EXPLAINABLE RECOMMENDER WITH GEOMETRIC INFORMATION BOTTLENECK

Anonymous authors
Paper under double-blind review

ABSTRACT

Explainable recommender systems have attracted much interest in recent years as they can explain their recommendation decisions, enhancing user trust in the systems. Most explainable recommender systems rely on human-generated rationales or annotated aspect features from user reviews to train models for rational generation or extraction. The rationales produced are often confined to a single review. To avoid the expensive human annotation process and to generate explanations beyond individual reviews, we propose an explainable recommender system trained on reviews by developing a transferable **G**eometric **I**nform**A**tio**N** bo**T**tleneck (**`GIANT`**), which leverages the prior knowledge acquired through clustering on a user-item graph built on user-item rating interactions, since graph nodes in the same cluster tend to share common characteristics or preferences. We then feed user reviews and item reviews into a variational network to learn latent topic distributions which are regularised by the distributions of user/item estimated based on their distances to various cluster centroids of the user-item graph. By iteratively refining the instance-level review latent topics with **`GIANT`**, our method learns a robust latent space from text for rating prediction and explanation generation. Experimental results on three e-commerce datasets show that our model significantly improves the interpretability of a variational recommender using a standard Gaussian prior, in terms of coherence, diversity and faithfulness, while achieving performance comparable to existing content-based recommender systems in terms of rating prediction accuracy.

1 INTRODUCTION

Typically, a recommender system compares users' preferences with item characteristics (e.g., item descriptions or item-associated reviews) or studies user-item historical interactions (e.g., ratings, purchases or clicking behaviours) in order to identify items that are likely of interest to users. In addition to predictive performance, interpretable recommenders aim to give rationale behind the rating given by a user on an item (Ghazimatin et al., 2020; Zhang et al., 2020). Most existing interpretable recommenders can either generate rationale or extract text spans from a given user-item review as explanations of model decisions. Both rationale generation and extraction require annotated data for training, e.g., short comments provided by users explaining their behaviours of interacting with items, or annotated sentiment-bearings aspect spans in reviews (Zhang et al., 2014; Ni et al., 2019; Chen et al., 2019; Li et al., 2020; Tan et al., 2021a).

We argue that generating explanations based on a specific user-item review document suffers from the following limitations. First, some reviews may be too general to explain the rating, rendering them useless for explanation generation. For example, the review '*I really like the smartphone, will recommend it to my friends*' does not provide any clue why the user likes the smartphone. Second, features directly extracted from a review document may fail to reflect some global properties which can only be identified from implicit user-item interactions. For example, meaningful insights could still be derived from reviews towards items that are not directly purchased/rated by a user but preferred by other like-minded users. Finally, learning an explanation generation model from user/item reviews often requires text spans annotated as rationales explaining the corresponding review ratings. Such annotations are labour-intensive to obtain in practice.

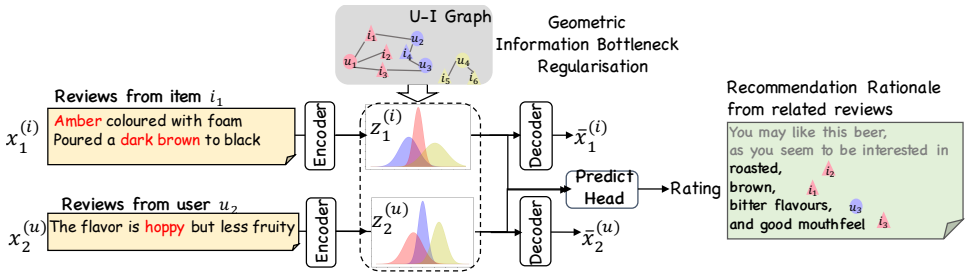

Figure 1: An encoder-decoder structure with a geometric information bottleneck regularisation, which is derived from the user-item interaction graph and used as a prior imposed on Z. The latent variable z thus can capture the geometric affiliations in graph and be the soft cluster assignment distributions. It essentially enables the use of like-minded users u_3 and similar items i_2 and i_3 for generating explanations beyond the input user-item pair (i_1, u_2). Existing rationale extraction methods can only extract indicative words (shown in red) from the given review pair. Our method can generate rationale from reviews written by other like-minded users or on similar items, which are assigned into the same latent dimension/cluster. (Refer to Table 3 for words extraction.)

To address the aforementioned limitations, we propose an AutoEncoder (AE) framework with variational **G**eometric **I**nform**A**tio**N** bo**T**tleneck (**GIANT**) to incorporate the distributions of users and items derived from the user-item interaction graph as regularisation to refine the induced latent factors of user and item. For a user-item pair, all reviews written by the user and reviews posted on the item are fed into two separate encoders to infer latent factors of users/items, and predict rating based on the match of their latent factors. The latent variables are supposed to capture the key semantic information to recover the original review text written by the user on the item. Different reviews are assigned to different latent dimensions according to posterior distributions and we extract the representative words in each dimension to offer recommendation explanations.

Our proposed framework is illustrated in Figure 1. The geometric regularisation refers to the cluster-based distance with Gaussian variance, which is derived by firstly clustering the users/items in the user-item interaction graph, and then calculating the distribution of a user/item as its distance to each cluster centroid by a Gaussian kernel (§ 4.1). To impose the regularisation on the latent variables of the AE framework taking the set of user reviews and item reviews as input, we apply the KL divergence to minimize the discrepancy between the induced posterior distribution of users $z^{(u)}$, and items $z^{(i)}$, and the geometric regularisation as the prior (§4.2), after linking the cluster to the review text encoder via prior-centralisation (§4.2).

Experimental results on the three commonly used benchmarking datasets show that the proposed method achieves performance comparable with several strong baselines in recommendation. Moreover, the quantity and the quality analysis in interpretability show that our method can generate coherent, diverse and faithful explanations.

2 Related Work

We review recommender systems, with particular attention to those built on Variational Autoencoder (VAE) (Kingma & Welling, 2014) or offering explanations of recommendation decisions.

Recommender with variational autoencoder Text data such as user reviews, item or brand descriptions could be important for developing a high-quality recommender system as they can be exploited to address the sparsity issue in user-item interactions. HFT (McAuley & Leskovec, 2013) and CTR (Wang & Blei, 2011) adopted Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to extract latent topics from review text. VAE can also be used for text modeling due to its ability in extracting latent and interpretable features (Fei et al., 2021; Truong et al., 2021; Wang et al., 2020). Truong et al. (2021) argued that the commonly used isotropic Gaussian in VAE is over-simplified and proposed BiVAE by introducing constrained adaptive prior (CAP) for learning user- and item-dependent prior distributions. More recently, review information is used to disentangle the latent user intents at the finer granularity, for example, DisenGCN (Ma et al., 2019a) and DisenHan (Wang et al., 2020) used the graph attention mechanism to differentiate multiple relations and features.

Explainable Recommender System One popular interpretable method is to train a language generation model with the ground-truth explanations supplied, which can be the first sentence of a given review or human-annotated text spans in the review text (Li et al., 2017; Chen et al., 2019; Ni et al., 2019; Li et al., 2022). Feature-level explanations aim to provide the important item features the users care about when making decisions (Zhang & Chen, 2020). Explicit Factor Model adopted Matrix Factorization (MF) to predict review ratings and aligned each dimension of latent factors of the MF results with some item features, thus explaining the recommendation decision by the corresponding features (Zhang et al., 2014). Other approaches firstly processed the review text to extract user-aspect-opinion and item-aspect-opinion tuples. They then modeled the interactions between the two tuples to derive the rating and the indicative features (Chen et al., 2016; Wang et al., 2018; Tan et al., 2021b; Chen et al., 2020). However, the aforementioned approaches suffer from two major issues: they rely heavily on the accuracy of sentiment analysis tools (Guan et al., 2018); they tend to ignore implicit interactions not expressed directly in the individual review text.

