Disentangled CVAEs with Contrastive Learning for Explainable Recommendation

Linlin Wang[1]*, Zefeng Cai[1], Gerard de Melo[2], Zhu Cao[3], Liang He[1]

[1] East China Normal University
[2] Hasso Plattner Institute, University of Potsdam
[3] East China University of Science and Technology
{llwang,lhe}@cs.ecnu.edu.cn, oklen@foxmail.com, gdm@demelo.org, caozhu55@gmail.com

Abstract

Modern recommender systems are increasingly expected to provide informative explanations that enable users to understand the reason for particular recommendations. However, previous methods struggle to interpret the input IDs of user–item pairs in real-world datasets, failing to extract adequate characteristics for controllable generation. To address this issue, we propose disentangled conditional variational autoencoders (CVAEs) for explainable recommendation, which leverage disentangled latent preference factors and guide the explanation generation with the refined condition of CVAEs via a self-regularization contrastive learning loss. Extensive experiments demonstrate that our method generates high-quality explanations and achieves new state-of-the-art results in diverse domains.

Introduction

Due to the high demand for increasing users' trust, recommender systems are often expected to provide informative explanations to better reveal why particular items are selected for recommendation (Wang et al. 2018a,b; Chen et al. 2019). Real-world explanations for recommendations can be presented in various forms (Zhang and Chen 2020). In this paper, we focus on post-hoc explanations that are expressed in natural language. As depicted in Fig. 1, our task requires a model to interpret the given user ID, item ID, and rating score from recommender systems, and thereafter generate appropriate textual explanations.

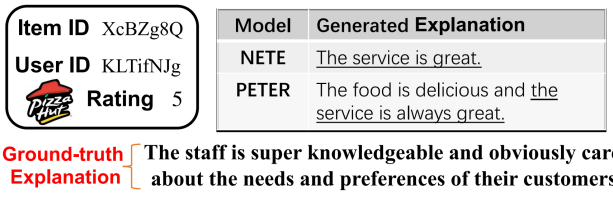

Figure 1: An example of explanation generation

However, most existing approaches tend to generate generic explanations that seldom account for the particular traits and attributes of users and items (Cao et al. 2018). An underlying reason is that previous models are insensitive to the ID strings and typically fail to extract sufficient evidence as generative signals from such opaque inputs (Li, Zhang, and Chen 2021b). This problem is particularly pronounced for neural network-based architectures that directly embed input user and item IDs in a similar way as normal words, given that such IDs occur only infrequently and are easily regarded as out-of-vocabulary tokens. Hence, previous models struggle to capture specific features behind users and items identified with IDs, instead delivering generic and often identical explanations (underlined in Fig. 1) for different user–item pairs. It can be observed that the quality of generated explanations from NETE (Li, Zhang, and Chen 2020) and PETER (Li, Zhang, and Chen 2021b), two previous state-of-the-art models, is far from satisfactory compared with the ground-truth reference.

In fact, the key to explanation generation is to recognize essential characteristics of user–item pairs (Ma et al. 2019). Yet, it is non-trivial for models to acquire better representations and thereby guarantee diversity when merely considering opaque ID strings for users and items. Conventional encoder–decoder approaches learn hidden representations from the inputs alone. To capture supplemental signals, several prior studies extend the encoder–decoder architecture by designing additional modules or introducing auxiliary tasks. For example, the NETE model relies on a neural template-based framework to incorporate feature-specific details, however at the expense of a severely restricted diversity and expressivity of the generated explanations. The PETER model exploits a Transformer-based architecture with context prediction to interpret IDs, which however entails a heavy dependency on auxiliary tasks.

A promising choice to overcome these drawbacks and ensure the informativeness of generated explanations is the use of conditional variational autoencoders (CVAEs), which leverage sampled latent variables to capture underlying semantics and further guide the learning process towards diverse generation with an extra condition (Sohn, Yan, and Lee 2015). Still, to align the CVAE-based architecture with our expectations, considerable modifications are required, particularly when the user and item IDs are opaque identifiers that are challenging to interpret. Since there is a sizeable semantic gap between input IDs and the corresponding tex-

*Corresponding author.

tual explanations during training (Hu et al. 2021), regular approaches of representing the condition of CVAEs are not applicable. Additionally, prior research shows that user behavior data in recommender systems is driven by complex latent preference factors that are highly entangled (Ma et al. 2019). These entangled factors characterize users' preferences to purchasing items and cover a large range of product attributes, each of which may correspond to different concepts separately (e.g., the size or the color of a shirt). Disentangling the underlying explanatory factors enables the model to force each dimension of the representations to independently reflect an isolated factor, which we conjecture can increase the diversity of generated explanations. However, this remains a fairly unexplored research avenue in explanation generation.

In this paper, we propose a novel disentangled CVAE-based architecture to learn better representations for explainable recommendation. It disentangles the latent variables to encourage separate dimensions to reflect pertinent user preferences for generation, and refines the condition signal of the CVAE with a self-regularization loss to more precisely interpret the inputs. Our key contributions are as follows.

- We leverage factor disentanglement to decompose the latent factors behind user preferences, enabling each isolated variable to capture informative and indispensable signals for the decoder to generate better explanations.
- We propose a novel self-regularization technique to refine the condition signal of CVAEs, which leverages a word identity loss to explore discriminative representations for user–item pairs identified with uninformative IDs, and captures essential characteristics of recommendation to guide the generative process for explanations.
- Based on the disentangled latent variables and refined condition signal, our model successfully reconstructs the given information and generates high-quality explanations, yielding state-of-the-art results on real-world datasets with long-tail users in various domains.

