Fast Multi-Step Critiquing for VAE-based Recommender Systems

DIEGO ANTOGNINI and BOI FALTINGS, École Polytechnique Fédérale de Lausanne, Switzerland

Recent studies have shown that providing personalized explanations alongside recommendations increases trust and perceived quality. Furthermore, it gives users an opportunity to refine the recommendations by *critiquing* parts of the explanations. On one hand, current recommender systems model the recommendation, explanation, and critiquing objectives jointly, but this creates an inherent trade-off between their respective performance. On the other hand, although recent latent linear critiquing approaches are built upon an existing recommender system, they suffer from computational inefficiency at inference due to the objective optimized at each conversation's turn. We address these deficiencies with M&Ms-VAE, a novel variational autoencoder for recommendation and explanation that is based on multimodal modeling assumptions. We train the model under a weak supervision scheme to simulate both fully and partially observed variables. Then, we leverage the generalization ability of a trained M&Ms-VAE model to embed the user preference and the critique separately. Our work's most important innovation is our critiquing module, which is built upon and trained in a self-supervised manner with a simple ranking objective. Experiments on four real-world datasets demonstrate that among state-of-the-art models, our system is the first to dominate or match the performance in terms of recommendation, explanation, and multi-step critiquing. Moreover, M&Ms-VAE processes the critiques up to 25.6x faster than the best baselines. Finally, we show that our model infers coherent joint and cross generation, even under weak supervision, thanks to our multimodal-based modeling and training scheme.

Additional Key Words and Phrases: Conversational Recommendation, Critiquing, Variational Autoencoder

ACM Reference Format:
Diego Antognini and Boi Faltings. 2021. Fast Multi-Step Critiquing for VAE-based Recommender Systems. In *Fifteenth ACM Conference on Recommender Systems (RecSys '21), September 27-October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3460231.3474249

1 INTRODUCTION

Recommender systems accurately capture user preferences and achieve high performance. However, they offer little transparency regarding their inner workings. It has been shown that providing explanations along with item recommendations enables users to understand why a particular item has been suggested and hence to make better decision [3, 6]. Additionally, explanations increase the system's overall transparency and trustworthiness [21, 40, 49].

An important advantage of explanations is that they provide a basis for feedback. If users understand what has generated the suggestions, they can refine the recommendations by interacting directly with the explanations. Critiquing is a conversational recommendation method that incrementally adapts recommendations in response to user preferences [8]. Example critiquing was introduced in information retrieval [44] and first applied to recommender systems in [5]. Recognizing that critiquing is most useful when applied in multiple steps, [24] and [29] introduced mechanisms based on constraint programming [41] with an application to travel planning. Multi-step critiquing with constraint programming was recognized as a form of preference elicitation, which enabled the analysis and optimization of its performance [13] and the addition of suggestions for active preference elicitation [43], which yielded dramatic improvements in decision accuracy in user studies. Multi-step critiquing was also shown to be superior to compound

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
Manuscript submitted to ACM

critiquing, which groups multiple attributes in a single step [31]. A major limitation of all these approaches is that items have to be characterized by a set of discrete attributes.

After nearly a decade in which critiquing approaches received little attention, [45] introduced a collaborative filtering recommender with explanations and an embedding-based critiquing method. This method allows users to critique the recommendation using arbitrary languages; a set of attributes is mined from reviews, and the users can interact with them. Other works built upon the same paradigm [2, 9]. [26] showed that those models suffer from unstable training and high computational complexity, and they proposed a framework based on a variational autoencoder [20, 23]. However, these models learn a bidirectional mapping between the critique and the user latent space. This creates an inherent trade-off between the recommendation and explanation performance, and it yields poor results in multi-step critiquing.

Recently, [25] proposed a latent linear critiquing (LLC) method built upon the recommendation model PLRec [35]. LLC co-embeds keyphrase attributes in the same embedding space as the recommender. The critiquing process consists of a weighted average between the user-preference embedding and the critique embeddings obtained through the conversation. The weights are optimized in a linear programming formulation using a max-margin scoring-based objective (i.e., the pairwise difference of scores of items affected by the critique and the others). Following the same methodology, [22] changed the objective into a ranking-based one. While those models obtain good performance in multi-step critiquing, they suffer from computational inefficiency due to the objective function optimized at each turn.

To address both issues, we present M&Ms-VAE, a novel variational autoencoder for recommendation and explanation with a separate critiquing module. Inspired by multimodal generative models [37, 39, 42, 47], we treat the user's past interactions and keyphrase usage as different partially observed variables, and more importantly, we assume conditional independence between them. We can then approximate the variational joint posterior using a mixture of experts. We propose a training scheme that mimics weakly supervised learning to train the inference networks jointly but also independently. This is essential to our modeling, because M&Ms-VAE is robust to a missing unobserved variable and can thus embed separately and efficiently the user interactions, the keyphrases, and the critique, respectively.

In a second step, we leverage the generalization ability of M&Ms-VAE and design a novel blending module to re-rank recommended items according to a critique. The latter is trained only once on a synthetic dataset with a self-supervision objective. This generalizes into multi-step critiquing and enables fast critiquing.

To the best of our knowledge, this is the first work to revisit deep critiquing from the perspective of multimodal generative models and to propose a blending module trained in a simple self-supervised fashion. We evaluate our method using four real-world datasets. The results demonstrate that the proposed M&Ms-VAE model (1) achieves superior or competitive performance in terms of recommendation, explanation, and multi-step critiquing in comparison to the state-of-the-art recommendation and critiquing methods, (2) processes the critiques up to 26x faster than the best baselines and up to 9x faster using only the CPU, and (3) induces coherent joint and cross generation, even under weak supervision.

2 PRELIMINARIES

This section introduces the notation used in the paper and the variational autoencoder for recommendation [23]. Then, we review a recent study [26] that built upon [23] and revisited critiquing by proposing the critiquable-explainable VAE (CE-VAE) model. Finally, we highlight the key deficiencies that significantly limit its performance in practice.

2.1 Notation

Before proceeding, we define the following notation used throughout this paper:

- U, I, and K: The user, the item, and the keyphrase sets, respectively.
- $\boldsymbol{R} \in \mathbb{R}^{|U|\times|I|}$: The user-by-item interaction matrix obtained with implicit feedback. Entries $\boldsymbol{r}_{u,i}$ of 1 (respectively 0) denote a positive (respectively negative or unobserved) interaction between the user u and item i.
- $\boldsymbol{K} \in \mathbb{R}^{|U|\times|K|}$: The binary user-keyphrase matrix that reflects the user u's keyphrase-usage preference. Given user reviews from a corpus, we extract keyphrases that describe item attributes from all reviews (see Section 4.1).
- $\boldsymbol{K^I} \in \mathbb{R}^{|I|\times|K|}$: The binary item-keyphrase matrix. The process is similar to $\boldsymbol{K}$ with the aggregation per item.
- $\hat{\boldsymbol{r}}_u \in \mathbb{R}^{|I|}$ and $\hat{\boldsymbol{k}}_u \in \mathbb{R}^{|K|}$: The predicted feedback and keyphrase explanation, respectively.
- $\boldsymbol{z}_u \in \mathbb{R}^{|H|}$: The user u's latent embedding of dimension H from the observed interaction $\boldsymbol{r}_u$ and keyphrase-usage preference $\boldsymbol{k}_u$.
- $\boldsymbol{c}_u^t \in \mathbb{R}^{|K|}$: A one-hot vector of length $|K|$. The only positive value indicates the index of the keyphrase to be critiqued by the user u at a given step t of the user interaction with the recommender system.
- $\boldsymbol{z}_u^t \in \mathbb{R}^{|H|}$: The latent representation of the critique $\boldsymbol{c}_u^t$.
- $\tilde{\boldsymbol{z}}_u^t \in \mathbb{R}^{|H|}$: The updated latent representation of the user after the critique $\boldsymbol{c}_u^t$.
- $I^{+\boldsymbol{c}} \in \{i|\boldsymbol{k}^I_{i,\boldsymbol{c}} = 1, \forall i \in I\}$: The set of items that contain the critiqued keyphrase $\boldsymbol{c}$.
- $I^{-\boldsymbol{c}} \in \{i|\boldsymbol{k}^I_{i,\boldsymbol{c}} = 0, \forall i \in I\}$: The set of items that do not contain the critiqued keyphrase $\boldsymbol{c}$.

2.2 Variational Autoencoder for Recommendation (VAE)

A variational autoencoder (VAE) [20] is a generative model of the form $p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p_\theta(\boldsymbol{x}|\boldsymbol{z})$, where $p(\boldsymbol{z})$ is a prior and the likelihood $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is parametrized by a neural network with parameters θ. The model learns to maximize the marginal likelihood of the data $p_\theta(\boldsymbol{x})$ (i.e., the evidence that is intractable) by approximating the true unknown posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ with a variational posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. Applied to recommendation systems, the collaborative-filtering VAE [23] considers as input data the sparse user preferences $\boldsymbol{r}_u$ over $|I|$ items. More formally, the model optimizes a variational lower bound on the log likelihood of all observed user feedback $\sum_{u\in U}\log p(\boldsymbol{r}_u)$ through stochastic gradient descent:

$$\log p(\boldsymbol{r}_u) \geq \int_{\boldsymbol{z}_u} q_\phi(\boldsymbol{z}_u|\boldsymbol{r}_u)\log\frac{p_\theta(\boldsymbol{r}_u, \boldsymbol{z}_u)}{q_\phi(\boldsymbol{z}_u|\boldsymbol{r}_u)}d\boldsymbol{z}_u \geq \mathbb{E}_{q_\phi(\boldsymbol{z}_u|\boldsymbol{r}_u)}\left[\log p_\theta(\boldsymbol{r}_u|\boldsymbol{z}_u)\right] - \beta\,\mathrm{D}_{\mathrm{KL}}\left[q_\phi(\boldsymbol{z}_u|\boldsymbol{r}_u)\;||\;p(\boldsymbol{z}_u)\right], \tag{1}$$

where $\boldsymbol{z}_u$ is sampled[1] from the distribution $q_\phi(\boldsymbol{z}_u|\boldsymbol{r}_u)$ with parameters $\boldsymbol{\mu}_u$ and Σ_u, and $\mathrm{D}_{\mathrm{KL}}[q, p]$ denotes the Kullback-Leibler divergence (KL) between the distributions p and q. In practice, the prior $p(\boldsymbol{z})$ is usually a spherical Gaussian with parameters $\boldsymbol{\mu}$ and Σ. Finally, β is a hyperparameter that controls the strength of the regularization relative to the reconstruction error, as motivated by the β-VAE of [18], and is slowly annealed to 1, similarly to [4].