3 GEOMETRIC INFORMATION BOTTLENECK

To consider implicit user-item interactions for developing an interpretable recommender system, we propose to incorporate the geometric regularisation derived from user-item interaction graphs to learn the latent factors of users and items from review text in a variational network. Although various work has been proposed to adopt richer priors to infer a more complex and realistic posterior in standard VAEs (Tomczak & Welling, 2017; Zhao et al., 2017), our solution is different from existing approaches as we need to impose the prior knowledge learned from the graph modality to constrain the learning of latent variables in the text modality. In what follows, we show how this can be done under the theory of information bottleneck.

Based on the Information Bottleneck (*IB*) theory (Tishby & Zaslavsky, 2015), we can train an encoder-decoder architecture $x \xrightarrow{encode} z \xrightarrow{decode}, \bar{x}$, where $x \in X$, $z \in Z$, $\bar{x} \in \bar{X}$ are the input, hidden, and reconstructed representations respectively, by optimising

$$\max_{\theta} I(\bar{X}, Z|\theta) \quad \text{s.t.} \quad I(X, Z|\theta) \leq I_c, \tag{1}$$

where $I(\cdot)$ is the mutual information and θ is the parameter in the encoder-decoder. The goal of maximising the mutual information in Eq. (1) is to learn an encoding Z that can maximally reconstruct $\bar{X}$ while maximally compress X under a constraint I_c. In this paper, the goal is to build a mechanism which is able to transfer the constraint from graph to text, i.e., across different modalities. Intuitively, we assume the input x has representations $x_t \in X_t$ and $x_g \in X_g$ under different modalities[1] satisfying the constraint, $I(X_t, X_g) \geq I_x$, since X_t and X_g are derived from the same input, they should be relevant. Then, for two independent encoder-decoder architectures optimised by Eq. (1) with the constraints, $I(X_t, Z_t) \leq I_c$ and $I(X_g, Z_g) \leq I_c$, the following property holds [2]:

$$I(X_t, Z_g) \leq H(X_t) - H(X_g) + H(X_t|X_g) + I_c \tag{2}$$

where $H(\cdot)$ and $H(\cdot|\cdot)$ are entropy and conditional entropy, respectively. Equivalently, with the introduction of a Lagrange multiplier β, we can maximise the objective function:

$$O_{IB} = I(\bar{X}_t, Z_t) - \beta \cdot I(X_t, Z_g) \tag{3}$$

According to (Alemi et al., 2017), the above lower bound can be approximated by applying the reparameterisation trick (Kingma & Welling, 2013) with a random Gaussian noise ϵ:

$$O_{IB} = \Sigma_{n=1}^{N} \mathbb{E}_{\epsilon \sim p(\epsilon)} [- \underbrace{\log q(\bar{x}_n | f(x_n, \epsilon)]}_{\text{Recons. term: Q}(\text{x}_\text{n})} / N + \beta \cdot \underbrace{\text{KL}(p(Z_t|x_n, \epsilon)|r(Z_g))}_{\text{KL-div. term: KL}(\text{Z}_\text{t}|\text{Z}_\text{g})}, \tag{4}$$

where the x_n is the training sample from one modality X_t, $\bar{x}_n$ is the VAE-based reconstruction result, $q(\cdot)$ and $p(\cdot)$ are the posterior probabilities estimated by decoder and encoder respectively, $r(Z_g)$ is the estimated distribution from other modality. Here, Eq. (4) has a similar form to β-VAE which contains a reconstruction term $Q(x_n)$ defined by posterior probability and a KL-divergence term $\text{KL}(\text{Z}_\text{t}|\text{Z}_\text{g})$ regularised by the prior from other modality Z_g. By this equation, we are able to guarantee that the information bottleneck between two modalities are:

[1] In the rest of paper, we use t for text and g for graph.

[2] The proof is shown in the Appendix B.

- **Transferable** – According to Eq. (2), the information bottleneck between modalities is still constrained by the global information bottleneck, which means that we can take the bound from one modality to guide the training on the other.
- **Practicable** – One challenge in the training of variational models is the choice of the prior distribution. By Eq. (2), we do not have to utilise a pre-defined prior but instead use the posterior constraints from other modalities, which are more flexible and efficient.
- **Interpretable** – The alignment between representations from different modalities and corresponding probabilities defined on geometric space makes it possible to use modalities offering better interpretability, such as review text, to explain the black-box features from other modalities.

4 GIANT FRAMEWORK

We propose a novel explainable recommender built on VAE with the geometric information bottleneck. For a given $(user, item)$ pair, the input to our recommender is a set of reviews written by the user and a set of reviews on the item. As each user or item is associated with multiple reviews, and some reviews might be very long, the Convolutional Neural Network (CNN) is used to encode reviews due to its efficiency in encoding long sequences, as have been previously studied in Chen et al. (2018); Zheng et al. (2017). We then generate the user or item representation by aggregating their associated multiple reviews by the attention mechanism. The final contextual vector for user u and item i is denoted as $x^{(u)}$ and $x^{(i)}$, respectively.

Given the contextual vectors, we employ two geometric regularised variational networks to infer their latent factors, in order to better capture their latent semantic topics. According to Eq. (2), we use graph-based Z_g as the transferable information bottleneck to regularise the optimisation of the text-based Z_t. To ensure feasibility and robustness, we propose an initialisation trick called prior-centralisation which uses the Gaussian kernel based geometric estimation to regularise the initialisation and optimisation of the encoders based on the transferable information bottleneck. In the experiments section, we verify the interpretability of our proposed method by the alignment between the two latent feature spaces from graph and text modalities.

4.1 DERIVING PRIORS FROM USER/ITEM CLUSTERS IN INTERACTION GRAPH

In the user-item bipartite graph $\mathcal{G}$, a node can be either a user or an item. For each user-item pair, an edge is created if the user has previously rated on the item and the rating score is higher than the average rating score calculated across all user-item ratings.[3] We use the *LightGCN* (He et al., 2020) to encode our interaction graph and obtain the user and item embeddings.

Once user/item embeddings are learnt, we apply K-means on the learned node embeddings to derive the cluster centroid vectors, denoted as $\boldsymbol{C}_k^u$ for users and $\boldsymbol{C}_k^i$ for items, $k \in \{1, 2, \cdots, K\}$. We use the Radial Basis Function (RBF) kernel to compute the distance between a user or item with the cluster k as the probability of assigning the user or item j to the k-th cluster, ρ_{jk}:

$$\rho_{jk} = \frac{\exp\left(-\left\|\boldsymbol{e}_j - \boldsymbol{C}_k\right\|^2 / 2\alpha^2\right)}{\sum_{k'=1}^{K} \exp\left(-\left\|\boldsymbol{e}_j - \boldsymbol{C}_{k'}\right\|^2 / 2\alpha^2\right)}, \tag{5}$$

where $\boldsymbol{e}_j$ is graph node embedding for user or item j, $\boldsymbol{C}_k$ is k-th cluster centroid vector for user embeddings or item embeddings, α is a hyper-parameter adjusted according to the data density. The cluster-based distribution $\boldsymbol{\rho} = \{\rho_{jk}\}$ is then used to regularise the latent factors for users and items, as will be shown in § 4.2.