Related Work

Explainable Recommendation One prominent way of enabling explainable recommendation is to generate post-hoc explanations, which typically are free-text explanations (Tintarev 2007). In this setting, most prior work can be categorized as based either on templates (Zhang et al. 2014) or on neural natural language generation. However, several typical instances of the latter, e.g., NRT (Li et al. 2017) and Att2Seq (Dong et al. 2017), frequently produce generic and insufficiently diverse explanations (Zhang and Chen 2020). To address this problem, the subsequent model NETE combines template-based and neural generation methods (Li, Zhang, and Chen 2020). Nevertheless, the expressivity of generated explanations still remains far from satisfactory, since the simple combination only causes the model to fit the given samples instead of crafting new sentences (Li, Zhang, and Chen 2021a). The recent PETER model (Li, Zhang, and Chen 2021b) achieves strong results, but at the expense of relying heavily on the auxiliary task of context prediction.

Conditional Variational Autoencoders Conditional variational autoencoders (CVAEs) introduce a latent variable to capture the underlying semantics behind the data, and can further guide the target textual generation with an extra condition signal (Sohn, Yan, and Lee 2015). Previous studies have adopted CVAEs to improve the generation diversity for a range of different tasks, such as dialogue generation (Song et al. 2019). However, our post-hoc explanation generation requires the system to interpret special input ID strings that refer to particular users and items, which is a notable difference compared with other generation tasks. Hence, it is insufficient to leverage vanilla encoder–decoder approaches to extract characteristics for user–item pairs, because these models merely generate explanations with the hidden representations learnt from the inputs alone. A CVAE-based architecture is a promising choice (Cai et al. 2022) to learn better representations for user–item pairs based on uninformative IDs, because CVAEs can utilize sampled latent variables to increase the generation diversity and control the learning process via an extra condition signal. Still, to align CVAEs with our expectations, considerable architectural modifications are required that will be explained in the following.

Disentangled Representation Learning Learning disentangled representations that uncover the underlying factors has shown to improve the robustness and controllability of variational autoencoders (VAEs) (Bengio, Courville, and Vincent 2013; Dittadi et al. 2021). Since user behavior data are driven by highly entangled latent preference factors, we design a specific disentanglement technique to uncover the underlying factors for explainable recommendation. As each dimension of the disentangled representations is encouraged to independently reflect an isolated factor (Ma et al. 2019), this also guarantees an explainable generative process.

Contrastive Representation Learning Contrastive learning (He et al. 2020) brings distinguishable representations to boost the performance of models in various tasks. Previous work incorporates this technique into a VAE-based framework to extract salient features for better generation (Aneja et al. 2021). In contrast, our approach aims to learn better representations to refine the condition signal of a CVAE, which prior work seldom considers. Note that we do not use the conventional contrastive loss, but rather a custom form.

Proposed Model

Since it is non-trivial to interpret user–item IDs for informative explanation generation, we propose a novel disentangled CVAE-based architecture to learn pertinent characteristics of users and items. As illustrated in Fig. 2, the variational neural encoder first leverages both prior and recognition networks to deal with ID–rating signals and explanations, respectively. Subsequently, we draw on factor disentanglement to decompose the latent variables that are conditioned on the input IDs, and exploit self-regularization to refine the condition signal of CVAEs. Finally, our variational decoder accomplishes the reconstruction using these enhanced signals to generate informative explanations.

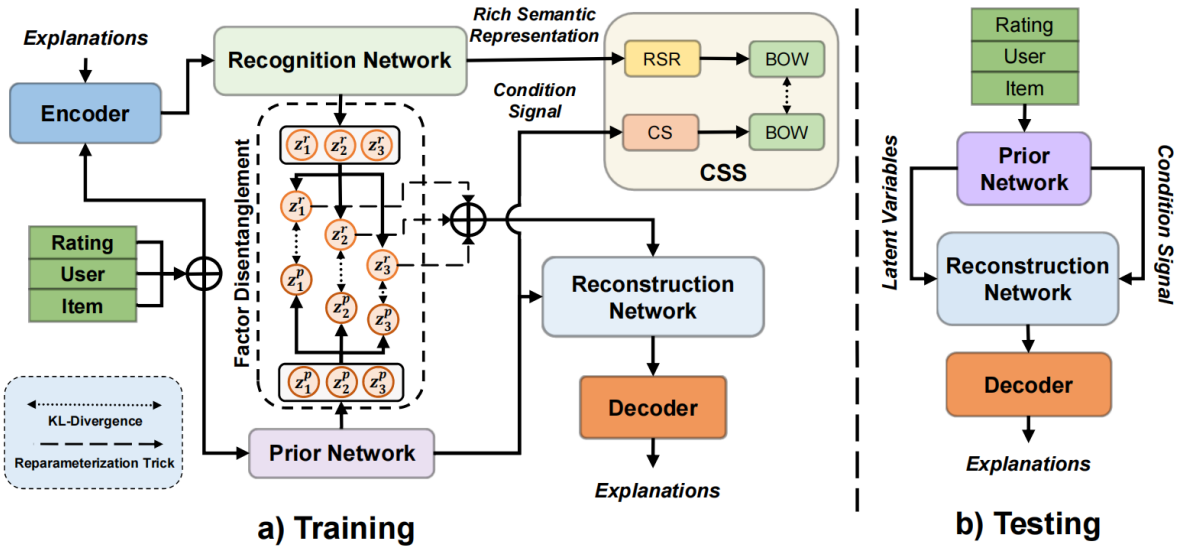

Figure 2: Overview of the Proposed Model.

