# Bayesian Critiquing with Keyphrase Activation Vectors for VAE-based Recommender Systems

Hojin Yang
University of Toronto
Toronto, ON, Canada
hojin@mie.utoronto.ca

Tianshu Shen
University of Toronto
Toronto, ON, Canada
tshen@mie.utoronto.ca

Scott Sanner[*]
University of Toronto
Toronto, ON, Canada
ssanner@mie.utoronto.ca

## ABSTRACT

Critiquing is a method for conversational recommendation that incrementally adapts recommendations in response to user preference feedback. Recent advances in critiquing have leveraged the power of VAE-CF recommendation in a critiquable-explainable (CE-VAE) framework that updates latent user preference embeddings based on their critiques of keyphrase-based explanations. However, the CE-VAE has two key drawbacks: (i) it uses a second VAE head to facilitate explanations and critiquing, which can sacrifice recommendation performance of the first VAE head due to multiobjective training, and (ii) it requires iterating an inverse decoding-encoding loop for multi-step critiquing that yields poor performance. To address these deficiencies, we propose a novel Bayesian Keyphrase critiquing VAE (BK-VAE) framework that builds on the strengths of VAE-CF, but avoids the problematic second head of CE-VAE. Instead, the BK-VAE uses a Concept Activation Vector (CAV) inspired approach to determine the alignment of item keyphrase properties with latent user preferences in VAE-CF. BK-VAE leverages this alignment in a Bayesian framework to model uncertainty in a user's latent preferences and to perform posterior updates to these preference beliefs after each critique — essentially achieving CE-VAE's explanation and critique inversion through a simple application of Bayes rule. Our empirical evaluation on two datasets demonstrates that BK-VAE matches or dominates CE-VAE in both recommendation and *multi-step* critiquing performance.

## CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → *Neural networks.*

## KEYWORDS

Recommendation Systems, Critiquing, Concept Activation Vector

**ACM Reference Format:**
Hojin Yang, Tianshu Shen, and Scott Sanner. 2021. Bayesian Critiquing with Keyphrase Activation Vectors for VAE-based Recommender Systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463108

## 1 INTRODUCTION

Critiquing is a method for conversational recommendation that incrementally adapts recommendations [1] in response to user preferences. Recent work has revisited critiquing in the era of latent recommendation systems [2, 9] to propose the Critiquable-Explainable VAE (CE-VAE) [8] architecture that builds on the strengths of the VAE-CF recommender [6, 7]. While the CE-VAE makes an important step forward in critiquing, we observe a few key deficiencies that significantly limit its performance in practice (as we later demonstrate empirically): (i) it uses a second VAE head to facilitate keyphrase explanations and critiquing, which can sacrifice recommendation performance of the original VAE head due to multi-objective training, and (ii) it requires training an inverse decoding-encoding loop for critiques that further complicates model training and yields poor performance when iterated for multi-step critiquing.

To address both deficiencies, we propose a Bayesian Keyphrase critiquing VAE (BK-VAE) that takes a radically different approach to VAE-based critiquing by leveraging Concept Activation Vectors (CAVs) [3] and Bayesian inference. Specifically, we address (i) by using CAVs to determine keyphrase description alignment with a VAE-CF [6, 7] recommender that precludes the need for training a second VAE head to model keyphrases. We address (ii) by taking a Bayesian perspective to handle the inverse step of updating latent user preferences based on keyphrase critiquing feedback — the user's initial VAE-CF preference embedding serves as the prior, and we obtain posterior updates on this preference belief after observing each keyphrase critique. BK-VAE achieves CE-VAE's critique inversion loop through the simple mechanism of Bayesian updating that *naturally* facilitates multi-step critiquing updates.

We empirically compare the CE-VAE with our proposed BK-VAE on two real-world datasets showing that BK-VAE (a) matches or exceeds CE-VAE in recommendation performance, (b) effectively updates the latent user representation during multi-step critiquing, and (c) significantly outperforms CE-VAE on multi-step critiquing.