According to definition given by Eq. (5), we have $\Sigma_k \rho_{jk} = 1$, which yields that the graph-feature based latent representation $Z_g = \{\sqrt{\rho_{jk}}\}$ are distributed on a S^{K-1} hyper-sphere. By the theorem given by (Diaconis & Freedman, 1987), the distribution of $\{\sqrt{\rho_{jk}}\}$, which yields a hyper spherical cap area on an uniform K-ball, weakly approximates to a Gaussian distribution with the increasing value of K. Therefore, through the clustering, we obtain a latent vector Z_g and a corresponding geometric space where the difference on each basis follows a Gaussian prior.

[3] We have also experimented with the creation of edges for each observed user-item pair, but observed worse performance.

4.2 A POSTERIOR OVER CONTROLLABLE DISTRIBUTIONS FROM TEXT

In section 4.1, we estimate the distribution of Z_g from a user-item bipartite graph. The next step is to optimise the objective of Eq. 4 to infer the latent factors of users and items, $z^{(u)}$ and $z^{(i)}$ in text modality. In this section, we perform the following two steps:(1) minimising the lower bound of O_{IB} by a RBF-kernel estimated posterior probability; and (2) a prior-centralisation term which encourages the encoder weight matrix in the variational network to be closer to the cluster centroid representations learned in the interaction graph, so that the learnt latent variables naturally capture the cluster-based distances.

Optimising Conditional Probability by Kernel Density For the $Q(x_n)$ in Eq. (4) which is defined as the conditional probability of $\log q(\bar{x}_n|f(x_n,\epsilon))$, we use the Nadaraya-Watson estimator (Hall et al., 1999):

$$Q(x_n) = \log\frac{\hat{p}(\bar{x}_{n,\epsilon}, x_n)}{\hat{p}(x_n)} = \log\frac{\frac{1}{N}\sum_{j=1}^{N}\kappa(\frac{x_n - x_j}{h})\cdot\kappa(\frac{\bar{x}_{n,\epsilon}-x_j}{h})}{\frac{1}{N}\sum_{j=1}^{N}\kappa(\frac{x_n - x_j}{h})}, \tag{6}$$

where $\kappa(\cdot)$ is a kernel function for density estimation, $\bar{x}_{n,\epsilon}$ is the reconstructed x_n with random noise ϵ. In practice, we choose the RBF kernel $\kappa = \exp(-||x - x'||^2))$ and ignore the denominator since the goal is to optimise the decoding of $\bar{x}_n$, then we are able to obtain the following bound by the triangle inequality in the learned metric space as $Q(x_n) \propto \log(\exp(-||\bar{x}_{n,\epsilon} - x_n||^2))$. To simplify the computation, we choose natural logarithm function in optimisation and obtain a mean squared error based objective function O_{IB}[4]:

$$O_{IB} = \frac{1}{N}\Sigma_{n=1}^{N}\mathbb{E}_{\epsilon\sim p(\epsilon)}[||\bar{x}_{n,\epsilon} - x_n||^2] + \beta\cdot\mathrm{KL}(p(Z_t|x_n,\epsilon)|r(Z_g)), \tag{7}$$

where $x_n \in \{x^{(u)}\}\bigcup\{x^{(i)}\}$ denote the input user and item representations, and $\bar{x}_{n,\epsilon}$ denote the reconstruction. By minimising the loss, we can preserve the key local similarity of nearby representations while ensuring different representations from different feature characteristics (Czolbe et al., 2020; Wang et al., 2004). Note that we stop the gradient back-propagation of the input x_n and optimise the encoder-decoder parameters only, because the updating of input representations might reduce the relevance between modalities, break the assumption of $I(X_t, X_g) \geq I_x$, and lead to two independent VAEs as a degeneracy of transferable information bottleneck given by Eq. 2.

Minimising Distribution Discrepancy To minimise the second term of KL-divergence in O_{IB}, we need to project Z_g and Z_t to a metric space with the same size. Inspired by (Van der Maaten & Hinton, 2008), a Student t-distribution is used to mitigate the crowding problem. Here, we use the softmax activation with temperature τ to adjust the distribution tail and map its values to the range of $[0, 1]$. The k-th dimension of a latent variable z_{nk}[5] denotes the probability of the corresponding user or item being assigned to the k-th cluster:

$$\eta_{nk} = \frac{1 + e^{\varphi(z_{nk})/\tau}}{\sum_{k'=1}^{K} e^{\varphi(z_{nk'})/\tau}}, \quad \mathrm{KL}(p(Z_t|x_n,\epsilon)|r(Z_g)) = \sum_{n}\sum_{k=1}^{K}\eta_{nk}\log\frac{\eta_{nk}}{\rho_{nk}}, \tag{8}$$

where $\varphi(z_n)$ is a linear transformation of z_n.Here we minimise the KL-divergence between ρ_{jk} (in the user-item interaction graph space) and η_{jk}. Therefore, we rewrite the O_{IB} as:

$$O_{IB} = \frac{1}{N}\Sigma_{n=1}^{N}\mathbb{E}_{\epsilon\sim p(\epsilon)}[||\bar{x}_{n,\epsilon} - x_n||^2] + \beta\cdot\sum_{n}\sum_{k=1}^{K}\eta_{nk}\log\frac{\eta_{nk}}{\rho_{nk}}, \tag{9}$$

Prior-centralisation: Linking Clusters with Encoder Weights In the encoding progress, we obtain the latent variable $z_n = f(x_n, \epsilon)$ by applying the reparameterisation trick (Kingma & Welling, 2013). However, due to the randomness in initialisation, the learned distribution of z_n might be chaotic. Therefore, in our one-layer MLP based encoder, we propose a prior-centralisation trick

[4]The detailed discussion can be found in Appendix C.

[5]Here, n can either be a user index u or or an item index i.

which links the pre-trained clusters to encoder by approximating the encoder weights to the cluster centroids $\boldsymbol{C}^{d\times K}$ obtained on the graph:

$$\mathcal{R}_{centroid} = \sum_{k=1}^{K} \left\| W_{en}^{k} - \boldsymbol{C}_k \right\|^2 \tag{10}$$

where W_{en}^{k} denotes the k-th column of the encoder weight matrix W_{en} [6] and $\boldsymbol{C}_k$ denotes the representation of the k-th cluster centroid vector. Note that the $R_{centroid}$ will force the distribution of z_n approximate to the prior and ignore the input. Therefore, in practise, this regularisation only deployed in several beginning epochs.