Explanation Generation Formulation

We assume that the explanation generation process, i.e., $p_{d,T,\Omega}(x|y)$, is guided by a latent variable z together with the input y, where y refers to the given IDs and rating, and x is the explanation. Hence, we use the following equations to formulate this process.

$$p_{d,T,\Omega}(x|y) = \int_z p_d(x|z,y)\, p_{T,\Omega}(z|y) d_z, \quad (1)$$

$$p_d(x|z,y) = \delta(d(z,y) - x), \quad (2)$$

$$p_{T,\Omega}(z|y) = \prod_i t(z^{(i)}) g(y^{(i)}) \exp[T(z^{(i)})^\top \Omega(y^{(i)})]. \quad (3)$$

Here, we use $p_{T,\Omega}(z|y)$ to represent the prior network and $p_d(x|z,y)$ as the response decoder. Eq. (2) defines our reconstruction network with a Dirac distribution, where $d(z,y)$ is an approximate injective function. Eq. (3) defines an exponential conditionally factorial distribution (Bishop 2006) used in our prior network, where $t(\cdot)$ is the base measure, $g(\cdot)$ obtains the normalizing constant, $T(\cdot)$ refers to sufficient statistics, $\Omega(\cdot)$ obtains parameters, and $z^{(i)}$ represents the i-th disentangled latent variable.

Transformation Module

For representation transformation, we define a transformation component **TB** that employs spectral normalization to improve the robustness of the model for input disturbance in the recognition network, prior network, and reconstruction network. Let $f_{d_x,d_y}(x) = \mathrm{SN}(W_{d_y \times d_x}\mathrm{GELU}(x) + b_{d_y})$, where d_x and d_y are the input and output dimensions of this function, $\mathrm{SN}(\cdot)$ is spectral normalization, $\mathrm{GELU}(\cdot)$ is an activation function, $W_{d_y \times d_x} \in \mathbb{R}^{d_y \times d_x}$, and $b_{d_y} \in \mathbb{R}^{d_y}$. Let

$$\mathbf{TB}_{d_1,d_2,d_3}(x) = f_{d_1,d_3}(\mathrm{LayerNorm}(T_{d_1,d_2}(x) + x)) \quad (4)$$

$$T_{d_1,d_2}(x) = f_{d_1,d_2} \circ f_{d_2,d_2} \circ f_{d_2,d_1} \circ f_{d_1,d_1}(x). \quad (5)$$

Note that a $\mathbf{TB}_{d_1,d_2,d_3}(\cdot)$ component comprises five trainable transformation functions, where x is the input. d_1, d_2, and d_3 are the input, intermediate, and output dimensions of **TB**, respectively. $T_{d_1,d_2}(\cdot)$ is a composite module that consists of four different $f_{d_x,d_y}(\cdot)$ functions, where $\circ$ denotes composition. The output of **TB** is split into equal-sized partitions if the output is assigned to more than one variable.

Encoder

Recognition Network To encode all tokens in the explanation x into compact hidden states, we first employ a Transformer (Vaswani et al. 2017) that takes the first output token C as the corresponding representation. Subsequently, we leverage a recognition network shown in Fig. 3 to model the posterior $q_\theta(z|x,y)$, where θ denotes the parameters.

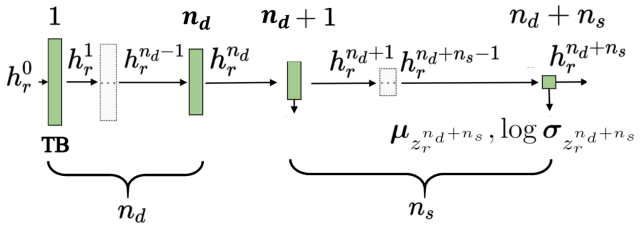

Figure 3: Illustration of the Recognition Network. Here, the first n_d-layer TB only generates the hidden states h, and the subsequent n_s-layer TB generates h, μ and σ.

As depicted in Fig. 3, the recognition network consists of $n_d + n_s$ **TB** with an initial input $\boldsymbol{h}_r^0 = \{C, e_u, e_i, e_r\}$, where e_u, e_i, and e_r are the embeddings of the given user ID, item ID, and rating. These embeddings are randomly initialized with respective embedding lookup tables. First, we feed $\boldsymbol{h}_r^0$ to the first n_d-layer **TB** for encoding, obtaining:

$$\boldsymbol{h}_r^l = \mathbf{TB}_r^l(\boldsymbol{h}_r^{l-1}), \quad (6)$$

where $l \in \{1, 2, \cdots, n_d\}$ refers to the l-th **TB** layer, and $\boldsymbol{h}_r^{n_d}$ is the output of this n_d-layer **TB**. We feed $\boldsymbol{h}_r^{n_d}$ to the subsequent n_s-layer **TB** to generate three key parameters:

$$\boldsymbol{\mu}_{z_r^j},\ \log \boldsymbol{\sigma}_{z_r^j},\ \boldsymbol{h}_r^j = \mathbf{TB}_r^j(\boldsymbol{h}_r^{j-1}), \tag{7}$$

where $j \in \{n_d+1, n_d+2, \cdots, n_d+n_s\}$ represents the j-th **TB** layer, and we consider $\boldsymbol{\mu}_{z_r^j}$ and $\boldsymbol{\sigma}_{z_r^j}$ as the mean and variance of the probability distribution $q(\boldsymbol{z}_r^j|x,y)$, respectively. Therefore, we formulate the distribution of latent variables as $q(\boldsymbol{z}_r^j|x,y) \sim \mathcal{N}(\boldsymbol{\mu}_{z_r^j}, \mathrm{diag}(\sigma_{z_r^j}))$, where $\mathrm{diag}(\cdot)$ is a function that transforms a vector to a diagonal matrix with the same dimensions. We use the reparametrization trick (Kingma and Welling 2014) to sample a latent variable $\boldsymbol{z}_r^j$. In addition, $\boldsymbol{h}_r^j$ in Eq. 7 is the output of the n_s-layer **TB**. The recognition network outputs $\boldsymbol{h}_r^{n_d+n_s}$, which corresponds to rich semantic representations depicted in Fig. 2.