## 2 BAYESIAN CRITIQUING FOR VAE-BASED RECOMMENDER

In this section, we review keyphrase critiquing and the CE-VAE and then introduce the BK-VAE to address CE-VAE's deficiencies.

**Keyphrase-based Critiquing.** In order to provide an overview of the critiquing process, we provide a sneak preview of BK-VAE's performance in the anecdotal examples of Table 1. In brief, BK-VAE

[*] Affiliate to Vector Institute of Artificial Intelligence, Toronto

*SIGIR '21, July 11–15, 2021, Virtual Event, Canada.*
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463108

**Table 1: Conversation flow of keyphrase-based critiquing. This table shows user simulations in BK-VAE framework using MovieLens and Yelp datasets. The numbers in parentheses next to items indicate the initial recommendation ranking.**

| Dataset | Time Step $t$ | Critiqued Keyphrase | Polarity | Top-3 Recommended Items | Information of Items |
|---|---|---|---|---|---|
| MovieLens | 0 | - | - | The Pier (1), 2001: A Space Odyssey (2), 8 1/2 (3) | Drama, Sci-fi/Adventure, Fantasy/Drama |
| | 1 | Fantasy | not preferred | The Pier (1), Sunrise: A Song of Two Humans (4), The Sorrow and the Pity (22) | Drama, Romance/Drama, Documentary |
| | 2 | Action | preferred | The Wild Bunch (9), Chinatown (24), The Godfather (52) | Western, Neo-noir, American Crime |
| Yelp | 0 | - | - | Fahrenheit Coffee (1), New Toronto Fish & Chips (2), Golden Dough (3) | Cafe, Fish & Chips, Middle Eastern |
| | 1 | Chips (Fries) | not preferred | Fahrenheit Coffee (1), Golden Dough (3), The Captain's Boil (6) | Cafe, Middle Eastern, Fresh Seafood |
| | 2 | Dessert | preferred | Yan's Soy Foods (27), Mr. Chestnut (44), Allan's Pastry Shop (329) | Tofu Pudding, Roasted Chestnut, Pastry |

has access to a user's item preference history and makes three initial recommendations at time step $t = 0$. The user then provides a keyphrase critique of the recommended items along with an indication of whether they want to see more (preferred) or less (not preferred) of the keyphrase property. By leveraging keyphrase information mined from subjective user reviews, the recommender then provides a second set of recommended items at $t = 1$. This process repeats itself once more in this example for $t = 2$. One can verify in both examples that BK-VAE's updated recommendations accurately capture the user keyphrase critiques at each time step.

**CE-VAE for Keyphrase-based Critiquing.** Before we proceed to introduce the BK-VAE, we begin by reviewing the seminal CE-VAE [8] framework for keyphrase-based critiquing and it's foundation on the state-of-the-art VAE-CF recommendation model [6, 7].

Figure 1(a) shows the basic VAE-CF model for recommendation, where a (sparse) vector of user preferences $\mathbf{r}_u$ over $n$ items are encoded by the VAE [4] into a Gaussian-distributed latent preference embedding $\mathbf{z}_u$ of width $d$. $\mathbf{z}_u$ is then stochastically decoded to a (dense) reconstruction $\hat{\mathbf{r}}_u$ that generalizes user preferences to unobserved items. Formally, VAE-CF optimizes the following objective over the respective parameters $\phi$ and $\theta$ of the encoder and decoder:

$$\sum_u \log p(\mathbf{r}_u) \geq \sum_u \left[E_{q_\phi(\mathbf{z}_u|\mathbf{r}_u)}[\log p_\theta(\mathbf{r}_u|\mathbf{z}_u)] - KL[q_\phi(\mathbf{z}_u|\mathbf{r}_u)||p(\mathbf{z}_u)]\right], \tag{1}$$

In practice, the approximation of user distribution $q_\phi(\mathbf{z}_u|\mathbf{r}_u)$ is usually a Normal distribution with learned parameters $\mu_u$ and $\Sigma_u$.