2.3 Co-embedding of Language-based Feedback with the Variational Autoencoder (CE-VAE)

Thus far, the variational autoencoder can only recommend items without generating any form of explanation. A recent study [26] proposed the CE-VAE model, which integrates an explanation and critiquing module based on keyphrases. The authors support critiquing by first modeling the joint probability of a user's item preferences and keyphrase usage:

$$\begin{aligned}\log p(\boldsymbol{r}_u, \boldsymbol{k}_u) = \log p(\boldsymbol{k}_u|\boldsymbol{r}_u) + \log p(\boldsymbol{r}_u) = \mathbb{E}_{q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)}\left[\log p_{\Theta_k}(\boldsymbol{k}_u|\boldsymbol{z}_u)\right] - \mathrm{D}_{\mathrm{KL}}\left[q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)\;||\;p(\boldsymbol{z}_u)\right] \\ + \mathbb{E}_{q_{\Psi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)}\left[\log p_{\Theta_r}(\boldsymbol{r}_u|\boldsymbol{z}_u)\right] + \mathcal{H}\left[q_{\Psi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)\right] + \mathbb{E}_{q_{\Psi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)}\left[p(\boldsymbol{z}_u)\right],\end{aligned} \tag{2}$$

where $\mathcal{H}$ is the entropy. Then, they incorporate an additional objective to learn a projection from the critiquing feedback into the latent space via another encoder (an inverse feedback loop). In other words, they reintroduce the user's

[1] Using the reparametrization trick [20, 33]: $\boldsymbol{z}_u = \boldsymbol{\mu}_u + \epsilon\boldsymbol{\sigma}_u$, where $\epsilon \sim \mathcal{N}(0, \mathbb{I}_H)$.

keyphrase usage $\boldsymbol{k}_u$ to approximate the variational lower bound of $p(z_u)$ by marginalizing over $\boldsymbol{k}_u$. More formally:

$$\begin{aligned}\log p(z_u) &\geq \mathbb{E}_{q(\boldsymbol{k}_u|z_u)}\big[\log p(z_u|\boldsymbol{k}_u)\big] - \mathrm{D_{KL}}\big[q(\boldsymbol{k}_u|z_u) \,||\, p(\boldsymbol{k}_u)\big] \\ &\approx \mathbb{E}_{p_{\Theta_k}(\boldsymbol{k}_u|z_u)}\big[\log p_{\Theta'_k}(z_u|\boldsymbol{k}_u)\big] - \mathrm{D_{KL}}\big[p_{\Theta_k}(\boldsymbol{k}_u|z_u) \,||\, p(\boldsymbol{k}_u)\big],\end{aligned} \tag{3}$$

where $p(\boldsymbol{k}_u)$ is a prior following a standard normal distribution and the weights of $q(\boldsymbol{k}_u|z_u)$ are shared with $p_{\Theta_k}(\boldsymbol{k}_u|z_u)$.

Finally, once the model is trained on the full objective function, the critiquing process for the critique $\boldsymbol{c}_u$ is performed as follows: (1) compute the critique representation z_u^c with $p_{\Theta'_k}(z_u|\boldsymbol{k}_u)$, (2) average both the user latent representation z_u and the critique representation z_u^c, and (3) predict the new feedback $\hat{\boldsymbol{r}}_u$ with the generative network $p_{\Theta_r}(\boldsymbol{r}_u|z_u)$.

Overall, the CE-VAE framework is effective in practice for recommendation, keyphrase explanation, and single-step critiquing. However, it suffers from two key deficiencies that limit its performance (as we later show empirically):

(1) The model learns a function to project the critiqued keyphrase into the user's latent space, from which the feedback and the explanation are predicted. This mapping is learned via an autoencoder, which perturbs the training. Thus, there is an inherent trade-off between the performance of the recommendation and that of the explanation.
(2) Although the joint objective also maximizes a latent representation likelihood with the Kullback-Leibler terms, it is unclear whether the inverse function embeds the critique effectively and whether the mean reflects a critiquing mechanism.

3 M&MS-VAE: A MIXTURE-OF-EXPERTS MULTIMODAL VARIATIONAL AUTOENCODER

Our goal is to build a more generalizable representation of users' preferences that is based on their observed interactions and keyphrase usage. Figure 1 depicts the graphical model of our proposed M&Ms-VAE, and Figure 2 shows the training scheme. Then, we leverage this representation to efficiently embed the user critiques and learn, in a self-supervised fashion, a blending module to re-rank recommended items for multi-step critiquing. Figure 3 illustrates the workflow.

3.1 Model Overview

Like previously developed variational autoencoders for recommendation, we assume that the observed user u's interactions $\boldsymbol{r}_u$ and the keyphrase-usage preference $\boldsymbol{k}_u$ are generated from a latent representation of the user preferences.

Differently from prior work, we seek to learn the joint distribution $p(\boldsymbol{r}_u, \boldsymbol{k}_u)$ under weak supervision. Our main goal is to learn a more generalizable representation of the user preferences. Therefore, we aim to design a generative model that can recommend and generate keyphrase explanation **jointly but also independently** from each of the observed variables (i.e., cross-modal generation). It also allows us to apply the same technique to users who have not written reviews or to cases in which keyphrases are unavailable. If this goal is achieved, we can then embed effectively the user's observed interactions, the user's keyphrase preference, and the critique with the **same** inference network $q_\Phi(z_u|\boldsymbol{r}_u, \boldsymbol{k}_u)$.

Inspired by multimodal generative models [37, 39, 42, 47], we treat $\boldsymbol{r}_u$ and $\boldsymbol{k}_u$ as different modalities, and we assume they are conditionally independent given the common latent variable z_u. In other words, we assume a generative model of the form $p_\Theta(\boldsymbol{r}_u, \boldsymbol{k}_u, z_u) = p(z_u)p_{\Theta_r}(\boldsymbol{r}_u|z_u)p_{\Theta_k}(\boldsymbol{k}_u|z_u)$. An advantage of such a factorization is that if $\boldsymbol{r}_u$ or $\boldsymbol{k}_u$ is unobserved, we can safely ignore it when evaluating the marginal likelihood [47].

We start with the derivation of the joint log likelihood $\sum_{u \in U} \log p(\boldsymbol{r}_u, \boldsymbol{k}_u)$ over the observed interactions $\boldsymbol{r}_u$ and keyphrase-usage preference $\boldsymbol{k}_u$ and all users u as shown in Figure 1:

$$\log\ p(\boldsymbol{r}_u, \boldsymbol{k}_u) = \log \int_{z_u} p_\Theta(\boldsymbol{r}_u, \boldsymbol{k}_u, z_u) dz_u \geq \mathbb{E}_{q_\Phi(z_u|\boldsymbol{r}_u,\boldsymbol{k}_u)}\big[\log p_\Theta(\boldsymbol{r}_u, \boldsymbol{k}_u|z_u)\big] - \beta\, \mathrm{D_{KL}}\big[q_\Phi(z_u|\boldsymbol{r}_u, \boldsymbol{k}_u) \,||\, p(z_u)\big], \tag{4}$$

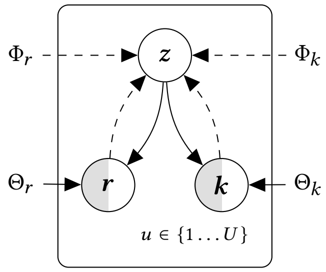


Fig. 1. Probabilistic-graphical-model view of our proposed M&Ms-VAE model. Both the implicit feedback $\boldsymbol{r}_u$ and the keyphrase $\boldsymbol{k}_u$ are generated from user u's latent representation $\boldsymbol{z}_u$. Solid lines denote the generative model, whereas dashed lines denote the variational approximation.

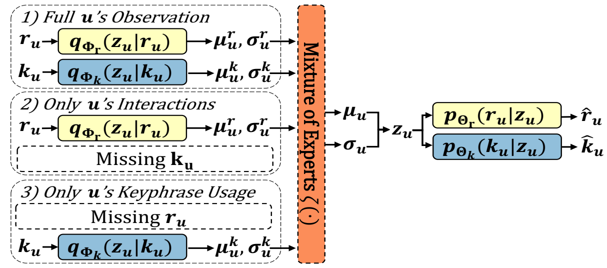


Fig. 2. The proposed M&Ms-VAE architecture and training scheme. Each pass infers the parameters $\boldsymbol{\mu}_u$ and $\boldsymbol{\sigma}_u$ with the mixture of experts using either the joint inference network $q_\Phi(\boldsymbol{z}_u|\boldsymbol{r}_u, \boldsymbol{k}_u)$ or one of the individual networks ($q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)$ or $q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)$, respectively). The final gradient is computed on the sum of each $ELBO(\cdot)$ term.

where we assume that the prior distribution $p(z)$ is a standard normal distribution and β is a hyperparameter that controls the strength of the regularization relative to the reconstruction error. Thanks to our assumption that $\boldsymbol{r}_u$ and $\boldsymbol{k}_u$ are conditionally independent given the common latent variable $\boldsymbol{z}_u$, we can rewrite Equation 4 as follows:

$$ELBO(\boldsymbol{r}_u, \boldsymbol{k}_u) = \mathbb{E}_{q_\Phi(\boldsymbol{z}_u|\boldsymbol{r}_u,\boldsymbol{k}_u)}\left[\log p_{\Theta_r}(\boldsymbol{r}_u|\boldsymbol{z}_u) + \log p_{\Theta_k}(\boldsymbol{k}_u|\boldsymbol{z}_u)\right] - \beta\, \mathrm{D_{KL}}\left[q_\Phi(\boldsymbol{z}_u|\boldsymbol{r}_u, \boldsymbol{k}_u) \;||\; p(\boldsymbol{z}_u)\right]. \quad (5)$$

Learning the variational joint posterior $q_\Phi(\boldsymbol{z}_u|\boldsymbol{r}_u, \boldsymbol{k}_u)$ of Equation 5 under its current form requires $\boldsymbol{r}_u$ and $\boldsymbol{k}_u$ to be presented at all times, thus making cross-modal recommendation difficult. Following our assumption, we can factorize the joint variational posterior as a function $\zeta(\cdot)$ of unimodal posteriors (or experts) $q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)$ and $q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)$: $q_\Phi(\boldsymbol{z}_u|\boldsymbol{r}_u, \boldsymbol{k}_u) = \zeta\left(q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u), q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)\right)$, similarly to [37, 39, 42, 47]. In our case, the function $\zeta(\cdot)$ should be (1) robust to overconfident experts if the marginal posterior $q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)$ or $q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)$ has low density, and (2) robust to missing unobserved variable $\boldsymbol{r}_u$ or $\boldsymbol{k}_u$. Therefore, we propose to rely on a mixture of experts (MoE) with uniform weights:

$$\zeta\left(q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u), q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)\right) = \alpha \cdot q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u) + (1-\alpha) \cdot q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u) \text{ with } \alpha = \begin{cases} \frac{1}{2}, & \text{if } \boldsymbol{r}_u \text{ and } \boldsymbol{k}_u \text{ are observed,} \\ 1, & \text{if only } \boldsymbol{r}_u \text{ is observed,} \\ 0, & \text{if only } \boldsymbol{k}_u \text{ is observed.} \end{cases} \quad (6)$$

We set the weights uniformly to explicitly enforce an equal contribution from each $\boldsymbol{r}_u$ and $\boldsymbol{k}_u$ when both are observed during training. In the case of an unobserved modality, we shift the importance distribution toward the presented one, which generalizes to weakly supervised learning (see Section 3.2). This is an important factor, because the inference network $q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)$ will later induce the critique representation. Finally, one might be tempted to learn α jointly with the variational lower bound or dynamically. However, doing so might miscalibrate the precisions of the $q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)$ or $q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)$ and thus be detrimental to the whole model in terms of both prediction performance and generalization.