Prior Network The prior network encodes the input y, comprising a user ID, item ID, and rating, which employs the same **TB** based structure as shown in Fig. 2 to model $p_{T,\Omega}(z|y)$ and contributes to the generation of the CVAE condition signal. As previously mentioned, e_u, e_i, and e_r represent the embeddings of the user ID, item ID, and rating. We first concatenate all these embeddings to obtain $\boldsymbol{h}_0^p = [e_u, e_i, e_r]$, and then feed $\boldsymbol{h}_0^p$ to the n_d-layer **TB** for further encoding. Finally, we employ the subsequent n_s-layer **TB** to generate three key parameters:

$$\boldsymbol{\mu}_{z_p^j},\ \log \boldsymbol{\sigma}_{z_p^j},\ \boldsymbol{h}_p^j = \mathbf{TB}_p^j(\boldsymbol{h}_p^{j-1}). \tag{8}$$

Here, $j \in \{n_d+1, \cdots, n_d+n_s\}$ refers to the j-th TB layer. $\boldsymbol{\mu}_{z_p^j}$ and $\boldsymbol{\sigma}_{z_p^j}$ are the mean and the variance of the probability distribution $p_{T,\Omega}(\boldsymbol{z}_p^j|y)$, respectively. The output of prior network $\boldsymbol{h}_p^{n_d+n_s}$ corresponds to the CVAE condition.

To achieve factor disentanglement, we extend Eq. (3) as

$$p_{T,\Omega}(z|y) = \prod_i t(z_r^{(i)}) g(h_p^{(i)}) \exp[T(z_r^{(i)})^\top \Omega(h_p^{(i)})], \tag{9}$$

$$g(h_p^{(i)}) = \frac{1}{\sqrt{2\pi}}, \quad t(z_r^{(i)}) = \frac{1}{\sigma_{z_p^{(i)}}}, \tag{10}$$

$$T(z_r^{(i)}) = (\bar{z}_r^{(i)} - \boldsymbol{\mu}_{z_p})^2, \quad \Omega(h_p^{(i)}) = \frac{1}{2\sigma^2_{z_p^{(i)}}} \tag{11}$$

Here, $\bar{z}_r^{(i)}$ denotes dynamic statistics that can be calculated as $\bar{z}_r^{(i)} = (1-\epsilon)\bar{z}_r^{(i)} + \epsilon z_r^{(i\prime)}$, where $z_r^{(i)}$ means the i-th factor of latent variable z_r, and $z_r^{(i\prime)}$ is sampled from the distribution based on the previous instance.

Finally, we incorporate the following KL-divergence to regularize the representation of latent variables:

$$\mathcal{L}_{\mathrm{dis}} = \mathrm{KL}(q_\theta(z|x,y) \,\|\, p_{T,\Omega}(z|y)). \tag{12}$$

For inference, the prior network replaces the recognition network to generate latent variables as shown in Fig. 2(b).

Condition Signal Self-regularization (CSS)

Our CSS technique employs self-regularization to improve the representation of the CVAE condition. Since the explanation encoded by the recognition network can be viewed as a rich semantic representation of the corresponding recommendation, we can regard the recognition network as a source of supervision to provide a useful training signal to improve the condition signal representation. However, there is no guarantee that such a representation will contain meaningful information, despite being generated by the corresponding explanation. Hence, we propose unordered target word prediction to encourage the recognition network to generate more informative representations. Simultaneously, we align the condition signal with the distribution of words predicted by the recognition network, which has been shown to be a more effective method of improving the representation compared with traditional cosine similarity based loss (Liang et al. 2021). We represent the unordered words in the target explanation as x_{bow}, i.e., a form of bag-of-words loss (Harris 1970). Let $f_p = \mathbf{MLP}_p(\boldsymbol{h}_p^{n_d+n_s})$, $f_r = \mathbf{MLP}_r(\boldsymbol{h}_r^{n_d+n_s})$. We then define:

$$\begin{aligned}
\log p(x_{\mathrm{bow}}|\boldsymbol{h}_p^{n_d+n_s}) &= \log \prod_{n=1}^{|x_{\mathrm{bow}}|} \frac{\exp\left(f_p[n]\right)}{\sum_{m=1}^{V} \exp(f_p[m])} \\
\log p(x_{\mathrm{bow}}|\boldsymbol{h}_r^{n_d+n_s}) &= \log \prod_{n=1}^{|x_{\mathrm{bow}}|} \frac{\exp(f_r[n])}{\sum_{m=1}^{V} \exp\left(f_r[m]\right)}.
\end{aligned} \tag{13}$$

where n represents the n-th word in an explanation, m is the m-th word in the vocabulary, and V is the vocabulary size.

The objective of CSS can be formulated as follows:

$$\begin{aligned}
\mathcal{L}_{\mathrm{CSS}} = \gamma_1 \log p(x_{\mathrm{bow}}|\boldsymbol{h}_p^{n_d+n_s}) + \gamma_2 \log p(x_{\mathrm{bow}}|\boldsymbol{h}_r^{n_d+n_s}) \\
+ \gamma_3 \mathrm{KL}(p(x_{\mathrm{bow}}|\boldsymbol{h}_r^{n_d+n_s}) \,\|\, p(x_{\mathrm{bow}}|\boldsymbol{h}_r^{n_d+n_s})).
\end{aligned} \tag{14}$$

Here, γ_1,γ_2, and γ_3 are the weights of different terms. The first two terms of Eq. (14) assess the identity of words to encourage $\boldsymbol{h}_p^{n_d+n_s}$ and $\boldsymbol{h}_r^{n_d+n_s}$ to contain information in the target explanation and thus improve their representations. The third term can further improve the representation of $\boldsymbol{h}_p^{n_d+n_s}$, since it provides additional regularization.