Rating Prediction and Final Objective Function After obtaining the regularised latent contextual representations $\boldsymbol{z}^{(u)}$ and $\boldsymbol{z}^{(i)}$, we add their corresponding ID features to obtain the final user and item embeddings ζ_u and ζ_i for rating prediction, $\zeta_u = W_u\boldsymbol{z}^{(u)} + \epsilon_u$, $\zeta_i = W_i\boldsymbol{z}^{(u)} + \epsilon_i$, where the ID features ϵ_u and ϵ_i are generated by feeding the user ID and item ID to an embedding layer. Inspired by the latent factor model in recommendation systems, we introduce the global biases for users and items in the final prediction layer as $\hat{r}_{ui} = \zeta_u^T\zeta_i + b_u + b_i$, where b_u and b_i are bias derived from ϵ_u and ϵ_i. The regression loss $\mathcal{L}_r$ of the predicted rating is calculated as the MSE on the given user-item pair. Combining all the components above, we derive the training objective as follows:

$$\mathcal{L} = \mathcal{L}_r + O_{IB} + \mathcal{R}_{centroid} \tag{11}$$

5 Experimental Setup

Datasets and Metrics The evaluation datasets include *BeerAdvocate* (McAuley et al., 2012) and two amazon review datasets, *Digital Music* and *Office Products* (He & McAuley, 2016) [7]. We calculate Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to evaluate the rating prediction accuracy.

Baselines We compare with several open-source recommenders, including *HFT* (McAuley & Leskovec, 2013), *DeepCoNN* (Zheng et al., 2017), *NAREE* (Chen et al., 2018). *HFT* combines reviews with ratings and uses an exponential transformation function to link review text and the ratings. *DeepCoNN* uses a shared layer for interaction modeling the users and items, which is on top of the two encoders for the users and items, respectively. *NAREE* introduces review-level attentions to select important reviews and incorporates the user and item IDs as discriminative features in rating prediction. Besides, we apply an encoder-decoder *AutoEncoder* on top of *NAREE* to learn the latent variable as a baseline to highlight the difference in our proposed regularisation.

Training Procedure We train the model parameters by minimising the objective function defined in Eq. (11). The β for the KL divergence term is set to be 0.01 and we introduce the L_2 regularisation for all the model parameters and the weight is 0.001. [8]. We only train the prior-centralisation term for 0.5 proportion to approximate the cluster centroids for the later training. The KL term in O_{IB} should be introduced later until the encoder centroid is well trained. To do so, we follow a similar cyclical schedule (Fu et al., 2019) to gradually adjust the anneal factor λ in each epoch. Specifically, we first train the model without the KL term for 0.5 proportion, then anneal it from 0.5 to 1 for 0.25 proportion, and finally fix $\lambda = 1$.

6 Experimental Results

We first show that the proposed model performs better in rating prediction compared to existing CNN-based recommenders. We then examine the contribution of each training objective and the geometric prior. We finally show that the learnt latent variables are disentangled, semantic coherent and informative, which benefit the explainable recommendation system.

[6] We also experiment with minimising the distance between decoder weight and the cluster centroid, as the encoder and decoder are asymmetrical.

[7] The dataset statistics and pre-processing details can be found in Appendix A.1.

[8] The hyper-parameter settings are described in Appendix A.2.

Models	*BeerAdvocate*		*Digital Music*		*Office Products*	
	RMSE(↓)	MAE(↓)	RMSE(↓)	MAE(↓)	RMSE(↓)	MAE(↓)
HFT	79.81	62.42	96.42	74.76	89.46	68.57
DeepCoNN	77.28	59.45	94.69	71.03	85.14	64.82
NAREE	76.80	58.94	93.69	69.30	84.40	63.40
AutoEncoder	75.94	58.93	93.97	69.13	85.03	64.32
Ours	**75.36***	**57.87***	**92.87***	**68.68***	**84.32**	**62.05***

Table 1: Performance comparison for all methods. * denotes the statistical significance for $p < 0.01$, compared to the best CNN-encoder based baseline.

6.1 Rating Prediction Results

The rating prediction results of our model in comparison with baselines on all datasets are given in Table 1. We have the following observations. (1) *DeepCoNN* built on the stack of non-linear neural networks for review semantic modeling outperforms *HFT* which leverages an exponential transformation function to link topic distributions in review text and latent factors derived from ratings. This shows the superiority of deep learning for feature extraction. (2) *AutoEncoder* gives slightly better performance than *NAREE* in *BeerAdvocate*, which shows its effectiveness of extracting key contextual information for rating prediction. (3) Our method consistently outperforms all the baselines and the improvement is more predominant on the *BeerAdvocate*, which has the smallest sparsity. The results demonstrate the effectiveness of our proposed information bottleneck regularisation applied on the latent semantic space. By doing this, the users and items can be grouped into different clusters according to the interaction data, which is not the case in other textual CNN-based recommenders.

6.2 Performance contributions from various modules

Effects of different loss terms We study the effects of the three loss terms in rating prediction accuracy and *Diversity* of the latent variables. As variational networks could easily collapse into an unconditional generative model, i.e., in the extreme case, all the input will be mapped into the same latent code (Ma et al., 2019b). We use the dimension index whose corresponding latent value is the maximum as the cluster assignment ID. We then derive the cluster assignment results $\mathcal{A} \in \mathbb{R}^{N \times K}$, where N is the number of test samples, K is the cluster number. The diversity is calculated based on entropy $H(X) = -\sum_{k \in K} p(k) log(p(k)))$, where $p(k)$ is the fraction of the number of samples falling into the k-th cluster among all the samples. A larger value means a better diversity. We record the larger diversity between user and item latent variables. The results are shown in Table 2.

Variants	*BeerAdvocate*		*Digital Music*		*Office Products*	
	RMSE(↓)	Div(↑)	RMSE(↓)	Div(↑)	RMSE(↓)	Div(↑)
Full Model	75.36	1.89	92.87	3.46	84.32	1.58
-w/o $\mathcal{R}_{centroid}$	75.96	0.04	92.62	0.07	84.58	0.02
-w/o KL term	75.64	1.43	93.83	2.86	84.23	1.21
-w/o $Q(x_n)$	76.02	1.04	93.95	3.10	84.94	0.93
$Q(x_n)$ with Cosine	75.35	0.11	92.85	1.76	84.97	0.07

Table 2: *RMSE* and *Diversity* among model variants.

We firstly remove one of the three losses, i.e., $\mathcal{R}_{centroid}$, KL term and $Q(x_n)$ to study their contributions. We observe that the overall model performance drops, with the removal of $Q(x_n)$ leading to the largest performance degradation. After removing $\mathcal{R}_{centroid}$, the latent variable diversity shrinks to near zero, indicating nearly all the latent variables fall into the same cluster. This highlights the capability of our prior-centralisation term in enabling the latent variables z_n effectively reflect the soft cluster assignments. We also found that replacing the MSE in $Q(x_n)$ with the cosine similarity $(1 - cosine)$ reduces the latent variable diversity significantly. This can be partly explained by the fact that the cosine similarity focuses on measuring the angle between the input x_n and the reconstructed output $\bar{x}_n$, ignoring the magnitude of the vectors which is however important in our case.

Effect of our proposed information bottleneck regularisation We compare our approach with *StandPrior*, *WassersteinVAE* and *IndivPrior*:
StandPrior (aka. *StandVAE*) uses two VAEs for users and items respectively, each with the standard Gaussian distribution $\mathcal{N}(0, 1)$ as the prior.
WassersteinVAE differs from the *StandPrior* in using the Wasserstein metric (Tolstikhin et al., 2017) to calculate the distribution discrepancy between posterior η and prior ρ.

IndivPrior assigns a separate Gaussian prior to each user or item with its mean value calculated from the user's or item's corresponding representation derived from the user-item interaction graph, i.e., $p(z^{(u)}) = \mathcal{N}(W_u \xi u, \boldsymbol{I})$, $p(z^{(i)}) = \mathcal{N}(W_i \xi i, \boldsymbol{I})$, where $\xi u \in \mathbb{R}^d$, $\xi i \in \mathbb{R}^d$ denote the user and item node representations in GCN learning, W_u and W_i are learnable parameters in a linear layer of size $(d \times d)$, and d is the dimension of graph node features.