3.2 Training Strategy

Combining Equations 5 and 6 gives the full objective function, and M&Ms-VAE can be trained on a complete dataset where all $\boldsymbol{r}_u$ and $\boldsymbol{k}_u$ are provided. However, in doing so, we never train the individual inference networks $q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)$ and $q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)$; only the relationship between the observed user interactions and keyphrase-usage preferences is captured. As a consequence, at inference, it is unclear how the model performs with a missing observation.

To reach our goal of recommending given at least $\boldsymbol{r}_u$ and embedding the critique effectively with the inference network

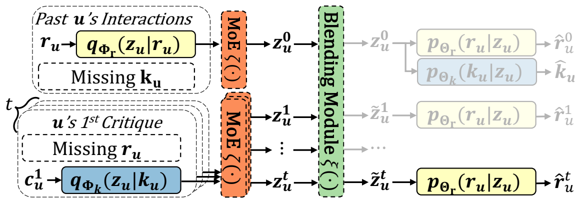


Fig. 3. Workflow of considering the recommendation of items to a user u over $t+1$ time steps. First, M&Ms-VAE produces the initial set of recommended items $\hat{\boldsymbol{r}}_u^0 = \hat{\boldsymbol{r}}_u$ using only the historical observed interactions $\boldsymbol{r}_u$. Then, the user can provide a critique $\boldsymbol{c}_u^t$ that is encoded into $\boldsymbol{z}_u^t$ via the inference model $q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)$. The blending module combines the previous representations $\boldsymbol{z}_u^0, \boldsymbol{z}_u^1, \ldots, \boldsymbol{z}_u^t$ into $\tilde{\boldsymbol{z}}_u^t$, from which the subsequent recommendation $\hat{\boldsymbol{r}}_u^t$ is computed. This process continues until the user u accepts the recommendation and ceases to provide additional critiques.

Algorithm 1 Synthetic Critiquing Dataset Creation

```
function GENERATE(R^val, K^I)
    Synthetic dataset D ← {}
    for each user u do
        for each target item i, where r^val_{u,i} = 1 do
            Randomly sample a critique c ∈ K\k_i^I
            Compute the item sets I^{+c} and I^{-c}
            Update D ← D ∪ {(u, i, c, I^{+c}, I^{-c})}
    return Synthetic dataset D
```

$q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)$, we propose a training strategy that mimics weakly supervised learning, similarly to [47]. Moreover, this allows us to handle incomplete datasets, where some samples are partially observed: data that contain only $\boldsymbol{r}_u$ or $\boldsymbol{k}_u$.[2] The training strategy is shown in Figure 2. For each minibatch, we compute the gradient on the evidence lower bound of the joint observation and each single observation $\boldsymbol{r}_u$ and $\boldsymbol{k}_u$. Our final training objective for all users u is

$$\begin{aligned}
\mathcal{L}(\boldsymbol{R},\boldsymbol{K}) = &\sum_{u\in U} \lambda \cdot \mathbb{E}_{q_\Phi(z_u|\boldsymbol{r}_u,\boldsymbol{k}_u)}\left[\log p_{\Theta_r}(\boldsymbol{r}_u|\boldsymbol{z}_u) + \log p_{\Theta_k}(\boldsymbol{k}_u|\boldsymbol{z}_u)\right] - \beta\, \mathrm{D_{KL}}\left[q_\Phi(\boldsymbol{z}_u|\boldsymbol{r}_u,\boldsymbol{k}_u)\;||\;p(\boldsymbol{z}_u)\right] && \mathit{ELBO}(\boldsymbol{r}_u,\boldsymbol{k}_u)\\
&+ \sum_{u\in U} \lambda \cdot \mathbb{E}_{q_{\Phi_r}(z_u|\boldsymbol{r}_u)}\left[\log p_{\Theta_r}(\boldsymbol{r}_u|\boldsymbol{z}_u)\right] - \beta\, \mathrm{D_{KL}}\left[q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)\;||\;p(\boldsymbol{z}_u)\right] && \mathit{ELBO}(\boldsymbol{r}_u)\\
&+ \sum_{u\in U} \lambda \cdot \mathbb{E}_{q_{\Phi_k}(z_u|\boldsymbol{k}_u)}\left[\log p_{\Theta_k}(\boldsymbol{k}_u|\boldsymbol{z}_u)\right] - \beta\, \mathrm{D_{KL}}\left[q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)\;||\;p(\boldsymbol{z}_u)\right], && \mathit{ELBO}(\boldsymbol{k}_u) \qquad (7)
\end{aligned}$$

where λ and β control the strength of the reconstruction error and regularization, respectively.

3.3 Self-Supervised Critiquing with M&Ms-VAE

The purpose of critiquing is to refine the recommendation $\hat{\boldsymbol{r}}_u$ based on the user u's interaction with the explanation $\hat{\boldsymbol{k}}_u$, represented with a binary vector. The user can accept the recommended items, at which point the session terminates. In the other case, the user can provide a critique $\boldsymbol{c}_u^t$ and obtain a new recommendation $\hat{\boldsymbol{r}}_u^t$. The process is repeated over T iterations until the user u is satisfied with the recommendation. Each critique $\boldsymbol{c}_u^t$ is encoded as a one-hot vector where the positive value indicates a keyphrase the user u dislikes. The overall process is depicted in Figure 3.

We leverage the generalization ability of the trained M&Ms-VAE, especially the inference models $q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)$ and $q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)$. We use the former to represent the initial user preferences $\boldsymbol{r}_u$ and the latter to embed the critique $\boldsymbol{c}_u^t$. However, a crucial question remains: how should we blend the user representation $\boldsymbol{z}_u$ with the t^{th} critique representation $\boldsymbol{z}_u^t$?

Prior work has implemented a blending function as a simple average [26, 45] or as a linear programming task that looks for a convex combination of embeddings provided with a specific linear optimization objective [22, 25]. As we demonstrate empirically later, the former yields poor performance when iterated for multi-step critiquing, whereas the latter is computationally slow because the optimization is performed for each critique and it cannot leverage GPUs.

[2]It also enables another way to solve the cold-start problem: new users can select a set of items and/or relevant keyphrases that reflect their preferences.

3.3.1 Blending Function Design. We propose to learn a blending function $\xi(\cdot)$ built upon a trained M&Ms-VAE model whose weights are frozen. This two-step approach has several advantages:

(1) The original training of M&Ms-VAE is not perturbed by the critiquing objective.
(2) $\xi(\cdot)$ is decoupled from the model. It allows more flexibility in its architecture, objective function, and training.

We assume that each critique is independent, as in [2, 25, 45]. At time step t, we express the new user preferences $\tilde{z}_u^t$ with a linear interpolation between the original latent representation z_u^0 and the critique c_u^t's representation z_u^t. More precisely, we use the gating mechanism of the gated recurrent unit [10] (we omit the biases to reduce notational clutter):

$$\tilde{z}_u^t = \xi(z_u^0, z_u^t) = h_2 \qquad \begin{aligned} h_2 &= (1-u_1)\cdot n_1 + u_1 \cdot h_1 \\ n_1 &= \tanh(W_{in} z_u^t + W_{hn}(r_1 \odot h_1)) \\ u_1 &= \sigma(W_{iu} z_u^t + W_{hz} h_1) \\ r_1 &= \sigma(W_{ir} z_u^t + W_{hr} h_1) \end{aligned} \qquad \begin{aligned} h_1 &= (1-u_0)\cdot n_0 + u_0 \cdot h_0 \\ n_0 &= \tanh(W_{in} z_u^0 + W_{hn}(r_0 \odot h_0)) \\ u_0 &= \sigma(W_{iu} z_u^0 + W_{hz} h_0) \\ r_0 &= \sigma(W_{ir} z_u^0 + W_{hr} h_0) \end{aligned}, \tag{8}$$

where $W_{ir}, W_{iu}, W_{in}, W_{hr}, W_{hz}, W_{hn}$, and the bias vectors are the model parameters.

3.3.2 Training. Thanks to our assumption and the generalization ability of M&Ms-VAE, we can learn the weights of the blending module $\xi(\cdot)$ by creating a synthetic dataset based only on the validation set (see Algorithm 1). For each user and observed interaction, we randomly sample a keyphrase c that is inconsistent with the target item. Then, we calculate the item sets I^{+c} that contain the critique and, symmetrically, the item sets I^{-c} for those that do not contain it.[3]

Our final objective is to re-rank items based on the user preferences and the provided critique c. Recall that M&Ms-VAE's weights are frozen. Let $\hat{r}_u^0$ be the user u's initial predictions $\hat{r}_u$ and $\hat{r}_u^1$ those inferred from $\tilde{z}_u^t$ after the critique. We express this overall ranking-based objective via two differentiable max-margin objective functions:

$$\mathcal{L}(\hat{R}^0, \hat{R}^1, u, c, I^{+c}, I^{-c}) = \sum_{i^+ \in I^{+c}} \left[\max\left\{0, h - (\hat{r}^0_{u,i^+} - \hat{r}^1_{u,i^+})\right\}\right] + \sum_{i^- \in I^{-c}} \left[\max\left\{0, h - (\hat{r}^1_{u,i^-} - \hat{r}^0_{u,i^-})\right\}\right], \tag{9}$$

where h is the margin. Intuitively, $\xi(\cdot)$ is encouraged to create a representation $\tilde{z}_u^t$ from which $p_{\Theta_r}(\cdot)$ gives a lower ranking to the items affected by the critique in the next iteration (i.e., $\hat{r}^1_{u,i^+} < \hat{r}^0_{u,i^+}$) and a higher ranking to the unaffected items (i.e., $\hat{r}^1_{u,i^-} > \hat{r}^0_{u,i^-}$). Finally, Equation 9 is efficiently parallelizable on both CPUs and GPUs.