Decoder

Reconstruction Network. The reconstruction network combines disentangled latent variables and the condition signal to generate natural language explanations. Fig. 4 provides an illustration of the reconstruction network.

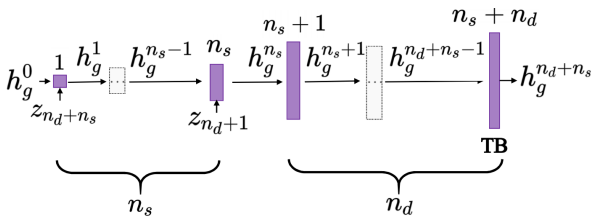

Figure 4: Illustration of the Reconstruction Network.

As depicted, we first use the condition signal $\boldsymbol{h}_p^{n_d+n_s}$ as the initial hidden state $\boldsymbol{h}_g^0$, and then conduct a two-stage decoding. Specifically, in the first stage, we add the sampled

latent variables to the hidden states, and deal with the corresponding outcome with the first n_s-layer **TB** of the reconstruction network as:

$$\boldsymbol{h}_g^j = \mathbf{TB}_g^j(\boldsymbol{h}_g^{j-1} + z^{n_d+n_s-j+1}), \tag{15}$$

where $j \in \{1, \ldots, n_s\}$ and $z^{n_d+n_s-j+1}$ is sampled from the distribution $q(\boldsymbol{z}_r^{n_d+n_s-j+1}|x, y)$ during training. Note that $z^{n_d+n_s-j+1}$ serves as the mean of $q(\boldsymbol{z}_p^{n_d+n_s-j+1}|y)$ during testing. For the second stage, we employ the subsequent n_d-layer **TB** to further decode the hidden states as

$$\boldsymbol{h}_g^c = \mathbf{TB}_g^c(\boldsymbol{h}_g^{c-1}), \tag{16}$$

where $c \in \{n_s+1, \ldots, n_s+n_d\}$ indicates the c-th **TB** layer.

Finally, the output of reconstruction network $\boldsymbol{h}_g^{n_s+n_d}$ is fed into a GPT decoder (Floridi and Chiriatti 2020) as an initial token to reconstruct the explanations. We regard the conventional negative log-likelihood $\mathcal{L}_{\text{rec}}$ as an objective term.

Training Objective

As for training, we use a reconstruction loss $\mathcal{L}_{\text{rec}}$ to optimize the decoder for explanation generation, a disentanglement loss $\mathcal{L}_{\text{dis}}$ to decompose latent variables, and our CSS loss $\mathcal{L}_{\text{CSS}}$ to refine the condition of CVAEs. Thus, the overall training objective $\mathcal{L}$ can be defined as

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \sum_{j=n_d+1}^{n_d+n_s} \alpha \mathcal{L}_{\text{dis}}^j \beta \mathcal{L}_{\text{CSS}}, \tag{17}$$

where α and β are pre-defined hyperparameters, and $\mathcal{L}_{\text{dis}}^j$ comes from the j-th **TB** layer of the prior network.

Experimental Setup

Dataset. We use three large-scale datasets including Yelp[1], Amazon 5-core Movie & TV[2] and TripAdvisor[3], and follow the common practice (Li, Zhang, and Chen 2020) to extract valid explanations and conduct pre-processing.

Metrics. We leverage several standard metrics to conduct the evaluation, including BLEU-1, BLEU-4, ROUGE-1, ROUGE-L, and METEOR. Specifically, we use the BLEU (Papineni et al. 2002) scores with 1-gram and 4-grams, respectively. ROUGE-1 refers to the ROUGE score (Lin 2004) measured with 1-grams. ROUGE-L finds the longest common subsequence and takes the sentence-level structural similarity into account. METEOR (Banerjee and Lavie 2005) accounts for synonyms in sentences, leading to a better correlation with human evaluations.

Experimental settings. We set the hidden size of our 2-layer Transformer encoder and decoder to be 768, and n_s, n_d to be 3 for **TB**. After every **TB** transformation, the input variable is compressed to half of the original size in encoding or expanded to be twice as large in reconstruction. The size of the fixed vocabulary is 20,000, and the batch size is 512. The hyper-parameters β and γ are consistently set to 1.0 and 0.8, respectively. For training, we use AdamW (Kingma and Ba 2015) with an initial learning rate 2×10^{-5}, and decrease the learning rate by a factor of 0.8 when the decrease ratio of the validation loss is smaller than 2%. We run our model five times to report average results.

Main Results

Generic Evaluation We conduct extensive experiments on three datasets for comparison with competitive baselines, including **Att2Seq**, **NETE** (Li, Zhang, and Chen 2020), and **PETER** (Li, Zhang, and Chen 2021b).[4] Table 1 provides a comparison of results on explanation generation. Overall, we achieve strong results across all three datasets, demonstrating the effectiveness of our CVAE in learning sufficient features from the input IDs. Most notably, our model obtains significant improvements compared to state-of-the-art PETER, which incorporates additional context prediction.