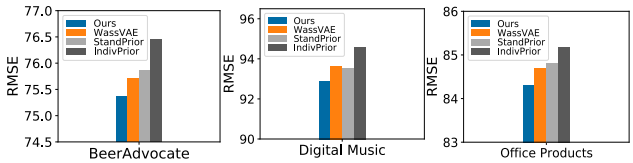

Figure 2: *RMSE* with different information bottleneck regularisation. Our method achieves the best, followed by *StandPrior* and *WassersteinVAE*. *IndivPrior* is the worst, even worse than *AutoEncoder*.

The rating prediction results are shown in Figure 2. *WassersteinVAE* and *StandPrior* demonstrate overall better results than *IndivPrior* and all the baselines in Table 1. The improvement can be explained by the nature of VAE, which is particularly beneficial when dealing with sparse data where few observations are available. However, *IndivPrior*, which imposes a separate Gaussian prior on individual user/item, shows worse performance compared to the other three variational frameworks, and is even worse than *AutoEncoder*. Our model benefits from a rich prior separated from different user/item clusters, while avoiding using a global normal prior (*StandPrior* and *WassersteinVAE*) or a separate instance-level prior for each user/item (*IndivPrior*).

6.3 Latent Variable Interpretability Evaluation

Cluster Separability and Coherence As each dimension of the latent variables z_n corresponds to a cluster, for the k-th dimension, we can search for its relevant reviews which have the highest value in z_{nk} and list the most frequent words in the review set as the representative topic words [9]. Representative words in randomly selected three clusters on *BeerAdvocate Dataset* are shown in Table 3. For our model, the most prominent words are different across different clusters (thus being colored), i.e., '*brown roasted*', '*pine, caramel*' and '*good flavour*'. We fail to see a clear topic pattern in *StandPrior* as the top words largely overlap in different clusters. The first several words are all '*hop*', '*malt*' and '*good*'. The results show that with the incorporation of the priors derived from user/item clusters, our proposed approach is able to learn latent variables in the review semantic space which can produce better separable topic clusters.

GIANT	***StandPrior***
pour, color, **brown**, feel, **roasted**, malts, almost, moderate, coffee, dark	hop, good, malt, aroma, **hint**, pour, **great**, well, mouthfeel, much
pine, **caramel**, lacing, **citrus**, mouthfeel, hint, ipa, note, body, strong	hop, aroma, good, sweet, light, malt, finish, **brew**, poured, **glass**
good, **flavour**, dark, **better**, brew, style, bad, Canadian, nose, drinking	hop, malt, light, sweet, much, **carbonation**, good, well, mouthfeel, pour

Table 3: The most prominent words (sorted by occurrence frequency) in three randomly selected clusters from **GIANT** and *StandPrior*. We highlight the top 3 words that are not found in the other two clusters.

Models	***BeerAdvocate***		***Digital Music***		***Office Products***	
	User	Item	User	Item	User	Item
Graph	0.092	0.085	0.155	0.214	0.245	0.200
GIANT	0.490	0.481	0.457	0.471	0.492	0.407

Table 4: Average cosine similarity of review pairs within each cluster as the coherence measure. A larger similarity value means a better coherence. **GIANT** generates significantly more coherent clusters than graph clusters.

We explore the difference between our generated clusters z_{nk} and the clusters derived in the graph. To verify the capability of creating *semantically coherent clusters*, we propose to measure the cluster coherence as the average cosine similarity between every review document pair within a cluster. We first obtain the document-level review representations by feeding the reviews from the test set to our pre-trained CNN encoder. We then calculate the average cosine similarity between the representations of each review pair in a cluster, and are further averaged across all clusters (in Table 4).

[9] For better visualization, we exclude the stopwords and most frequent words appeared in all clusters, such as *'beer'* which occurs in every cluster in the *BeerAdvocate* dataset. The same post-processing is performed for comparison methods. The results of *Digital Music* and *Office Products* are shown in Appendix D.1.

Comprehensiveness Evaluation by perturbing on latent variables To explore the faithfulness of our generated explanations, i.e., the identified predominant cluster in the latent dimension, we calculate the performance change before and after removing the specific dimension and define *Comprehensiveness* as: $\text{Comprehensiveness} = \frac{\sum_i^N \left(r(z_i) - r(z_i/e_i)\right)^2}{N}$, Where $r(\cdot)$ is the predicted rating, z_i is the latent variable for i-th evaluated user-item pair, e_i is the generated explanations, i.e., the identified top k latent dimensions. To delete the associated indicative features, we replace the values in the top k dimensions with the average value of the latent variables according to Fong & Vedaldi (2017). In the Figure 3 line chart, our model demonstrates a larger performance change as more important dimensions are removed and its changes are more obvious than the *StandVAE*. To avoid the effects from performance fluctuation caused by the model itself, we randomly delete k latent dimensions and calculate the relative performance change by subtracting the changes caused by random removal (Table below Figure 3). Some relative changes of *StandVAE* are negative, showing that the removal of random latent dimensions can bring even larger changes to the predominant ones.

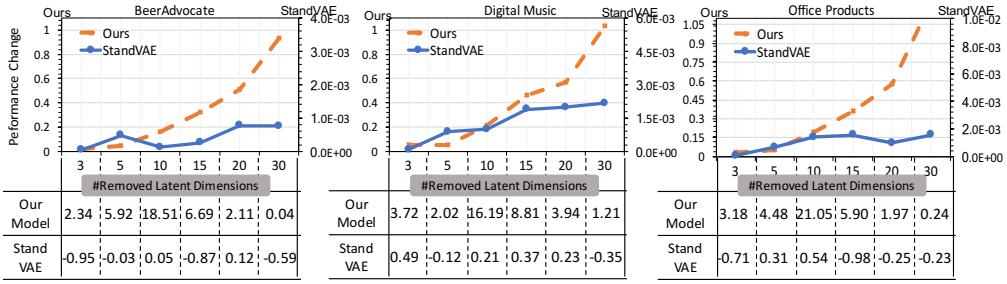

Figure 3: **Top**: The *Comprehensiveness* values by removing the top k most important latent dimensions identified, $k \in \{3, 5, 10, 15, 20, 30\}$. **Bottom Table**: Relative performance changes after subtracting the changes caused by randomly removing k latent dimensions.

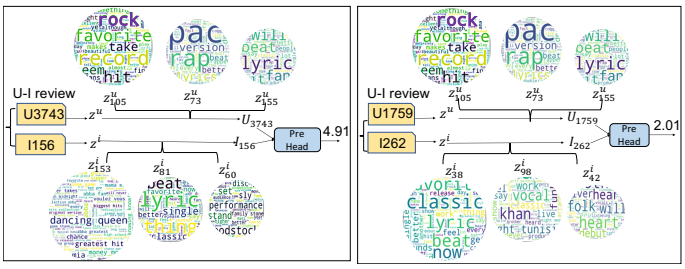

Figure 4: Two user-item review pairs are fed into our model to derive the latent variables for rating prediction. We identify the top 3 prominent clusters which has the larger latent values (illustrated by size) to show the fine-grained explanations.

