsoft Fellowship.