Explainability Evaluation For further analysis, we assess the quality of generated explanations with several newly-adopted metrics (Wen et al. 2022), including **Relevance**, **Polarity**, **Subjectivity**, and **Grammar Correctness**. For Relevance, we invoke Sentence-BERT [5] to obtain embeddings, and compute the cosine similarity between generated explanations and gold standard references. Polarity reflects the confidence levels of whether explanations are positive or negative, and Subjectivity considers the subjectivity of generated explanations. For Polarity and Subjectivity, we use TextBlob[6] to calculate the mean squared error of measured values. In addition, we use the average number of grammatical errors in generated explanations to assess Grammar Correctness using a grammar checker[7]. All explainability results are provided in Table 2. Overall, we achieve new state-of-the-art results in terms of Relevance, Polarity, and Subjectivity. The high Relevance scores suggest that our model can generate high-quality explanations that correspond well with the ground truth, while a more subjective explanation, for instance, may consist of personal opinions or judgements, which implies that our model can better capture user preferences and thereby generate more personalized explanations.

Quantitative Analysis

Ablation Study

To better quantify the contributions of different components, we conduct ablation studies with three simplified architectures. The first simplification **Ours-w/o FDis** omits Factor Disentanglement from our model. **Ours-w/o CSS** removes the loss in our CSS self-regularization, while **Ours-w/o FDis&CSS** refers to our backbone model without factor disentanglement or CSS. Table 3 provides the results of these ablations on Yelp. We observe that all aforementioned

[1] www.yelp.com/dataset

[2] www.jmcauley.uscd.edu/data/amazon

[3] www.tripadvisor.com

[4] Approaches that require extra data or additional training are omitted to keep comparisons fair.

[5] https://www.sbert.net/docs/pretrained_models.html

[6] https://textblob.readthedocs.io/en/dev/

[7] https://pypi.org/project/language-tool-python/

Model	BLEU		ROUGE-1			ROUGE-L			METEOR
	BLEU-1	BLEU-4	P	R	F1	P	R	F1	METEOR
					Yelp				
Att2Seq	11.80(±0.3)	0.82(±0.02)	14.72	11.41	12.86(±0.4)	11.03	9.36	10.13(±0.4)	4.86(±0.1)
NETE	13.93(±0.4)	1.12(±0.02)	18.33	14.57	16.23(±0.7)	13.46	12.13	12.76(±0.5)	6.43(±0.2)
PETER	16.93(±0.5)	1.24(±0.03)	20.74	16.71	18.51(±0.8)	15.82	14.21	14.97(±0.6)	6.68(±0.2)
Ours	**20.11**(±0.8)	**1.50**(±0.03)	**20.99**	**17.91**	**18.73**(±0.9)	**16.35**	**14.51**	**16.91**(±0.6)	**6.95**(±0.2)
Imp (%)	18.78	20.97	1.21	7.18	1.89	3.35	2.11	12.96	4.04
					Amazon				
Att2Seq	10.33(±0.2)	0.72(±0.02)	12.88	9.99	11.25(±0.4)	9.65	8.20	8.86(±0.4)	4.26(±0.1)
NETE	14.57(±0.4)	1.11(±0.03)	18.24	13.35	15.42(±0.9)	13.43	10.57	13.15(±0.6)	6.13(±0.1)
PETER	16.94(±0.4)	1.21(±0.05)	19.60	14.88	16.92(±1.1)	14.32	12.15	13.15(±0.6)	6.13(±0.2)
Ours	**18.51**(±0.9)	**1.42**(±0.07)	**19.73**	**15.42**	**17.31**(±1.3)	**15.21**	**12.99**	**14.01**(±0.7)	**6.51**(±0.2)
Imp (%)	9.27	17.36	0.66	3.63	2.30	6.21	6.91	6.54	6.20
					TripAdvisor				
Att2Seq	13.05(±0.2)	0.90(±0.02)	16.27	12.62	14.21(±0.7)	12.19	10.35	11.20(±0.8)	5.38(±0.2)
NETE	17.52(±0.5)	1.36(±0.03)	21.63	17.57	19.39(±1.0)	16.67	14.28	15.39(±0.9)	7.31(±0.2)
PETER	19.24(±0.8)	1.36(±0.05)	23.51	19.69	21.43(±1.2)	18.45	15.51	16.85(±1.1)	8.03(±0.3)
Ours	**22.76**(±1.4)	**1.59**(±0.08)	**25.04**	**20.13**	**22.28**(±1.6)	**20.47**	**16.36**	**18.19**(±1.3)	**8.75**(±0.3)
Imp (%)	18.30	16.91	6.51	2.23	3.97	10.95	5.48	7.95	8.97

Table 1: Generic Explanation generation evaluation, where Imp (improvements) are computed as relative gains compared with the previous strong baseline model PETER.

	Rel.	Pol.	Sub.	G.C.
Att2Seq	0.2073	0.7867	0.7653	**-0.7512**
NETE	0.2673	0.7928	0.8025	-0.7581
PETER	0.2764	0.7968	0.8258	-0.7607
Ours	**0.3221**	**0.8094**	**0.8660**	-0.7633

Table 2: Explainability evaluation on Yelp. "Rel.", "Pol.", "Sub.", "G.C." are Relevance, Polarity, Subjectivity, Grammar Correctness. Higher scores indicate better results.

model components consistently yield noticeable improvements. The removal of factor disentanglement causes an obvious performance degradation, showing that our model succeeds at disentangling latent variables behind user preferences to explore key characteristics. Likewise, the removal of CSS causes a severe degradation, demonstrating the effectiveness of our CSS in enabling CVAEs to enhance the condition signal for better explanation reconstruction.

In-Depth Analysis

We further reveal the causes of gains with two strategies.