Fine-grained Interpretations We illustrate how our approach could explain the recommender behaviour by providing fine-grained interpretations in Figure 4. The two users are mapped into the same top 3 clusters ($z_{105}^{(u)}, z_{73}^{(u)}, z_{155}^{(u)}$), showing interests about the '*rock*' music, '*rap*' and '*lyrics*', respectively. The first item (I156) is mapped to three clusters with main features about the rock group '*ABBA*' and another features in '*lyrics*', '*performance'*, while the second item (I262) is assigned to '*classic lyric*', '*classic vocal*' and '*folk*', respectively. There is a good match between the user primary interest (*rock music*) and the item main characteristics (*ABBA band*) and hence explains a higher rating of 4.91. Our model outputs also explain that the user 'U1759', who is also featured as a *rock music* fan, expresses little interests (2.01 rating score) in item 'I262' about *classic style music*.

7 Conclusion

In this paper, we leverage the user/item clusters sharing common interests/characteristics obtained on the interaction graph to refine the review text latent factor via our proposed geometric information bottleneck (**GIANT**). We use the information bottleneck theory to elaborate its transferability and feasibility across modalities and design a variational framework to impose the prior derived from the graph to review latent space. We empirically show that **GIANT** is better in learning a semantically coherent and interpretable latent space and the generated explanations are faithful to the model, while achieving comparable rating prediction accuracy on three commonly used datasets.

REFERENCES

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HyxQzBceg.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003. URL http://jmlr.org/papers/v3/blei03a.html.

Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (eds.), *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pp. 1583–1592. ACM, 2018. doi: 10.1145/3178876.3186070. URL https://doi.org/10.1145/3178876.3186070.

Tong Chen, Hongzhi Yin, Guanhua Ye, Zi Huang, Yang Wang, and Meng Wang. Try this instead: Personalized and interpretable substitute recommendation. In Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (eds.), *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pp. 891–900. ACM, 2020. doi: 10.1145/3397271.3401042. URL https://doi.org/10.1145/3397271.3401042.

Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. Learning to rank features for recommendation over multiple categories. In Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel (eds.), *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pp. 305–314. ACM, 2016. doi: 10.1145/2911451.2911549. URL https://doi.org/10.1145/2911451.2911549.

Zhongxia Chen, Xiting Wang, Xing Xie, Tong Wu, Guoqing Bu, Yining Wang, and Enhong Chen. Co-attentive multi-task learning for explainable recommendation. In Sarit Kraus (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 2137–2143. ijcai.org, 2019. doi: 10.24963/ijcai.2019/296. URL https://doi.org/10.24963/ijcai.2019/296.

Steffen Czolbe, Oswin Krause, Ingemar J. Cox, and Christian Igel. A loss function for generative neural networks based on watson's perceptual model. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/165a59f7cf3b5c4396ba65953d679f17-Abstract.html.

Persi Diaconis and David Freedman. A dozen de Finetti-style results in search of a theory. *Annales de l'I.H.P. Probabilités et statistiques*, 23(S2):397–423, 1987. URL http://www.numdam.org/item/AIHPB_1987__23_S2_397_0/.

Hao Fei, Yafeng Ren, Shengqiong Wu, Bobo Li, and Donghong Ji. Latent target-opinion as prior for document-level sentiment classification: A variational approach from fine-grained perspective. In Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (eds.), *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pp. 553–564. ACM / IW3C2, 2021. doi: 10.1145/3442381.3449789. URL https://doi.org/10.1145/3442381.3449789.

Ruth Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *CoRR*, abs/1704.03296, 2017. URL http://arxiv.org/abs/1704.03296.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*

NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pp. 240–250. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1021. URL https://doi.org/10.18653/v1/n19-1021.

Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. Prince: Provider-side interpretability with counterfactual explanations in recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 196–204, 2020.

Xinyu Guan, Zhiyong Cheng, Xiangnan He, Yongfeng Zhang, Zhibo Zhu, Qinke Peng, and Tat-Seng Chua. Attentive aspect modeling for review-aware recommendation. *CoRR*, abs/1811.04375, 2018. URL http://arxiv.org/abs/1811.04375.

Peter Hall, Rodney C. L. Wolff, and Qiwei Yao. Methods for estimating a conditional distribution function. *Journal of the American Statistical Association*, 94(445):154–163, 1999. ISSN 01621459. URL http://www.jstor.org/stable/2669691.

Ruining He and Julian J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Jacqueline Bourdeau, Jim Hendler, Roger Nkambou, Ian Horrocks, and Ben Y. Zhao (eds.), *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pp. 507–517. ACM, 2016. doi: 10.1145/2872427.2883037. URL https://doi.org/10.1145/2872427.2883037.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (eds.), *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pp. 639–648. ACM, 2020. doi: 10.1145/3397271.3401063. URL https://doi.org/10.1145/3397271.3401063.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. URL https://arxiv.org/abs/1312.6114.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Lei Li, Yongfeng Zhang, and Li Chen. Generate neural template explanations for recommendation. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20, pp. 755–764, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3411992. URL https://doi.org/10.1145/3340531.3411992.

Lei Li, Yongfeng Zhang, and Li Chen. Personalized prompt learning for explainable recommendation. *CoRR*, abs/2202.07371, 2022. URL https://arxiv.org/abs/2202.07371.

Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. Neural rating regression with abstractive tips generation for recommendation. In Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (eds.), *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pp. 345–354. ACM, 2017. doi: 10.1145/3077136.3080822. URL https://doi.org/10.1145/3077136.3080822.

Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. Disentangled graph convolutional networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4212–4221. PMLR, 2019a. URL http://proceedings.mlr.press/v97/ma19a.html.

Xuezhe Ma, Chunting Zhou, and Eduard H. Hovy. MAE: mutual posterior-divergence regularization for variational autoencoders. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019b. URL https://openreview.net/forum?id=Hke4l2AcKQ.

Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pp. 165–172, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450324090. doi: 10.1145/2507157.2507163. URL https://doi.org/10.1145/2507157.2507163.

Julian J. McAuley, Jure Leskovec, and Dan Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In Mohammed Javeed Zaki, Arno Siebes, Jeffrey Xu Yu, Bart Goethals, Geoffrey I. Webb, and Xindong Wu (eds.), *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pp. 1020–1025. IEEE Computer Society, 2012. doi: 10.1109/ICDM.2012.110. URL https://doi.org/10.1109/ICDM.2012.110.

Jianmo Ni, Jiacheng Li, and Julian J. McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 188–197. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1018. URL https://doi.org/10.18653/v1/D19-1018.

Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. *Counterfactual Explainable Recommendation*, pp. 1784–1793. Association for Computing Machinery, New York, NY, USA, 2021a. ISBN 9781450384469. URL https://doi.org/10.1145/3459637.3482420.

Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. Counterfactual explainable recommendation. In Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (eds.), *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pp. 1784–1793. ACM, 2021b. doi: 10.1145/3459637.3482420. URL https://doi.org/10.1145/3459637.3482420.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle, 2015. URL https://arxiv.org/abs/1503.02406.

Ilya O. Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf. Wasserstein auto-encoders. *CoRR*, abs/1711.01558, 2017. URL http://arxiv.org/abs/1711.01558.

Jakub M. Tomczak and Max Welling. VAE with a vampprior. *CoRR*, abs/1705.07120, 2017. URL http://arxiv.org/abs/1705.07120.