4 EXPERIMENTS

In this section, we proceed to evaluate the proposed M&Ms-VAE model in order to answer the following questions:

- **RQ 1**: How does M&Ms-VAE perform in terms of recommendation and explanation performance?
- **RQ 2**: Can M&Ms-VAE with the self-supervised critiquing objective enable multi-step critiquing?
- **RQ 3**: What is our proposed critiquing algorithm's computational time complexity compared to prior work?
- **RQ 4**: How does M&Ms-VAE perform under weak supervision; how coherent is the joint and cross generation?

4.1 Datasets

We evaluate the quantitative performance of M&Ms-VAE using four real-world, publicly available datasets: BeerAdvocate [27], Amazon CDs&Vinyl [14, 28], Yelp [12], and HotelRec [1]. Each contains more than 100k reviews with five-star ratings. For the purpose of Top-N recommendation, we binarize the ratings with a threshold $t > 3.5$. Because people tend to rate beers and restaurants positively, we set the threshold $t > 4$ and $t > 4.5$, respectively. We split each dataset

[3] In early experiments, we generalized $\xi(z_u^0, z_u^t)$ to $\xi(z_u^0, z_u^1, \ldots, z_u^t)$ and updated Algorithm 1 accordingly. However, our synthetic dataset cannot cover such a space due to the exponential number of combinations; session-based recommenders require millions of real sessions as training data [16, 17].

Table 1. Descriptive statistics of the datasets. Coverage shows the ratio of reviews having at least one of the selected keyphrases (KPs)

Dataset	#Users	#Items	#Interactions	Sparsity	#KPs	KP Coverage	Avg. KPs/Review	AVG. KPs/User
Beer	6,370	3,669	263,244	1.13%	75	99.27%	7.16	1,216
CDs&Vinyl	6,060	4,395	152,783	0.57%	40	74.59%	2.13	73
Yelp	9,801	4,706	140,496	0.30%	234	96.65%	7.45	300
Hotel	7,044	4,874	143,612	0.42%	141	99.99%	17.42	419

into 60%/20%/20% for the training, validation, and test sets. Table 1 shows the statistics of the datasets. All contain complete observations. The datasets do not contain preselected keyphrases. Hence, we extract the keyphrases for the explanations and critiquing with the frequency-based processing of [22, 45]. Some examples are shown in the appendix.

4.2 Experimental Settings

Across experiments, we treat the prior and the likelihood as standard normal and multinomial distributions, respectively. The inference and generative networks consist of a two-layer neural network with a tanh nonlinearity as the activation function between the layers. We normalize the input and use dropout [38]. For learning, we employ the Adam optimizer [19] with AMSGrad [30] and a learning rate of $5 \cdot 10^{-5}$. We anneal linearly the regularization parameter β of the Kullback-Leibler terms. For the baselines, we reused the authors' code and tuning procedure. We select hyperparameters and architectures for each model by evaluating NDCG on the validation set. We limit the search to a maximum of 100 trials. For critiquing, we tune our blending module on the synthetic dataset with the Falling MAP metric on the validation set, which measures the effect of a critique [45]. For reproducibility purposes, we include additional details and the best hyperparameters in the supplementary material.

4.3 RQ 1: How does M&Ms-VAE perform in terms of recommendation and explanation performance?

4.3.1 Baselines. We compare our proposed M&Ms-VAE model to the following baseline models. **POP** returns the most popular items without any kind of personalization. **AutoRec** [36] is a neural autoencoder-based recommendation system. **BPR** [32] is a Bayesian personalized ranking model that explicitly optimizes pairwise rankings. **CDAE** [48] denotes a collaborative denoising autoencoder that is specifically optimized for implicit feedback recommendation tasks. **NCE-PLRec** [46] represents the linear recommendation projected by noise-contrastive estimation; it augments PLRec with noise-contrasted item embeddings. **PLRec** [35] is the ablation variant of NCE-PLRec without the noise-contrastive estimation. **PureSVD** [11] denotes a similarity-based recommendation method that constructs a similarity matrix through SVD decomposition of the implicit rating matrix. **VAE-CF** is the variational autoencoder for collaborative filtering described in Section 2.2. **CE-VNCF** [45] is the extension of the neural collaborative filtering model [15] that is augmented with an explanation and a critiquing neural component. Finally, **CE-VAE** [26] is a significant improvement over CE-VNCF, and it produces state-of-the-art performance (more details in Section 2.3). For a fair comparison, we encode the user observations in M&Ms-VAE using solely the inference network $q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)$ at test time.

4.3.2 Top-N Recommendation Performance. We report the following five metrics: R-Precision and NDCG; MAP, Precision, and Recall at different Top-N. The main results are presented in Table 2. We make the following key observations. Overall, our proposed M&Ms-VAE model shows the best recommendation performance for all metrics on three datasets and nearly all metrics on the CDs&Vinyl dataset. Compared to the original VAE recommender (VAE-CF), M&Ms-VAE achieves an improvement of 13% on average. We conjecture that the additional loss terms (i.e., $ELBO(\boldsymbol{r}_u, \boldsymbol{k}_u)$ and

Table 2. Top-N recommendation results of all datasets. **Bold** and underline denote the best and second-best results, respectively. We omit the error bars because the 95% confidence interval is in 4th digit.

				MAP@N			Precision@N			Recall@N		
	Model	**R-Precision**	**NDCG**	$N=5$	$N=10$	$N=20$	$N=5$	$N=10$	$N=20$	$N=5$	$N=10$	$N=20$
Beer	POP	0.0307	0.0777	0.0388	0.0350	0.0319	0.0346	0.0298	0.0279	0.0241	0.0408	0.0737
	AutoRec	0.0496	0.1140	0.0652	0.0591	0.0527	0.0574	0.0503	0.0438	0.0392	0.0663	0.1129
	BPR	0.0520	0.1214	0.0646	0.0597	0.0538	0.0596	0.0525	0.0449	0.0451	0.0744	0.1214
	CDAE	0.0414	0.0982	0.0504	0.0477	0.0434	0.0482	0.0432	0.0368	0.0330	0.0576	0.0969
	NCE-PLRec	0.0501	0.1151	0.0643	0.0594	0.0532	0.0589	0.0518	0.0440	0.0418	0.0714	0.1177
	PLRec	0.0497	0.1113	0.0655	0.0599	0.0532	0.0590	0.0515	0.0431	0.0421	0.0704	0.1127
	PureSVD	0.0450	0.1052	0.0479	0.0473	0.0446	0.0493	0.0455	0.0396	0.0391	0.0689	0.1131
	VAE-CF	0.0538	0.1275	0.0642	0.0594	0.0536	0.0595	0.0525	0.0448	0.0473	0.0808	0.1327
	CE-VAE	0.0520	0.1215	0.0675	0.0618	0.0555	0.0620	0.0536	0.0461	0.0442	0.0737	0.1255
	CE-VNCF	0.0440	0.1099	0.0546	0.0512	0.0472	0.0504	0.0465	0.0411	0.0353	0.0635	0.1116
	M&Ms-VAE (Ours)	**0.0545**	**0.1307**	**0.0706**	**0.0650**	**0.0580**	**0.0649**	**0.0563**	**0.0473**	**0.0492**	**0.0833**	**0.1349**
CDs&Vinyl	POP	0.0088	0.0265	0.0108	0.0102	0.0095	0.0098	0.0095	0.0082	0.0088	0.0182	0.0327
	AutoRec	0.0227	0.0537	0.0284	0.0257	0.0220	0.0255	0.0213	0.0165	0.0254	0.0418	0.0627
	BPR	0.0632	0.1516	0.0724	0.0639	0.0543	0.0640	0.0513	0.0408	0.0807	0.1263	0.1939
	CDAE	0.0135	0.0365	0.0173	0.0158	0.0141	0.0152	0.0136	0.0116	0.0143	0.0262	0.0451
	NCE-PLRec	0.0749	0.1739	0.0728	0.0678	0.0586	0.0698	0.0584	0.0441	**0.1010**	**0.1608**	0.2308
	PLRec	0.0760	0.1626	**0.0889**	0.0777	0.0642	0.0773	0.0608	0.0444	0.0960	0.1461	0.2025
	PureSVD	0.0652	0.1551	0.0570	0.0565	0.0509	0.0612	0.0527	0.0405	0.0914	0.1486	0.2149
	VAE-CF	0.0638	0.1699	0.0540	0.0554	0.0517	0.0600	0.0540	0.0440	0.0949	0.1593	**0.2381**
	CE-VAE	0.0708	0.1532	0.0816	0.0711	0.0588	0.0715	0.0555	0.0411	0.0903	0.1357	0.1937
	CE-VNCF	0.0654	0.1524	0.0746	0.0662	0.0560	0.0663	0.0534	0.0411	0.0829	0.1299	0.1931
	M&Ms-VAE (Ours)	**0.0801**	**0.1765**	0.0885	**0.0784**	**0.0660**	**0.0779**	**0.0628**	**0.0482**	0.0983	0.1529	0.2263
Yelp	POP	0.0026	0.0129	0.0024	0.0026	0.0026	0.0028	0.0028	0.0025	0.0042	0.0087	0.0151
	AutoRec	0.0034	0.0133	0.0032	0.0030	0.0028	0.0027	0.0027	0.0025	0.0038	0.0081	0.0153
	BPR	0.0160	0.0609	0.0168	0.0156	0.0143	0.0156	0.0140	0.0122	0.0236	0.0435	0.0748
	CDAE	0.0028	0.0135	0.0027	0.0028	0.0027	0.0030	0.0027	0.0026	0.0044	0.0084	0.0161
	NCE-PLRec	0.0197	0.0739	0.0220	0.0200	0.0177	0.0198	0.0169	0.0143	0.0300	0.0505	0.0864
	PLRec	0.0191	0.0703	0.0207	0.0189	0.0171	0.0185	0.0166	0.0143	0.0291	0.0513	0.0866
	PureSVD	0.0253	0.0825	0.0279	0.0249	0.0217	0.0240	0.0206	0.0173	0.0357	0.0597	0.1008
	VAE-CF	0.0214	0.0801	0.0232	0.0216	0.0195	0.0214	0.0192	0.0163	0.0319	0.0589	0.0995
	CE-VAE	0.0136	0.0533	0.0143	0.0132	0.0121	0.0125	0.0119	0.0104	0.0197	0.0367	0.0636
	CE-VNCF	0.0166	0.0693	0.0175	0.0167	0.0157	0.0165	0.0154	0.0144	0.0251	0.0467	0.0889
	M&Ms-VAE (Ours)	**0.0264**	**0.0909**	**0.0284**	**0.0261**	**0.0231**	**0.0260**	**0.0223**	**0.0188**	**0.0395**	**0.0682**	**0.1154**
Hotel	POP	0.0047	0.0188	0.0049	0.0047	0.0042	0.0047	0.0043	0.0036	0.0054	0.0098	0.0167
	AutoRec	0.0051	0.0193	0.0053	0.0050	0.0044	0.0052	0.0042	0.0037	0.0061	0.0097	0.0169
	BPR	0.0181	0.0623	0.0198	0.0185	0.0169	0.0183	0.0168	0.0146	0.0219	0.0409	0.0713
	CDAE	0.0050	0.0190	0.0054	0.0049	0.0044	0.0050	0.0043	0.0037	0.0057	0.0098	0.0172
	NCE-PLRec	0.0229	0.0684	0.0244	0.0226	0.0200	0.0228	0.0195	0.0160	0.0283	0.0484	0.0785
	PLRec	0.0242	0.0664	0.0265	0.0234	0.0201	0.0228	0.0190	0.0155	0.0284	0.0466	0.0758
	PureSVD	0.0179	0.0541	0.0193	0.0173	0.0152	0.0169	0.0145	0.0121	0.0206	0.0357	0.0594
	VAE-CF	0.0243	0.0755	0.0267	0.0241	0.0213	0.0238	0.0206	0.0171	0.0295	0.0511	0.0848
	CE-VAE	0.0147	0.0538	0.0151	0.0146	0.0136	0.0148	0.0137	0.0122	0.0184	0.0334	0.0595
	CE-VNCF	0.0165	0.0575	0.0180	0.0166	0.0152	0.0159	0.0149	0.0129	0.0200	0.0370	0.0635
	M&Ms-VAE (Ours)	**0.0272**	**0.0804**	**0.0290**	**0.0265**	**0.0235**	**0.0260**	**0.0227**	**0.0189**	**0.0314**	**0.0555**	**0.0928**