To support keyphrase-based critiquing, the CE-VAE [8] models the joint probability of a user's item preferences *and* keyphrase usage preferences. Formally, the CE-VAE is trained by maximizing the amortized variational lower-bound of the joint log likelihood $\sum_u \log p(\mathbf{r}_u, \mathbf{e}_u)$. Here, $\mathbf{e}_u$ denotes the *explanation vector* (with length equal to the keyphrase dictionary size), where the value of each entry is either 1 (used by $u$) or 0 (not used), reflecting the user's keyphrase usage history. As shown in Figure 1(d), the CE-VAE augments the base VAE architecture with a second head to generate $e_u$ along with $r_u$. During critiquing, $z_u$ is naïvely updated as the average of the original user embedding and the critique embedding produced by the inverse feedback loop ($e_u \rightarrow z_u$).

With these formal definitions, we can now concretely revisit our criticisms of the CE-VAE that detract from its practical usage: (i) its two-headed architecture and multi-objective training must inherently trade off recommendation performance with keyphrase modeling accuracy, and (ii) averaging the latent user update and critique embeddings from the inverse encoding loop is naïve and yields poor performance when iterated for multi-step critiquing.

**Keyphrase Explanation Alignment.** The CE-VAE framework was proposed to achieve alignment between keyphrase explanations and latent user preferences in order to facilitate critiquing tasks. Inspired by CAV [3] methodology, BK-VAE achieves the same goal by deriving Keyphrase Activation Vectors (KAV) with the *off-the-shelf* VAE-CF framework, addressing CE-VAE's first drawback.

To achieve this goal, we first observe in (1) that it is common to use a single layer decoder with Mean Squared Error (MSE) reconstruction loss for explicit ratings. We denote this decoder's weight matrix as $X \in \mathbb{R}^{n \times d}$, where the $i^{\text{th}}$ vector $\mathbf{x}_i$ is used to compute item $i$'s rating given the latent preference embedding $\mathbf{z}_u$ of user $u$. Viewing the MSE from the lens of a log likelihood, we recognize

$$P(r_{u,i}|\mathbf{z}_u, \mathbf{x}_i) = \mathcal{N}({\mathbf{x}_i}^T \mathbf{z}_u, \tau_r^{-1}), \tag{2}$$

where the precision (i.e., inverse variance) $\tau_r$ is a hyperparameter.

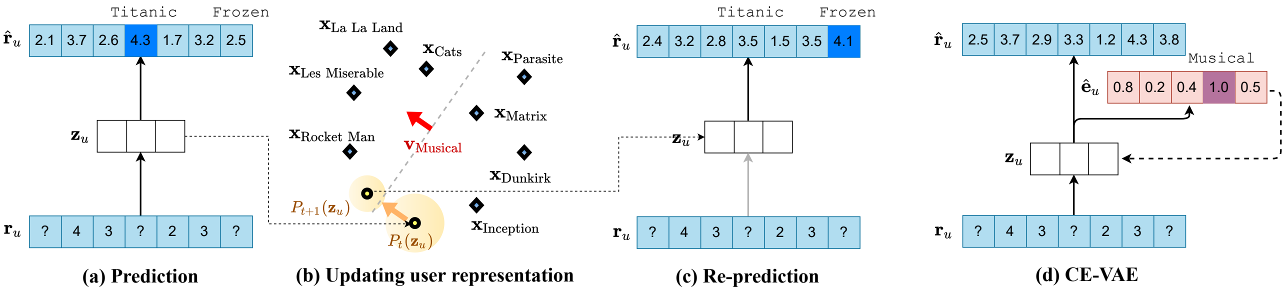
To find the KAV for keyphrase $k$ (e.g., *Musical*), we consider the item embedding vectors from the decoder matrix $X$. We define a KAV for $k$ as the normal to a hyperplane that separates items with and without $k$ in the item embedding space as shown in Figure 1(b). To obtain the activation vector for $k$, we sample $m$ examples from a positive set of items $\mathcal{I}_k$ (e.g., *Les Miserables*, *Cats*), and a negative set ${\mathcal{I}_k}^c$(e.g., a set of random movies), respectively. A binary linear classifier is trained to distinguish between the embeddings of the two groups. Instead of training once, we perform multiple training runs, and use the averaged classifier $\mathbf{v}_k \in \mathbb{R}^d$ as the KAV for $k$.