Quoc-Tuan Truong, Aghiles Salah, and Hady W. Lauw. Bilateral variational autoencoder for collaborative filtering. In Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich (eds.), *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, pp. 292–300. ACM, 2021. doi: 10.1145/3437963.3441759. URL https://doi.org/10.1145/3437963.3441759.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In Chid Apté, Joydeep Ghosh, and Padhraic Smyth (eds.), *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pp. 448–456. ACM, 2011. doi: 10.1145/2020408.2020480. URL https://doi.org/10.1145/2020408.2020480.

Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. Explainable recommendation via multi-task learning in opinionated text data. In Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (eds.), *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pp. 165–174. ACM, 2018. doi: 10.1145/3209978.3210010. URL https://doi.org/10.1145/3209978.3210010.

Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. Disentangled graph collaborative filtering. In Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (eds.), *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pp. 1001–1010. ACM, 2020. doi: 10.1145/3397271.3401137. URL https://doi.org/10.1145/3397271.3401137.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861. URL https://doi.org/10.1109/TIP.2003.819861.

Yongfeng Zhang and Xu Chen. Explainable recommendation: A survey and new perspectives. *Found. Trends Inf. Retr.*, 14(1):1–101, 2020. doi: 10.1561/1500000066. URL https://doi.org/10.1561/1500000066.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin (eds.), *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, pp. 83–92. ACM, 2014. doi: 10.1145/2600428.2609579. URL https://doi.org/10.1145/2600428.2609579.

Yongfeng Zhang, Xu Chen, et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020.

Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun. Adversarially regularized autoencoders for generating discrete structures. *CoRR*, abs/1706.04223, 2017. URL http://arxiv.org/abs/1706.04223.

Lei Zheng, Vahid Noroozi, and Philip S. Yu. Joint deep modeling of users and items using reviews for recommendation. In Maarten de Rijke, Milad Shokouhi, Andrew Tomkins, and Min Zhang (eds.), *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, pp. 425–434. ACM, 2017. doi: 10.1145/3018661.3018665. URL https://doi.org/10.1145/3018661.3018665.

A EXPERIMENT SETUP

A.1 DATASET DETAILS AND DATA PROCESSING

We choose three commonly used e-commerce datasets which not only contain the interaction interactions between user and item but also the review texts. Amazon review data [10] is one of the popular dataset collection, consisting of 24 kinds of products. As our method derive the user/item mainly based on their reviews, we use the dense subset, 5-core, of the amazon review dataset that extract the user-item pairs, such that each of the remaining users and items have at least 5 reviews each. Considering the training efficiency and topics should be familiar to general readers, we select the two datasets with around 50k reviews, *Digital Music* and *Office Products*. Besides, we use a sparser dataset *BeerAdvocate* dataset, i.e., the lower rating of observed ratings between total interactions. Table A1 summarizes the statistics of the three datasets. We further filter out users or items with more than 100 reviews/interactions and truncate the reviews to length of [200,400,300] for the three datasets, respectively. We randomly select 80%, 10%, 10% as the train/validation/test sets.

Datasets	**# Interactions**	**# Users**	**# Items**	**# Reviews Per User**	**# Reviews Per Item**	**AvgRating**	**Sparsity**
BeerAdvocate	35450	6939	13122	56	31	3.74	0.00044
Digital Music	51768	5541	3568	20	40	4.22	0.00294
Office Products	35999	4902	2364	86	99	4.33	0.00349

Table A1: Datasets statistics. All the datasets have ratings in the range of 1-5. The *BeerAdvocate* dataset has the smallest sparsity.

A.2 HYPER-PARAMETER SETTINGS

We use Adam for optimization with the initial learning rate set to 0.001. The batch size is set to 32, 64, and 32, and training epochs are 10, 10, and 15 for the three datasets, respectively. The adjacency matrix in graph is binary: user-item entries with the interacted ratings higher than the average corpus-wide rating is set to 1, and 0 otherwise. We stop training GCN when it reaches the best recall value in the validation set. The cluster centroid vectors are derived by K-means. The graph node feature dimension d is 64, and the number of clusters K are searched in $[16, 32, 64, 128, 256, 512]$. In CNN review encoding, the word embedding and ID embedding are both initialized by the uniform distribution $(-0.1, 0.1)$, their dimensions are set to 300 and 64, respectively. We use two CNN kernels with size 2 and 3, and the dimension of 32 to encode the reviews. As such, the CNN-encoder outputs the review representations with dimension of 64 (i.e., 32×2). To combine multiple reviews for each user or item, we use the attention mechanism where the attention weights are derived through two consecutive linear transformation layers of $64 \rightarrow 32 \rightarrow 1$. In the **GIANT**, the latent variable dimension is set to the same as the number of clusters K. The softmax temperature τ is searched in $[1, 2, 4, 6, 8]$. The non-linear activation function in both encoder and decoder is ReLU. We use a linear layer with $[K, 64]$ weights to map the the latent variable z_n to the same space of the ID features, and the linear rating prediction head is $[64, 1]$.

B TRANSFERABLE INFORMATION BOTTLENECK

Suppose we use independent encoder-decoder based architectures to model the features for different modalities (e.g., graph and text), then a given input x has representations $x_t \in X_t$ and $x_g \in X_g$ under different modalities with the constraint $I(X_t, X_g) \geq I_x$ since X_t and X_g which are from the same input, should be relevant. Also, assuming $I(Z_t, X_t) \leq I_c$ and $I(Z_g, X_g) \leq I_c$, then the following property holds:

$$I(X_t, Z_g) \leq H(X_t) - H(X_g) + H(X_t|X_g) + I_c$$

Proof: We first apply the chain rule twice for three sources X_t, X_g and Z_g, it can be written:

$$H(X_g|X_t) = H(Z_g|X_t) + H(X_g|X_t, Z_g) - H(Z_g|X_t, X_g)$$

[10] `https://jmcauley.ucsd.edu/data/amazon/`

Thus following inequality must be true:

$$\begin{aligned} H(X_t|Z_g) &\geq H(X_g|Z_g) - H(X_t|X_g) \\ &= H(X_g) - I(X_g, Z_g) - H(X_t|X_g) \end{aligned}$$

By applying the above equation to the mutual information between X_t and Z_g, where $I(X_t, Z_g) = H(X_t) - H(X_t|Z_g)$, we have:

$$I(X_t, Z_g) \leq H(X_t) - H(X_g) + H(X_t|X_g) + I(X_g, Z_g)$$

Here, the lowest upper bound of the transferable information bottleneck is $I(X_t, Z_g) \leq I_c$ when $X_t = X_g$. It means that the distribution of two modalities are the same and the conditional entropy of $H(X_t|X_g)$ is 0. Based on a similar proof, we can obtain another results holds by $I(X_t, Z_t)$: $I(X_t, Z_g) \leq H + I(X_t, Z_t)$ where $H = H(X_g) - H(X_t) + H(X_g|X_t)$.