$ELBO(\boldsymbol{k}_u)$) help to generate better user representations by leveraging both the user preferences and keyphrase usage with the mixture of experts.

Table 3. Top-K keyphrase explanation quality of all datasets. **Bold** and underline denote the best and second-best results, respectively. We omit the error bars because the 95% confidence interval is in 4th digit.

		NDCG@K			**MAP@K**			**Precision@K**			**Recall@K**		
	Model	$K=5$	$K=10$	$K=20$	$K=5$	$K=10$	$K=20$	$K=5$	$K=10$	$K=20$	$K=5$	$K=10$	$K=20$
Beer	UserPop	0.0550	0.0852	0.1484	0.0812	0.0795	0.0824	0.0782	0.0799	0.0933	0.0446	0.0913	0.2186
	ItemPop	0.0511	0.0807	0.1428	0.0767	0.0749	0.0777	0.0697	0.0740	0.0895	0.0402	0.0856	0.2107
	CE-VAE	<u>0.2803</u>	<u>0.4104</u>	<u>0.5916</u>	<u>0.9418</u>	<u>0.9168</u>	<u>0.8820</u>	<u>0.9186</u>	<u>0.8760</u>	<u>0.8186</u>	0.1448	0.2683	0.4829
	CE-VNCF	0.2390	0.3221	0.4080	0.3414	0.3117	0.2690	0.3145	0.2633	0.2026	**0.1966**	**0.3263**	**0.4962**
	M&Ms-VAE (Ours)	**0.2817**	**0.4147**	**0.5960**	**0.9463**	**0.9237**	**0.8885**	**0.9243**	**0.8861**	**0.8256**	0.1457	0.2722	<u>0.4869</u>
CDs&Vinyl	UserPop	0.1028	0.1285	0.1869	0.0910	0.0807	0.0700	0.0860	0.0636	0.0595	0.1157	0.1704	0.3392
	ItemPop	0.1109	0.1357	0.1935	0.0928	0.0833	0.0716	0.0929	0.0657	0.0606	0.1288	0.1825	0.3499
	CE-VAE	<u>0.5243</u>	<u>0.6468</u>	<u>0.7454</u>	<u>0.6441</u>	<u>0.5609</u>	<u>0.4575</u>	<u>0.5564</u>	<u>0.4324</u>	<u>0.3040</u>	0.4795	<u>0.6808</u>	<u>0.8757</u>
	CE-VNCF	0.4590	0.5338	0.5860	0.3657	0.2981	0.2258	0.2893	0.2010	0.1260	**0.5081**	0.6698	0.8127
	M&Ms-VAE (Ours)	**0.5447**	**0.6659**	**0.7628**	**0.6648**	**0.5779**	**0.4700**	**0.5777**	**0.4441**	**0.3091**	<u>0.5015</u>	**0.6996**	**0.8894**
Yelp	UserPop	0.0007	0.0009	0.0066	0.0009	0.0010	0.0016	0.0013	0.0009	0.0061	0.0007	0.0011	0.0129
	ItemPop	0.0008	0.0011	0.0073	0.0009	0.0011	0.0018	0.0015	0.0011	0.0065	0.0009	0.0013	0.0149
	CE-VAE	<u>0.1935</u>	<u>0.2763</u>	<u>0.3803</u>	<u>0.6653</u>	<u>0.6356</u>	<u>0.5916</u>	<u>0.6363</u>	<u>0.5876</u>	<u>0.5181</u>	<u>0.1017</u>	<u>0.1819</u>	<u>0.3080</u>
	CE-VNCF	0.0883	0.1164	0.1505	0.1195	0.1052	0.0901	0.1023	0.0848	0.0690	0.0779	0.1270	0.2027
	M&Ms-VAE (Ours)	**0.1949**	**0.2787**	**0.3837**	**0.6738**	**0.6428**	**0.5976**	**0.6434**	**0.5935**	**0.5229**	**0.1019**	**0.1834**	**0.3107**
Hotel	UserPop	0.0436	0.0681	0.1059	0.1091	0.1059	0.1054	0.1018	0.1036	0.1050	0.0265	0.0557	0.1120
	ItemPop	0.0483	0.0756	0.1152	0.1159	0.1137	0.1124	0.1108	0.1131	0.1111	0.0303	0.0633	0.1237
	CE-VAE	<u>0.2425</u>	<u>0.3521</u>	<u>0.4984</u>	<u>0.9389</u>	<u>0.9113</u>	<u>0.8638</u>	<u>0.9209</u>	<u>0.8629</u>	<u>0.7831</u>	0.1153	<u>0.2105</u>	<u>0.3704</u>
	CE-VNCF	0.1873	0.2527	0.3280	0.3975	0.3671	0.3230	0.3732	0.3154	0.2558	**0.1252**	0.2060	0.3230
	M&Ms-VAE (Ours)	**0.2500**	**0.3595**	**0.5054**	**0.9752**	**0.9393**	**0.8829**	**0.9498**	**0.8776**	**0.7895**	<u>0.1182</u>	**0.2131**	**0.3726**

M&Ms-VAE also significantly outperforms CE-VAE on the Yelp and Hotel datasets (by a factor of 1.9 and 1.7, respectively) and achieves an average improvement of 9% on the Beer and CDs&Vinyl datasets. We remark the same trend with CE-VNCF. These results emphasize the noise introduced in CE-VAE and CE-VNCF during training when learning the mapping between the keyphrases and the latent space. This is even more pronounced with a large number of keyphrases (i.e., over 100). In contrast, M&Ms-VAE is more robust thanks to our factorization and training strategy.

Interestingly, PureSVD exhibits the second-best performance on the CDs&Vinyl and Yelp datasets. This shows that classic algorithms often remain competitive with state-of-the-art VAE-based recommender systems.

4.3.3 Top-K Explanation Performance. We also compare M&Ms-VAE with the user and item-popularity baselines [45] that predict the explanation through counting and ranking the frequency of keyphrases for the users (symmetrically, the items) in the training set. Among the recommender baselines, only CE-VAE and CE-VNCF produce an explanation alongside the recommendation. We report the following metrics: NDCG, MAP, Precision, and Recall at different Top-K.

Table 3 contains the main results. Both popularity baselines clearly underperform, showing that the task is not trivial (see Table 1 for the number of keyphrases per dataset). Remarkably, the proposed M&Ms-VAE model significantly outperforms the CE-VNCF baseline by a factor of 2.5 on average and by approximately by 3.5 on MAP and Precision. Moreover, M&Ms-VAE retrieves 89% percent of relevant keyphrases within the Top-20 explanations on the CDs&Vinyl dataset.

We observe that CE-VAE performs similarly to M&Ms-VAE but still slightly underperforms by approximately 2% on average. Finally, we note that CE-VNCF achieves the best results in terms of Recall for the Beer dataset and Recall@5 for the CDs&Vinyl and Hotel datasets. Nevertheless, as seen in the recommendation performance, CE-VNCF significantly underperforms, highlighting the trade-off between recommendation and explanation.

4.4 RQ 2: Can M&Ms-VAE with the self-supervised critiquing objective enable multi-step critiquing?

4.4.1 Baselines. **UAC** [25] denotes uniform average critiquing, in which the user embedding and all critique embeddings are averaged uniformly. **BAC** [25] is balanced average critiquing. It first averages the critique embeddings and then averages them again with the initial user embedding. **CE-VAE** and **CE-VNCF** were introduced in Section 4.3.1. During training, both learn an inverse feedback loop between a critique and the latent space. At inference, they average the original user embedding with the critique embedding. **LLC-Score** [25] and **LLC-Rank** [22] first extend the PLRec recommender system [35] to co-embed keyphrases into the user embedding space with a linear regression. Afterwards, the models apply a weighted average between the initial user embedding and each critique embedding; the weights are optimized in a linear programming formulation; LLC-Score uses a max-margin scoring-based objective and LLC-Rank a ranking-based objective. To limit computational complexity, the authors limit the number of constraints to the top-100 rated items. For a fair comparison, we also consider in M&Ms-VAE the top-100 rated items meeting the criteria for $I^{+\boldsymbol{c}}$ and $I^{-\boldsymbol{c}}$ for each critique $\boldsymbol{c}$, although the computational time remains identical using the full sets.

4.4.2 User Simulation. Similarly to prior work [22, 25], we conduct a user simulation to asses each model's performance in a multi-step conversational recommendation scenario. Concretely, the simulation considers all users and follows Algorithm 1 with the following differences: (1) we track the conversational interaction session of simulated users by selecting all target items from their **test** set, (2) the maximum allowed critiquing iterations is set to 10, and, (3) the conversation stops if the target item appears within the top-N recommendations on that iteration.