**Posterior Belief Updating After Critiques.** With a KAV $\mathbf{v}_k$ for each keyphrase $k$, we can now perform a closed-form Bayesian update over latent user preferences $\mathbf{z}_u$ in response to critiques over $k$, thus obviating the need for CE-VAE's inverse feedback loop.

As KAVs are aligned with the latent item (and user) embedding space, we adapt the log likelihood view of (2) to generate a Normal distribution over keyphrase preference $y_{u,k}$ proportional in strength to the inner product of keyphrase $\mathbf{v}_k$ and user $\mathbf{z}_u$ embeddings:

$$P(y_{u,k}|\mathbf{z}_u, \mathbf{v}_k) = \mathcal{N}({\mathbf{v}_k}^T \mathbf{z}_u, \tau_y^{-1}), \tag{3}$$

where precision $\tau_y$ is a constant hyperparameter. With this likelihood of keyphrase preference feedback $\tilde{y}_{u,k}$, Bayesian updating of user $u$'s latent preferences $\mathbf{z}_u$ for fixed $\mathbf{v}_k$ is essentially a Bayesian linear regression; updated beliefs over $\mathbf{z}_u$ can then be decoded through BK-VAE's decoder to produce a recommendation consistent with the keyphrase critiques. In our critiquing setting, a user can express keyphrase preferences with either positive or negative polarity. We facilitate such binary feedback by mapping the negative and positive feedback to $\tilde{y}_{u,k} = min_k {\mathbf{v}_k}^T \mathbf{z}_u$ and $\tilde{y}_{u,k} = max_k {\mathbf{v}_k}^T \mathbf{z}_u$, respectively, to calibrate upper and lower bounds of the user's initial keyphrase preference range as likelihood targets.



**(a) Prediction** **(b) Updating user representation** **(c) Re-prediction** **(d) CE-VAE**

**Figure 1: (a)~(c): Step-by-step flow of BK-VAE architecture. (a) An initial user representation is obtained from off-the-shelf VAE-CF models. (b) Then the system performs a posterior update to latent user preference beliefs after each critique using KAV and (c) generates a revised recommendation. The user may continue with further rounds of critiquing by repeating (b) and (c), or terminate with the best recommendation. (d) The previously proposed CE-VAE architecture [8].**

**Table 2: Summary of datasets.**

| Dataset | # Users | # Items | # Keyphrases | # Ratings | Sparsity |
|---|---|---|---|---|---|
| MovieLens | 69,878 | 10,677 | 164 | 10,000,054 | 98.66% |
| Yelp | 7,000 | 4,997 | 245 | 203,683 | 99.42% |

Once the system receives a user's critique on keyphrase $k$ with the corresponding preference $\tilde{y}_{u,k}$ at critiquing step $t$, the latent user's belief can be updated through Bayes rule:

$$P_{t+1}(\mathbf{z}_u) = P_{t+1}(\mathbf{z}_u|\mathbf{v}_k, \tilde{y}_{u,k}) \propto P_t(\tilde{y}_{u,k}|\mathbf{v}_k, \mathbf{z}_u)P_t(\mathbf{z}_u), \text{where } t \geq 0.$$