Condition Signal Self-regularization (CSS). We use **Uniformity** and **Alignment** (Wang and Isola 2020) as metrics to evaluate the representation quality, aiming to assess whether CSS refines the condition signal of CVAEs. Uniformity reflects to what extent the embeddings of the condition are uniformly distributed, and Alignment measures the similarity of condition signals that contain the same essential information. For evaluation, we normalize $\boldsymbol{h}^p_{n_d+n_s}$ and employ Principal Components Analysis to reduce the dimensionality to 2 for visualization on Yelp. Fig. 5 (a) shows that our model maintains a more uniform embedding space compared to the variant without CSS, confirming the effectiveness of CSS in improving uniformity. In Fig. 5 (b), the two

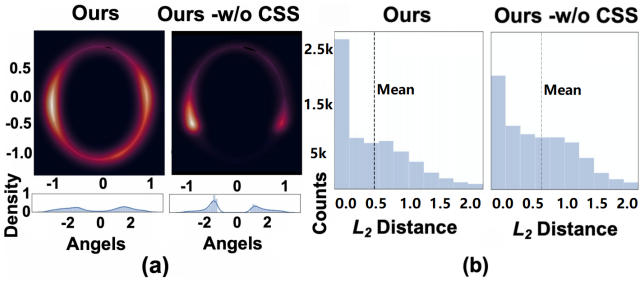

Figure 5: Uniformity (a) and Alignment (b) of the condition signals of CVAEs. (a) depicts the distributions of condition embeddings with KDE density estimation and angles calculated using the arctan2 function. (b) shows the distribution of L_2 distances between two embeddings in the positive pair.

embeddings from the same positive pairs are closer to each other, which implies that CSS results in a better alignment for the refined condition of CVAEs. In general, greater uniformity indicates that more information is preserved in condition signals, and a high alignment score suggests a higher

	BLEU-1	BLEU-4	ROUGE-1	ROUGE-L	METEOR
Ours	20.11	1.50	16.91	14.51	6.95
-w/o FDis	19.37 (↓3.68%)	1.41 (↓6.00%)	16.52 (↓2.31 %)	13.91 (↓4.14%)	6.76 (↓2.73%)
-w/o CSS	18.25 (↓9.25%)	1.27 (↓15.33%)	15.53(↓9.46%)	13.01 (↓10.34%)	6.29 (↓9.50%)
-w/o FDis&CSS	17.88 (↓11.09%)	1.24 (↓17.33%)	15.13 (↓10.53%)	12.52 (↓13.71%)	5.92 (↓14.68%)

Table 3: Ablation Study on Yelp. Relative drops are computed in comparison with Ours.

similarity between the embeddings of condition signals from an input recommendation pair and its corresponding explanations. Thus, we conclude that our model with CSS brings better refined condition signals for explanation generation.

Factor Disentanglement To assess the degree of factor disentanglement, we employ Mutual Information to evaluate the disentanglement between different groups of latent variables. We adopt avgMI to measure their relevance as

$$\mathrm{MI}(x, y) = \sum_{x_i} \sum_{y_j} [p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}], \qquad (18)$$

where x_i and y_i refer to the values in the i-th dimension of x and y, respectively. We select the groups of latent variables from the bottom layer of the model, and plot the mutual information values between them in the matrix given in Fig. 6. An optimal matrix will have a value of 1 along its diagonal and 0 elsewhere. Fig. 6 shows that our model is effective at reducing the Mutual Information between different groups of latent variables. Compared with the variant without factor disentanglement (Ours-w/o FDis), we conclude that the proposed disentanglement regularization greatly reduces the dependency between different groups of latent variables, whereas applying a vanilla KL regularization towards a prior is less useful.

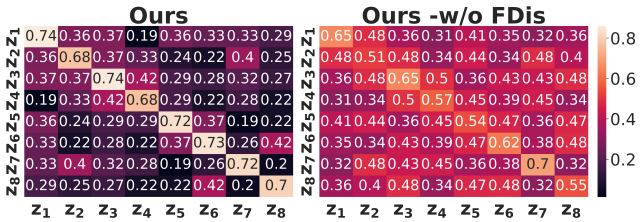

Figure 6: AvgMI matrix for factor disentanglement on Yelp, where z refers to latent variables.

Comparison with Model Variants

For further analysis, we evaluate additional variants that replace our CSS with different kinds of regularization losses, including SimCSE and the vanilla contrastive learning (VCL) loss. In Fig. 7, we observe that our model obtains significantly better results, confirming the superiority of our CSS over other prominent contrastive learning approaches. We also devise a variant that replaces our factor disentanglement with the well-known hierarchical disentanglement of Chen et al. (2018). Our model using factor disentanglement outperforms this variant with relative improvements of 3.5%, 3.4%, 0.8%, 1.6%, and 5.9% on BLEU-1, BLEU-4, ROUGE-1, ROUGE-L, and METEOR, respectively.

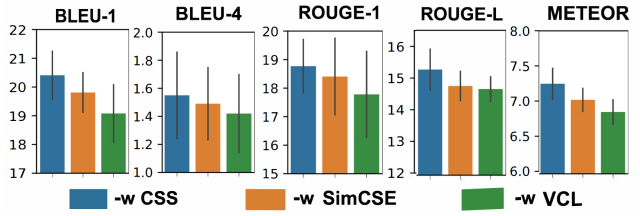

Figure 7: Explanation generation performance of model variants with different choices of contrastive loss.

Case Study

To intuitively show the improvements of our model, we present a randomly sampled explanation from Yelp in Table 4. Our model provides more specific characteristics (italic) compared with all baselines, thereby generating more concrete and informative explanations to avoid generic sentences. Moreover, as for the example in Fig. 1, our model can deliver a more informative explanation "The staff is very friendly and helpful, and customer service is impressive".