As in [22], we simulate a variety of user-critiquing styles. For each, the candidate keyphrases to critique are inconsistent (i.e., disjoint) with the target item's known keyphrase list. We experiment with the following three methodologies:

(1) **Random**: we assume the user randomly chooses a keyphrase to critique.
(2) **Pop**: we assume the user selects a keyphrase to critique according to the general keyphrase popularity.
(3) **Diff**: we assume the user critiques a keyphrase that deviates the most from the known target item description. To do so, we compare the top recommended items' keyphrase frequency to the target item's keyphrases and select the keyphrase with the largest frequency differential.

4.4.3 Multi-Step Critiquing Performance. Following [22, 25], we asses the models over all users and all items on the test set using two metrics: the average success rate and session length. The former is the percentage of target items that successfully reach a rank within the Top-N, and the latter is the average length of these sessions (with a limit of 10 iterations). In our experiment framework, for each user and target item, we sample alongside 299 unseen items.

The results for each dataset and each keyphrase critiquing selection method are depicted in Figure 4. Overall, all models' performance is generally better on the Beer and CDs&Vinyl datasets due to the higher density in terms of the number of interactions. Generally, all models tend to find the target item within the Top-20 in more than half the time and under six turns. This highlights that in practice, users are likely to find the desired item in a limited amount of time.

Impressively, M&Ms-VAE clearly outperforms all the baselines on both metrics on the Beer, CDs&Vinyl, and Yelp datasets. On the Hotel dataset, the success rate is significantly higher for the Random and Pop cases and similar for the Diff case, whereas the session length is higher for the Pop and Diff selection methods. This is unsurprising, because the blending module is trained only once on the random keyphrase critiquing selection (i.e., no assumption on the user's behavior).

Although the simple self-supervision objective in M&Ms-VAE mimics only one-step random critiquing, we remark

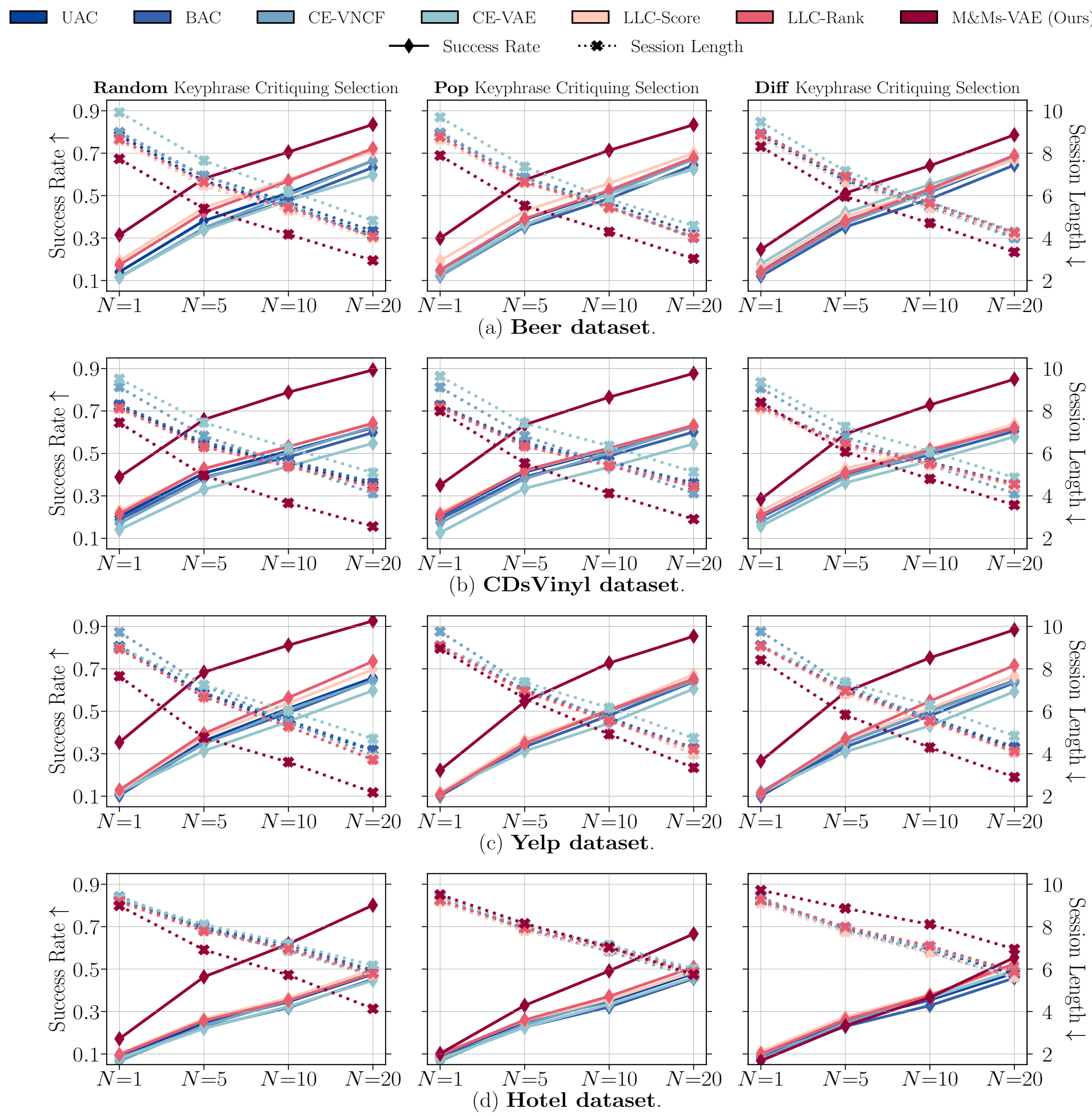
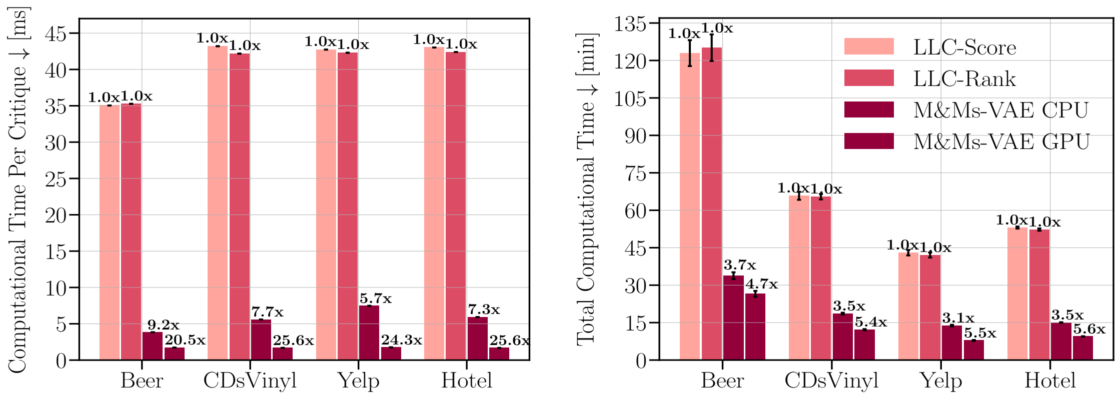


Fig. 4. Multi-step critiquing performance after 10 turns of conversation. For each dataset and keyphrase critiquing selection method, we report the average success rate (left y-axis) and session length (right y-axis) at different Top-N with 95% confidence interval.

that the training strategy generalizes for multi-step critiquing and other keyphrase critiquing selection as well. This shows that M&Ms-VAE efficiently embeds the critique, thanks to the multimodal modeling.

We observe that the simple UAC and BAC methods perform similarly or better than CE-VNCF and CE-VAE. However, they are outperformed by LLC-Score, LLC-Rank, and M&Ms-VAE. These results confirm our observation in Section 2.3 that the critiquing objective introduces noise during training and does not accurately reflect the critiquing mechanism.

Finally, we note that LLC-Score performs similarly to LLC-Rank in most cases. When compared to M&Ms-VAE, both



(a) Average computational time of a single critique.

(b) Average total computational time of the simulation.

Fig. 5. Average time consumed for completing 10 runs for 1,000-user simulation after ten turns of conversation with 95% confidence intervals. LLC-Score and LLC-Rank cannot leverage GPUs; we thus report the performance of M&Ms-VAE on CPU and GPU. Additionally, we report the inference speedup compared to the slowest model.

significantly underperform on both metrics in 10 out of 12 cases. This highlights the effectiveness of our proposed critiquing algorithm compared to linear aggregation methods.

4.5 RQ 3: What is the critiquing computational time complexity for M&Ms-VAE compared to prior work?

Now, we aim to empirically determine how the critiquing in M&Ms-VAE compares to the best baselines in Section 4.4, LLC-Score and LLC-Rank, in terms of computational time. Because the baselines can not leverage the GPU due to their optimization framework, we also run M&Ms-VAE on the CPU. We follow the same experiment settings as in Section 4.4 and limit ourselves to 1,000 users and the Random keyphrase critiquing selection method. All models process exactly 10 critiques for each user-item pair. We employ a machine with a 2.5GHz 24-core CPU, Titan X GPU, and 256GB memory.

Figure 5 shows the average computational time over 10 runs. Particularly, the Figure 5a denotes the critiquing computational time in milliseconds, and Figure 5b the overall simulation time in minutes. Impressively, we observe that the critiquing in M&Ms-VAE's is approximately 7.5x faster on CPU and up to 25.6x on the GPU than LLC-Score and LLC-Rank. This shows that once the critiquing module of M&Ms-VAE is trained, which takes less than five minutes on the machine, the model achieves a lower latency (batch size of one). In Figure 5b, the overall simulation in M&Ms-VAE is at least 3.1x faster on CPU and approximately 5.3x faster on GPU. Finally, in real-life applications, we could leverage multiple users' critiques simultaneously and increase the throughput by considering a larger batch size.

4.6 RQ4: How does M&Ms-VAE perform under weak supervision; is the joint & cross inference coherent?

We first quantify the coherence of joint and cross generations of our M&Ms-VAE model. We denote three cases at test time: (1) only the user's interactions are used, and the encoder is $q_{\Phi_r}(\boldsymbol{z}_u|\boldsymbol{r}_u)$; (2) only the user's keyphrase preference is used, and the encoder is $q_{\Phi_k}(\boldsymbol{z}_u|\boldsymbol{k}_u)$; and (3) both used with the encoder $q_{\Phi}(\boldsymbol{z}_u|\boldsymbol{r}_u, \boldsymbol{k}_u)$. Second, we simulate incomplete supervision by randomly selecting a subset of the training with fully observed samples. The other one is split into two even parts: the first includes only observed $\boldsymbol{r}_u$ and the second $\boldsymbol{k}_u$. We retain the models and settings of Section 4.3.