Specifically, the approximation of user distribution $q_\phi(\mathbf{z}_u|\mathbf{r}_u)$ in (1) serves as an informed prior over $u$'s latent preferences, which is denoted as $P_0(\mathbf{z}_u)$. Since both the likelihood and prior are Gaussian and form a conjugate prior-likelihood pair, the posterior is also Gaussian and can be computed incrementally in closed-form as follows (we omit the user subscript $u$ to reduce notational clutter):

$$\begin{aligned} P_{t+1}(\mathbf{z}_u) &= \mathcal{N}(\mu_{t+1}, \Sigma_{t+1}) \\ \text{where } \mu_{t+1} &= \Sigma_{t+1}({\Sigma_t}^{-1}\mu_t + \tilde{y}_k\tau_y\mathbf{v}_k) \\ \text{and } \Sigma_{t+1} &= ({\Sigma_t}^{-1} + \tau_y\mathbf{v}_k{\mathbf{v}_k}^T)^{-1}. \end{aligned}$$

Comparing to CE-VAE's naïve average-weighting algorithm for user belief updating which discards $\Sigma_0$ after training, BK-VAE obtains a per-dimensional weighted update through Bayesian inferences by fully utilizing the approximated user distribution. The likelihood precision $\tau_y$ indicates the system's degree of confidence in each user's critique. Technically, the larger the precision is, the more the system leverages the information when it updates user beliefs. By tuning this, the system can modulate the magnitude of the user's critiquing (e.g., extremely positive, neutral negative).

## 3 EXPERIMENTS AND EVALUATION

In this section, we evaluate BK-VAE by comparing it to CE-VAE [8] on two different benchmark datasets.[1] We evaluate: (RQ1) initial rating prediction recommendation performance comparison, (RQ2) a diagnostic analysis of BK-VAE's single-step critiquing, and (RQ3) multi-step critiquing performance comparison.

**Table 3: Hyper-parameters tuned on the experiments.**

| Functionality | Range | Algorithms affected |
|---|---|---|
| Latent Dimension | {50, 100, 150, 200} | BK-VAE, CE-VAE |
| Learning Rate | {1e-5, 1e-4, 1e-3, 1e-2} | BK-VAE, CE-VAE |
| Dropout Rate | {0., 0.3, 0.5} | BK-VAE, CE-VAE |
| L2 Regularization | {0., 1e-5, 1e-4 ··· 1e-1} | BK-VAE, CE-VAE |
| KL Regularization | {0.3, 0.5} | BK-VAE, CE-VAE |
| Relative Weighting for Explanation Head | {1e-4, 1e-3 ··· 1} | CE-VAE |
| Relative Weighting for Inverse Network | {1e-4, 1e-3 ··· 1} | CE-VAE |

**Datasets.** We conduct experiments on two datasets: MovieLens-10M (MovieLens) for movie recommendation, and Yelp for business recommendation. All datasets have keyphrase description assignments for items provided by users. MovieLens contains social tags; typically single words or short phrases assigned by users to movies. For Yelp, we follow the preprocessing steps described in [5] to extract keyphrases from user reviews. We only keep keyphrases that have been assigned by at least 15 users/items for both datasets. Table 2 shows the overall dataset statistics. All experiments use 20% of the data as a test set, and the remaining data is divided according to a ratio of of 4:1 into the training and validation set.

**Baseline.** Two main modifications were performed on CE-VAE [8] as the baseline model. First, while the original paper only suggested to zero-out $\mathbf{e}_u$ for keyphrases with negative critiques, we also implemented a one-out variation to accommodate positive critiques. Second, instead of using implicit feedback, we use CE-VAE in the same explicit rating-based setting as BK-VAE. For fair comparison, both BK-VAE and CE-VAE use the same backbone VAE structures. Table 3 presents our hyperparameter definitions and sweeps for the architecture and algorithm tuning on the held-out validation set.

**RQ1: Initial Rating Prediction Performance Comparison.** Table 4 shows the pre-critiquing recommendation performance comparison of VAE-CF (used by BK-VAE) with CE-VAE using the RMSE metric. As the results show, VAE-CF (BK-VAE) matches CE-VAE on Yelp and significantly outperforms CE-VAE on MovieLens.