Reference	The atmosphere is relaxing and enjoyable and the food especially sandwiches are good.
NETE	The *environment* is clear.
PETER	The staff is very friendly and the *facility* is clean and well maintained.
Ours	They offer extremely *great sandwiches* and it is a great *spot* to go for *relaxing*.

Table 4: Explanations generated by different models.

Conclusion

We present a disentangled CVAE-based model that generates natural language explanations for recommender systems. Most notably, it leverages a novel disentangling mechanism to extract essential characteristics pertinent to explanation generation. Specifically, we disentangle latent variables and refine the CVAE condition using a self-regularization loss for better reconstruction. Extensive experiments demonstrate the effectiveness of our model and confirm that it can generate high-quality explanations.

Acknowledgments

This work was supported by the National Innovation 2030 Major S&T Project of China (No. 2020AAA0104200 & 2020AAA0104205), National Natural Science Foundation of China (No. 62006077 & 12105105), Shanghai Sailing Program (No. 20YF1411800), and the Natural Science Foundation of Shanghai (No. 21ZR1415800).

References

Aneja, J.; Schwing, A. G.; Kautz, J.; and Vahdat, A. 2021. A Contrastive Learning Approach for Training Variational Autoencoder Priors. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021l*, 480–493.

Banerjee, S.; and Lavie, A. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 65–72.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Berlin, Heidelberg: Springer-Verlag. ISBN 0387310738.

Cai, Z.; Wang, L.; de Melo, G.; Sun, F.; and He, L. 2022. Multi-Scale Distribution Deep Variational Autoencoder for Explanation Generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, 68–78.

Cao, Z.; Li, W.; Li, S.; and Wei, F. 2018. Retrieve, Rerank and Rewrite: Soft Template Based Neural Summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 152–161.

Chen, T. Q.; Li, X.; Grosse, R. B.; and Duvenaud, D. 2018. Isolating Sources of Disentanglement in Variational Autoencoders. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems*, 2615–2625.

Chen, Z.; Wang, X.; Xie, X.; Wu, T.; Bu, G.; Wang, Y.; and Chen, E. 2019. Co-Attentive Multi-Task Learning for Explainable Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2137–2143.

Dittadi, A.; Träuble, F.; Locatello, F.; Wuthrich, M.; Agrawal, V.; Winther, O.; Bauer, S.; and Schölkopf, B. 2021. On the Transfer of Disentangled Representations in Realistic Settings. In *9th International Conference on Learning Representations, ICLR*.

Dong, L.; Huang, S.; Wei, F.; Lapata, M.; Zhou, M.; and Xu, K. 2017. Learning to Generate Product Reviews from Attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 623–632.

Floridi, L.; and Chiriatti, M. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4): 681–694.

Harris, Z. S. 1970. *Distributional Structure*, 775–794. ISBN 978-94-017-6059-1.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. B. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 9726–9735.

Hu, Y.; Liu, Y.; Miao, C.; Lin, G.; and Miao, Y. 2021. Hierarchical Aspect-guided Explanation Generation for Explainable Recommendation. *Transactions on knowledge and data engineering*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR*.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR*.

Li, L.; Zhang, Y.; and Chen, L. 2020. Generate Neural Template Explanations for Recommendation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 755–764. ISBN 9781450368599.

Li, L.; Zhang, Y.; and Chen, L. 2021a. EXTRA: Explanation Ranking Datasets for Explainable Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2463–2469. ISBN 9781450380379.

Li, L.; Zhang, Y.; and Chen, L. 2021b. Personalized Transformer for Explainable Recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021*, 4947–4957.

Li, P.; Wang, Z.; Ren, Z.; Bing, L.; and Lam, W. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 345–354.

Liang, X.; Wu, L.; Li, J.; Wang, Y.; Meng, Q.; Qin, T.; Chen, W.; Zhang, M.; and Liu, T.-Y. 2021. R-Drop: Regularized Dropout for Neural Networks. *ArXiv*, abs/2106.14448.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81.

Ma, J.; Zhou, C.; Cui, P.; Yang, H.; and Zhu, W. 2019. Learning disentangled representations for recommendation. In *The 33rd Conference on Neural Information Processing Systems*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311–318.

Sohn, K.; Yan, X.; and Lee, H. 2015. Learning Structured Output Representation Using Deep Conditional Generative Models. In *Proceedings of the 28th International*

Conference on Neural Information Processing Systems, 3483–3491.

Song, H.; Zhang, W.-N.; Cui, Y.; Wang, D.; and Liu, T. 2019. Exploiting persona information for diverse generation of conversational responses. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 5190–5196.

Tintarev, N. 2007. Explanations of recommendations. In *Proceedings of the 2007 ACM conference on Recommender systems*, 203–206.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, H.; Zhang, F.; Wang, J.; Zhao, M.; Li, W.; Xie, X.; and Guo, M. 2018a. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 417–426.

Wang, T.; and Isola, P. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, volume 119 of *Proceedings of Machine Learning Research*, 9929–9939.

Wang, X.; Chen, Y.; Yang, J.; Wu, L.; Wu, Z.; and Xie, X. 2018b. A reinforcement learning framework for explainable recommendation. In *2018 IEEE international conference on data mining, ICDM*, 587–596.

Wen, B.; Feng, Y.; Zhang, Y.; and Shah, C. 2022. ExpScore: Learning Metrics for Recommendation Explanation. In *Proceedings of the ACM Web Conference 2022*, 3740–3744. ISBN 9781450390965.

Zhang, Y.; and Chen, X. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends in Information Retrieval*, 14(1): 1–101.

Zhang, Y.; Lai, G.; Zhang, M.; Zhang, Y.; Liu, Y.; and Ma, S. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 83–92.