Figure 6 shows the results averaged on five runs for the three cases at different levels of supervision. The top row presents the explanation and recommendation performance in terms of NDCG and Recall@20 (the model behavior

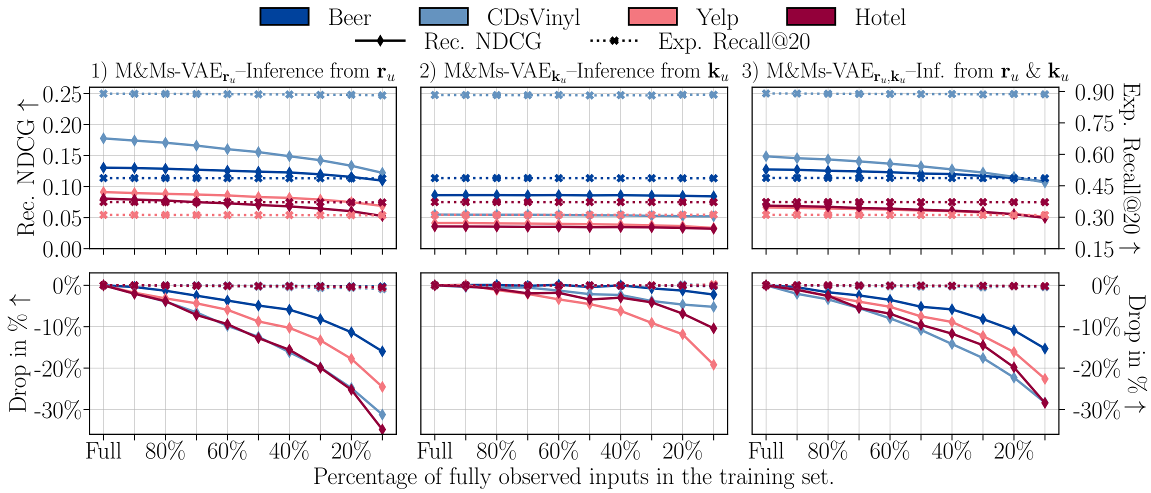


Fig. 6. Recommendation (left y-axis) and explanation (right y-axis) results averaged on 5 runs with different combinations of modalities observed at inference. The x-axis denotes the percentage of fully observed inputs during training; the rest is partially observed.

is consistent across the other metrics). On the full datasets, we note that the explanation performance is comparable across the three variants and higher than those of the baselines in Table 3. Regarding the recommendation performance, M&Ms-VAE$_{\mathbf{r}_u}$ obtains the best results, followed closely by M&Ms-VAE$_{\mathbf{r}_u,\mathbf{k}_u}$, and M&Ms-VAE$_{\mathbf{k}_u}$ clearly underperforms. This aligns with the observations in [7, 34]: recommender systems are limited if they use only review text as input, and not all reviews can be useful. Nevertheless, compared to Table 2, M&Ms-VAE$_{\mathbf{k}_u}$ always achieves better recommendation performance than the popularity baseline, and it performs better than AutoRec and CDAE on two datasets.

The bottom row of Figure 6 show the relative drop in performance on both metrics. The explanation performance seems unaffected by the sparsity, showing that the explanation task remains simple in comparison with the recommendation task. Remarkably, with only 50% fully observed inputs and the rest partially observed, the recommendation performance of M&Ms-VAE$_{\mathbf{r}_u}$ and M&Ms-VAE$_{\mathbf{r}_u,\mathbf{k}_u}$ is decreased by only 9% on average. More so, with 90% partial observations, the model can still achieve more than 70% of its performance quality on the full datasets. Finally, these results emphasize that M&Ms-VAE can effectively learn the joint distribution even in a weakly supervised setting.

5 CONCLUSION

Recommendations can have much more impact if they are supported by explanations that can be critiqued. Previous research has developed methods that either perform poorly in multi-step critiquing or suffer from computational inefficiency at inference. In this paper, we presented M&Ms-VAE, a novel variational autoencoder for recommendation and explanation that treats the user preference and keyphrase usage as different observed variables. Additionally, we proposed a strategy that mimics weakly supervised learning and trains the inference networks jointly and independently.

Our second contribution is a new critiquing module that leverages the generalizability of M&Ms-VAE to embed the user preference and the critique. With a self-supervised objective and a synthetic dataset, we enable multi-step critiquing in M&Ms-VAE. Experiments on four datasets show that M&Ms-VAE (1) is the first model to obtain substantially better recommendation, explanation, and multi-critiquing performance, (2) processes critiques up to 25.6x faster than previous state-of-the-art methods, and (3) produces coherent joint and cross generation, even under weak supervision.

REFERENCES

[1] Diego Antognini and Boi Faltings. 2020. HotelRec: a Novel Very Large-Scale Hotel Recommendation Dataset. In *Proceedings of The 12th Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 4917–4923.

[2] Diego Antognini, Claudiu Musat, and Boi Faltings. 2021. Interacting with Explanations through Critiquing. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conferences on Artificial Intelligence Organization. Main track.

[3] Vito Bellini, Angelo Schiavone, Tommaso Di Noia, Azzurra Ragone, and Eugenio Di Sciascio. 2018. Knowledge-aware autoencoders for explainable recommender systems. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*.

[4] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 10–21.

[5] Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. 1996. Knowledge-Based Navigation of Complex Information Spaces. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1 (AAAI'96)*. AAAI Press, 462–468.

[6] Shuo Chang, F Maxwell Harper, and Loren Gilbert Terveen. 2016. Crowd-based personalized natural language explanations for recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 175–182.

[7] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-Level Explanations. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, 1583–1592.

[8] Li Chen and Pearl Pu. 2012. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction* 22 (2012).

[9] Zhongxia Chen, Xiting Wang, Xing Xie, Mehul Parsana, Akshay Soni, Xiang Ao, and Enhong Chen. 2020. Towards Explainable Conversational Recommendation. In *Proceedings of the Twenty-Nineth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conferences on Artificial Intelligence Organization. Main track.

[10] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. http://arxiv.org/abs/1412.3555

[11] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (Barcelona, Spain) *(RecSys '10)*. Association for Computing Machinery, New York, NY, USA, 39–46. https://doi.org/10.1145/1864708.1864721

[12] Yelp Dataset. 2016. Yelp Open Dataset. https://www.yelp.com/dataset

[13] Boi Faltings, Pearl Pu, Marc Torrens, and Paolo Viappiani. 2004. Designing Example-critiquing Interaction. In *Proceedings of the 9th International Conference on Intelligent User Interfaces* (2004-01-01) *(IUI '04)*. ACM Press, Island of Madeira, Portugal, 22–29.

[14] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web* (Montréal, Québec, Canada) *(WWW '16)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 507–517. https://doi.org/10.1145/2872427.2883037

[15] 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*. 173–182.

[16] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-Based Recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 843–852. https://doi.org/10.1145/3269206.3271761

[17] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *the International Conference on Learning Representations (ICLR)*.

[18] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *the International Conference on Learning Representations (ICLR)*.

[19] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *the International Conference on Learning Representations (ICLR)*.

[20] Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *the International Conference on Learning Representations (ICLR)*.

[21] Johannes Kunkel, Tim Donkers, Catalin-Mihai Barbu, and Jürgen Ziegler. 2018. Trust-Related Effects of Expertise and Similarity Cues in Human-Generated Recommendations. (2018).

[22] Hanze Li, Scott Sanner, Kai Luo, and Ga Wu. 2020. A Ranking Optimization Approach to Latent Linear Critiquing for Conversational Recommender Systems. In *Fourteenth ACM Conference on Recommender Systems* (Virtual Event, Brazil) *(RecSys '20)*. Association for Computing Machinery, New York, NY, USA, 13–22. https://doi.org/10.1145/3383313.3412240

[23] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*. 689–698.

[24] Greg Linden, Steve Hanks, and Neal Lesh. 1997. Interactive Assessment of User Preference Models: The Automated Travel Assistant. In *In Proceedings of the Sixth International Conference on User Modeling*. Springer, 67–78.

[25] Kai Luo, Scott Sanner, Ga Wu, Hanze Li, and Hojin Yang. 2020. Latent Linear Critiquing for Conversational Recommender Systems. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2535–2541.

[26] Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. 2020. Deep Critiquing for VAE-Based Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1269–1278. https://doi.org/10.1145/3397271.3401091

[27] J. McAuley, J. Leskovec, and D. Jurafsky. 2012. Learning Attitudes and Attributes from Multi-aspect Reviews. In *2012 IEEE 12th International Conference on Data Mining*. 1020–1025. https://doi.org/10.1109/ICDM.2012.110

[28] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Santiago, Chile) *(SIGIR '15)*. Association for Computing Machinery, New York, NY, USA, 43–52. https://doi.org/10.1145/2766462.2767755

[29] Pearl Pu and Boi Faltings. 2000. Enriching Buyers' Experiences: The SmartClient Approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI '00)*. Association for Computing Machinery, New York, NY, USA, 289–296.

[30] Sashank Reddi, Satyen Kale, and Sanjiv Kumar. 2018. On the convergence of Adam and Beyond. In *International Conference on Learning Representations (ICLR)*.

[31] James Reilly, Jiyong Zhang, Lorraine McGinty, Pearl Pu, and Barry Smyth. 2007. Evaluating Compound Critiquing Recommenders: A Real-User Study. In *Proceedings of the 8th ACM Conference on Electronic Commerce* (San Diego, California, USA) *(EC '07)*. Association for Computing Machinery, New York, NY, USA, 114–123. https://doi.org/10.1145/1250910.1250929

[32] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. 452–461.

[33] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32 (ICML'14)*.

[34] Noveen Sachdeva and Julian McAuley. 2020. How Useful Are Reviews for Recommendation? A Critical Review and Potential Improvements. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1845–1848. https://doi.org/10.1145/3397271.3401281

[35] Suvash Sedhain, Hung Bui, Jaya Kawale, Nikos Vlassis, Branislav Kveton, Aditya Krishna Menon, Trung Bui, and Scott Sanner. 2016. Practical linear models for large-scale one-class collaborative filtering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*.

[36] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *Proceedings of the 24th International Conference on World Wide Web* (Florence, Italy) *(WWW '15 Companion)*. Association for Computing Machinery, New York, NY, USA, 111–112. https://doi.org/10.1145/2740908.2742726

[37] Yuge Shi, Siddharth N, Brooks Paige, and Philip Torr. 2019. Variational Mixture-of-Experts Autoencoders for Multi-Modal Deep Generative Models. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/0ae775a8cb3b499ad1fca944e6f5c836-Paper.pdf

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[39] Thomas Sutter, Imant Daunhawer, and Julia Vogt. 2020. Multimodal Generative Learning Utilizing Jensen-Shannon-Divergence. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 6100–6110. https://proceedings.neurips.cc/paper/2020/file/43bb733c1b62a5e374c63cb22fa457b4-Paper.pdf

[40] Nava Tintarev and Judith Masthoff. 2015. Explaining recommendations: Design and evaluation. In *Recommender systems handbook*. Springer, 353–382.