**RQ2: Single-step Critiquing Behavior Analysis.** Before we proceed to compare CE-VAE and BK-VAE, we first wish to test a diagnostic proof-of-concept for BK-VAE's single-step critiquing.

[1] https://github.com/hojinyang/bayesian-critiquing-recommender

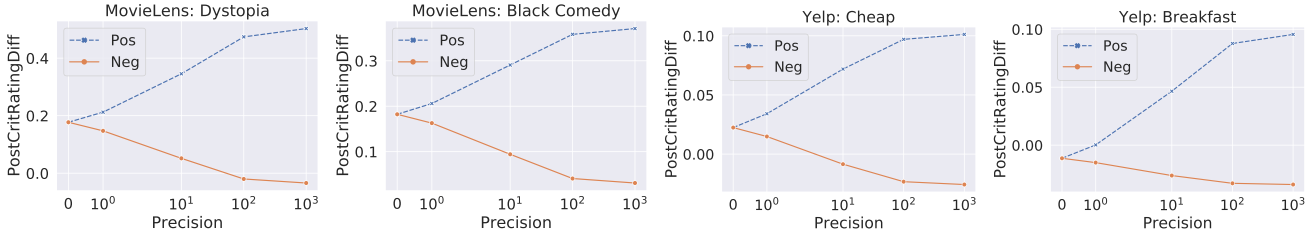


**Figure 2: Control over recommendations when keyphrase likelihood precision is adjusted with the values of $\{0, 10^0, 10^1, 10^2, 10^3\}$ for both positive and negative single-step critiquing cases. We observe the intended response as we increase KAV certainty.**

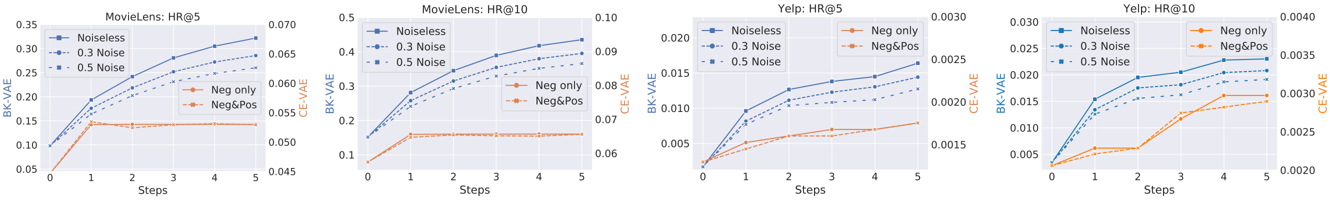


**Figure 3: HR@$\{5, 10\}$ comparison during the conversation session where the left and right y-axes represent the HR of BK-VAE and CE-VAE, respectively (we have dual axes to show variation in CE-VAE performance more clearly). While we focus on both negative and positive critiquing, we also test negative-only critiquing (originally proposed in [8]) for CE-VAE. We additionally test under noisy critiquing for BK-VAE, where users may provide a random critique with probability $\{0.3, 0.5\}$.**

**Table 4: Comparison of VAE-CF (used by BK-VAE) against CE-VAE; test RMSE with 95% confidence interval of 5-runs.**

| Dataset | CE-VAE | VAE-CF (BK-VAE) |
|---|---|---|
| MovieLens | 0.9328 ± 0.0079 | 0.7884 ± 0.0022 |
| Yelp | 1.0011 ± 0.0114 | 1.0015 ± 0.0152 |

Ideally, after user $u$'s positive critique on $k$, the updated ratings of items in $\mathcal{I}_k$ increase compared to those in $\mathcal{I}_k{}^c$, and decrease when user negatively critiqued on $k$. Hence, we measure the normalized difference of average rating of items in $\mathcal{I}_k$ compared to those in $\mathcal{I}_k{}^c$ after critiquing on $k$. Specifically,

$$\text{PostCritRatingDiff}(u, k; pol, prec) = \frac{\text{AR}_u^k(\mathcal{I}_k) - \text{AR}_u^k(\mathcal{I}_k{}^c)}{\text{AR}_u^k(\mathcal{I}_k{}^c)}, \qquad (4)$$

where $\text{AR}_u^k(\cdot)$ is the average rating of items in a given set after $u$'s critiquing on $k$, with polarity $pol$ and keyphrase precision $prec$.