C Optimising Objective of Transferable Information Bottleneck

Information Bottleneck: Recall that the IB objective in this work has the form $I(Z_t, \bar{X}_t) - \beta \cdot I(Z_g, X_t)$, where

$$I(Z_t, \bar{X}_t) = \int p(\bar{x}_t, z_t) \cdot \log \frac{p(\bar{x}_t|z_t)}{p(\bar{x}_t)} d\bar{x}_t dz_t,$$

where $p(\cdot)$ is the true distribution which is not observed. According to the proof given by (Alemi et al., 2017), we consider the posterior estimation $q(\bar{x}_t|z_t)$ in the decoder. According to Gibbs' inequality, we have:

$$\begin{aligned} I(Z_t, \bar{X}_t) &\geq \int p(\bar{x}_t, z_t) \cdot \log \frac{p(\bar{x}_t|z_t)}{q(\bar{x}_t)} d\bar{x}_t dz_t \\ &= \int p(\bar{x}_t, z_t) \cdot \log p(\bar{x}_t|z_t) d\bar{x}_t dz_t - H(\bar{X}_t). \end{aligned}$$

Since the entropy of the decoding results $H(\bar{X}_t)$ is independent of model optimisation, we only need to consider the posterior estimation based lower bound. According to(Alemi et al., 2017), we have the following empirical approximation by reparameterisation trick:

$$\frac{1}{N} \Sigma_{n=1}^{N} \mathbb{E}_{\epsilon \sim p(\epsilon)} [\log q(\bar{x}_n | f(x_n, \epsilon)]$$

For the Lagrange multiplied term $I(X_t, Z_g)$ which is different from the standard information bottleneck based β-VAE, we do consider the information bottleneck of $I(X_t, Z_t)$ first, because $I(X_t, Z_g) \leq H + I(X_t, Z_t)$ according to the proof given in section B. Here, we have the following bound by applying Gibbs' inequality:

$$I(X_t, Z_t) = \int p(x_t, z_t) \log p(z_t|x_t) dx_t dz_t - \int p(z_t) \log p(z_t) dz_t.$$

Since estimating the prior distribution of Z_t might be difficult, based on the definition of transferable information bottleneck, we apply the Gibbs' inequality and have:

$$I(X_t, Z_g) \leq \int p(z_t|x_t) p(x_t) \log \frac{p(z_t|x_t)}{r(z_g)} dx_t dz_t + H,$$

where $r(z_g)$ is the posterior distribution from the graph-based latent representation, and H is decided by the prior of two modalities which is independent of model optimisation. By combining the two results, the above bound can be approximated by the reparameterisation trick (Kingma & Welling, 2013) with a Gaussian random variable ϵ:

$$O_{IB} = \Sigma_{n=1}^{N} \mathbb{E}_{\epsilon \sim p(\epsilon)} [- \underbrace{\log q(\bar{x}_n | f(x_n, \epsilon)]}_{\text{Recons. term: Q}(\mathrm{x_n})} / N + \beta \cdot \underbrace{\mathrm{KL}(p(Z_t|x_n, \epsilon)|r(Z_g))}_{\mathrm{KL-div.\ term:\ KL(Z_t|Z_g)}},$$

Further discussion about the reconstruction term $Q(x_n)$

In section 4.2, we propose to use the Nadaraya-Watson estimator(Hall et al., 1999) to approximate the conditional probability $q(\bar{x}_n|f(x_n,\epsilon)]$. The idea is to insert the decoding result $\bar{x}_n$ to the space of training samples $\{x_n\}$, and take the kernel density estimation method to approximate $Q(x_n)$ by Bayesian rule:

$$Q(x_n) = \log\frac{\hat{p}(\bar{x}_{n,\epsilon}, x_n)}{\hat{p}(x_n)} = \log\frac{\frac{1}{N}\sum_{j=1}^{N}\kappa(\frac{x_n - x_j}{h})\cdot\kappa(\frac{\bar{x}_{n,\epsilon}-x_j}{h})}{\frac{1}{N}\sum_{j=1}^{N}\kappa(\frac{x_n - x_j}{h})},$$

Since the input feature for x_n is fixed, we only care about the updating of reconstruction through encoder-decoder architecture. Thus we have:

$$Q(x_n) \propto \frac{1}{N}\sum_{j=1}^{N}\kappa(\frac{x_n - x_j}{h})\cdot\kappa(\frac{\bar{x}_{n,\epsilon}-x_j}{h})$$

To simplify the above estimation, we apply the RBF-kernel and triangle inequality in estimation and have:

$$\begin{aligned}
Q(x_n) &\propto \log\frac{1}{N}\sum_{j=1}^{N}\kappa(\frac{x_n - x_j}{h})\cdot\kappa(\frac{\bar{x}_{n,\epsilon}-x_j}{h}) \\
&= \log\frac{1}{N}\sum_{j=1}^{N}\exp(-||x_n - x_j||^2)\cdot\exp(-||\bar{x}_{n,\epsilon}-x_j||^2) \\
&= \log\frac{1}{N}\sum_{j=1}^{N}\exp(-||x_n - x_j||^2 - ||\bar{x}_{n,\epsilon}-x_j||^2) \\
&\leq \log\frac{1}{N}\sum_{j=1}^{N}\exp(-||\bar{x}_{n,\epsilon}-x_n||^2) \\
&= \log[\exp(-||\bar{x}_{n,\epsilon}-x_n||^2)]
\end{aligned}$$

Thus, we are able to optimise the $Q(x_n)$ by mean square error if we choose the natural logarithm function:

$$-Q(x_n) \geq -\ln(\exp(-||\bar{x}_{n,\epsilon}-x_n||^2)) = ||\bar{x}_{n,\epsilon}-x_n||^2$$

Therefore, we can rewrite the lower boundary as well as the learning objective O_{IB} by:

$$O_{IB} = \frac{1}{N}\Sigma_{n=1}^{N}\mathbb{E}_{\epsilon\sim p(\epsilon)}[||\bar{x}_{n,\epsilon}-x_n||^2] + \beta\cdot\mathrm{KL}(p(Z_t|x_n,\epsilon)|r(Z_g)),$$

D Additional Experiment Results

D.1 Cluster Separability

As each dimension of the latent variables z_n corresponds to a cluster, for the k-th dimension, we can search for its relevant reviews which have the highest value in z_k and list the most frequent words in the review set as the representative topic words. We identify '*ABBA*' and their popular songs, computer peripherals such as '*network cable*' and '*router*' in *Digital Music* and *Office Products* datasets, respectively.

	GIANT	***StandPrior***
Digital Music	**dr dre**, **prince**, **used known**, west coast, mirrors, love, hate, westside story, jadekiss, always	great, love, best, still, well, band, lyric, make, fan, first
	bangles, **vicki**, **vocal**, place, live, hero, takes, fall, liverpool, beatles	love, great, best, band, still, well, first, even, make, lyric
	dancing queen, **take chance**, **mamma mia**, abba, greatest hit, money money, gimme gimme, super trouper, knowing knowing, gold	great, best, band, love, still, well, first, make, lyric, come
Office Products	**set** , **computer**, **wireless**, network, cable, router, edit, pencil, ink	use, work, well, binder, great, will, **easy**, **nice**, label, **ink**
	photo, **desk**, **laser**, scanner, work, scan, epson, scaning, enough, easy	use, label, product, tape, color, binder, **quality**, really, **scanner**, great
	pencil, **tape**, **ink**, pen, anyway, pen, **rubber**, clip, pretty, eraser	use, will, tape, color, great, using, work, make, well, really

Table A2: The most prominent words (sorted by occurrence frequency) in three randomly selected clusters from **GIANT** and *StandPrior* on *Digital Music* and *Office Products* datasets. For each dataset, we highlight the top 3 words that are not found in the other two clusters. **GIANT** generates better separable topics while *StandPrior* produces topics which contain largely overlapped words.