In Figure 2, we report average PostCritRatingDiff over all users for a wide range of $prec$ values for both polarities. The result confirms that a user's positive critique on $k$ leads to an increase in the average rating of items in $\mathcal{I}_k$ compared to $\mathcal{I}_k{}^c$ (vice versa for negative critiques), and the gap increases as $prec$ (certainty) increases.

**RQ3: Multi-step Critiquing Performance Comparison.** We conduct user simulation to comparatively evaluate CE-VAE and BK-VAE in a multi-step conversational recommendation scenario using our datasets. Specifically, given an observed user–item-rating triplet (u, i, r) in the test set, we may select item i as the ground truth target to recommend when r ≥ 4. For critique selection, we assume the user may prefer to critique a keyphrase that deviates the most from the known target item description for both polarities. At each step, we compare the top-10 recommended items' averaged keyphrase frequency to the target item's keyphrase frequency. Then we critique with the keyphrase having the largest frequency differential. We track the conversational interaction session of simulated users, repeating critique selection for 5 rounds. We measure recommendation quality using Hit-Rate@$\{5, 10\}$.

From Figure 3, we observe the following: (i) BK-VAE outperforms CE-VAE by a *significant* margin in multi-step critiquing. Further, the margin increases as the critiquing interactions progress. Compared to the poor performance of CE-VAE's trained inverse network, it appears that the closed-form Bayesian update of BK-VAE accurately models preference belief updates after critiques. (ii) While BK-VAE's critiquing performance decreases as the response noise level increases, BK-VAE with noise still outperforms CE-VAE.

Despite *extensive* hyper-parameter tuning of the CE-VAE model, it performs quite poorly on multi-step critiquing compared to BK-VAE. To recap, we hypothesize that (i) the additional head of CE-VAE makes it more difficult to train compared to BK-VAE's single head and (ii) the naïve averaging of user latent preferences and inverse critique embeddings performs poorly in the multi-step setting.

## 4 CONCLUSION AND FUTURE WORK

We introduced BK-VAE, a novel keyphrase-based critiquing framework for conversational recommendation built on the VAE-CF framework that avoids key caveats of the previously proposed architecture, CE-VAE [8]. Key results show that BK-VAE matches or outperforms CE-VAE in both recommendation and *multi-step* critiquing. The simplicity of BK-VAE combined with its strong performance and Bayesian uncertainty model should enable versatile future extensions such as a mixed-initiative preference elicitation *and* critiquing framework to actively elicit keyphrase preferences.

## REFERENCES

[1] Li Chen and Pearl Pu. 2012. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction* 22, 1-2 (2012), 125–150.

[2] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 173–182. https://doi.org/10.1145/3038912.3052569

[3] Been Kim, M. Wattenberg, J. Gilmer, C. J. Cai, James Wexler, F. Viégas, and Rory Sayres. 2018. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *ICML*. Stockholm, Sweden.

[4] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6114

[5] Hanze Li, Scott Sanner, Kai Luo, and Ga Wu. 2020. A Ranking Optimization Approach to Latent Linear Critiquing in Conversational Recommender System. In *ACM RecSys-20*. Online.

[6] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 305–314.

[7] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *WWW-18*. Lyon, France.

[8] Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. 2020. Deep Critiquing for VAE-based Recommender Systems. In *ACM SIGIR-20*. Xi'an, China.

[9] S. Sedhain, A. Menon, S. Sanner, and L. Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *Proceedings of the 24th International Conference on the World Wide Web (WWW-15)*. Florence, Italy.