[41] Marc Torrens, Boi Faltings, and Pearl Pu. 2002. Smart Clients: Constraint Satisfaction as a Paradigm for Scaleable Intelligent Information Systems. *Constraints. Kluwer Academic Publishers, Hingham, MA, USA.* 7, 1 (2002), 49–69. https://doi.org/10.1023/A:1017940426216

[42] Yao-Hung Hubert Tsai, Paul Pu Liang, Amir Zadeh, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Learning Factorized Multimodal Representations. In *International Conference on Learning Representations (ICLR)*.

[43] Paolo Viappiani, Boi Faltings, and Pearl Pu. 2006. Preference-based search using example-critiquing with suggestions. *Journal of Artificial Intelligence Research (JAIR* (2006), 465–503.

[44] Michael D Williams and Frederich N Tou. 1982. RABBIT: an interface for database access. In *Proceedings of the ACM Conference*. 83–87.

[45] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep Language-Based Critiquing for Recommender Systems. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 137–145.

[46] Ga Wu, Maksims Volkovs, Chee Loong Soon, Scott Sanner, and Himanshu Rai. 2019. Noise Contrastive Estimation for One-Class Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 135–144. https://doi.org/10.1145/3331184.3331201

[47] Mike Wu and Noah Goodman. 2018. Multimodal Generative Models for Scalable Weakly-Supervised Learning. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc.

[48] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (San Francisco, California, USA) *(WSDM '16)*. Association for Computing Machinery, New York, NY, USA, 153–162. https://doi.org/10.1145/2835776.2835837

[49] Jingjing Zhang and Shawn P Curley. 2018. Exploring explanation effects on consumers' trust in online recommender agents. *International Journal of Human–Computer Interaction* 34, 5 (2018), 421–432.

A M&MS-VAE DERIVATION

$$\log\ p(\boldsymbol{r}_u, \boldsymbol{k}_u) = \log \int_{z_u} p_\Theta(\boldsymbol{r}_u, \boldsymbol{k}_u, z_u) dz_u \tag{10}$$

$$= \log \int_{z_u} p_\Theta(\boldsymbol{r}_u, \boldsymbol{k}_u, z_u) \frac{q_\Phi(z_u|\boldsymbol{r}_u, \boldsymbol{k}_u)}{q_\Phi(z_u|\boldsymbol{r}_u, \boldsymbol{k}_u)} dz_u \tag{11}$$

$$= \log\left(\mathbb{E}_{q_\Phi(z_u|\boldsymbol{r}_u,\boldsymbol{k}_u)}\left[\frac{p_\Theta(\boldsymbol{r}_u, \boldsymbol{k}_u, z_u)}{q_\Phi(z_u|\boldsymbol{r}_u, \boldsymbol{k}_u)}\right]\right) \tag{12}$$

$$\geq \mathbb{E}_{q_\Phi(z_u|\boldsymbol{r}_u,\boldsymbol{k}_u)}\left[\log p_\Theta(\boldsymbol{r}_u, \boldsymbol{k}_u, z_u)\right] - \mathbb{E}_{q_\Phi(z_u|\boldsymbol{r}_u,\boldsymbol{k}_u)}\left[\log q_\Phi(z_u|\boldsymbol{r}_u, \boldsymbol{k}_u)\right] \tag{13}$$

$$= \mathbb{E}_{q_\Phi(z_u|\boldsymbol{r}_u,\boldsymbol{k}_u)}\left[\log p_\Theta(\boldsymbol{r}_u, \boldsymbol{k}_u|z_u)\right] + \mathbb{E}_{q_\Phi(z_u|\boldsymbol{r}_u,\boldsymbol{k}_u)}\left[\log p(z_u)\right] - \mathbb{E}_{q_\Phi(z_u|\boldsymbol{r}_u,\boldsymbol{k}_u)}\left[\log q_\Phi(z_u|\boldsymbol{r}_u, \boldsymbol{k}_u)\right] \tag{14}$$

$$= \mathbb{E}_{q_\Phi(z_u|\boldsymbol{r}_u,\boldsymbol{k}_u)}\left[\log p_\Theta(\boldsymbol{r}_u, \boldsymbol{k}_u|z_u)\right] - \mathrm{D_{KL}}\left[q_\Phi(z_u|\boldsymbol{r}_u, \boldsymbol{k}_u) \mid\mid p(z_u)\right] \tag{15}$$

$$= \mathbb{E}_{q_\Phi(z_u|\boldsymbol{r}_u,\boldsymbol{k}_u)}\left[\log p_{\Theta_r}(\boldsymbol{r}_u|z_u) + \log p_{\Theta_k}(\boldsymbol{k}_u|z_u)\right] - \mathrm{D_{KL}}\left[q_\Phi(z_u|\boldsymbol{r}_u, \boldsymbol{k}_u) \mid\mid p(z_u)\right] \tag{16}$$

B KEYPHRASE EXAMPLES

Table 4. Some keyphrases mined from the reviews. We manually grouped them by types for a better understanding.

Dataset	Type	Keyphrases
Beer	Head	white, tan, offwhite, brown
	Malt	roasted, caramel, pale, wheat, rye
	Color	golden, copper, orange, black, yellow
	Taste	citrus, fruit, chocolate, cherry, plum
CDs&Vinyl	Genre	rock, pop, jazz, rap, hip hop, R&B
	Instrument	orchestra, drum
	Style	concert, opera
	Religious	chorus, christian, gospel
Yelp	Cuisine	chinese, thai, italian, mexican, french
	Drink	tea, coffee, bubble tea, wine, soft drinks
	Food	chicken, beef, fish, pork, seafood, cheese
	Price & Service	cheap, pricy, expensive, busy, friendly
Hotel	Service	bar, lobby, housekeeping, guest, shuttle
	Cleanliness	toilet, sink, tub, smoking, toiletry, bathroom
	Location	airport, downtown, city, shop, restaurant
	Room	bed, tv, balcony, terrace, kitchen, business

C ADDITIONAL TRAINING DETAILS

The official baselines' codes from the respective authors, including the tuning procedure, are available in [4][5][6][7]. The final hyperparameters for all models and datasets are shown in Table 5. For all experiments, we used the following hardware: • **CPU**: 2x Intel Xeon E5-2680 v3 (Haswell), 2x 12 cores, 24 threads, 2.5 GHz, 30 MB cache; • **RAM**: 16x16GB DDR4-2133; • **GPU**: 1x Nvidia Titan X Maxwell; • **OS**: Ubuntu 18.04; • **Software**: Python 3.6, PyTorch 1.6.1, CUDA 10.2.

[4]https://github.com/wuga214/NCE_Projected_LRec
[5]https://github.com/k9luo/DeepCritiquingForVAEBasedRecSys
[6]https://github.com/wuga214/DeepCritiquingForRecSys
[7]https://github.com/litosly/RankingOptimizationApproachtoLLC

Table 5. Best hyperparameter setting for each model. The top table refers to Section 4.3 and the bottom one to Section 4.4.

Dataset	**Model**	H	LR	λ_{L2}	λ	λ_{KP}	λ_C	β	Iteration	Epoch	Dropout	γ	Neg. Samples
Beer	AutoRec	200	0.0001	0.00001	1.0	-	-	-	-	300	-	-	-
	BPR	200	-	0.0001	1.0	-	-	-	-	30	-	-	1
	CDAE	200	0.0001	0.00001	1.0	-	-	-	-	300	0.2	-	-
	NCE-PLRec	50	-	10000.0	1.0	-	-	-	10	-	-	1.1	-
	PLRec	400	-	10000.0	1.0	-	-	-	10	-	-	-	-
	PureSVD	50	-	-	-	-	-	-	10	-	-	-	-
	VAE-CF	50	0.0001	0.0001	1.0	-	-	0.2	-	-	0.4	-	-
	CE-VAE	100	0.0001	0.0001	1.0	0.01	0.01	0.001	-	300	0.5	-	-
	CE-VNCF	100	0.0005	0.00005	1.0	1.0	1.0	0.1	-	100	0.1	-	5
	M&Ms-VAE	300	0.00005	1e-10	3.0	-	-	0.7	-	300	0.4	-	-
CDs&Vinyl	AutoRec	200	0.0001	0.00001	1.0	-	-	-	-	300	-	-	-
	BPR	200	-	0.0001	1.0	-	-	-	-	30	-	-	1
	CDAE	200	0.0001	0.00001	1.0	-	-	-	-	300	0.2	-	-
	NCE-PLRec	200	-	1000.0	1.0	-	-	-	10	-	-	1.3	-
	PLRec	400	-	1000.0	1.0	-	-	-	10	-	-	-	-
	PureSVD	200	-	-	-	-	-	10	-	-	-	-	
	VAE-CF	200	0.0001	0.00001	1.0	-	-	0.2	-	-	0.3	-	-
	CE-VAE	200	0.0001	0.0001	1.0	0.001	0.001	0.0001	-	600	0.5	-	-
	CE-VNCF	100	0.0001	0.0001	1.0	1.0	1.0	0.1	-	100	0.1	-	5
	M&Ms-VAE	400	0.00005	1e-12	1.0	-	-	0.4	-	400	0.4	-	-
Yelp	AutoRec	50	0.0001	0.001	1.0	-	-	-	-	300	-	-	-
	BPR	100	-	0.0001	1.0	-	-	-	-	30	-	-	1
	CDAE	50	0.0001	0.001	1.0	-	-	-	-	300	0.4	-	-
	NCE-PLRec	50	-	10000.0	1.0	-	-	-	10	-	-	1.3	-
	PLRec	400	-	10000.0	1.0	-	-	-	10	-	-	-	-
	PureSVD	50	-	-	-	-	-	-	10	-	-	-	-
	VAE-CF	50	0.0001	0.001	1.0	-	-	0.2	-	-	0.2	-	-
	CE-VAE	200	0.0001	0.0001	1.0	0.01	0.01	0.001	-	600	0.4	-	-
	CE-VNCF	100	0.0005	0.0001	1.0	1.0	1.0	0.1	-	100	0.1	-	5
	M&Ms-VAE	500	0.00005	1e-10	10.0	-	-	0.8	-	300	0.7	-	-
Hotel	AutoRec	50	0.0001	1e-05	1.0	-	-	-	-	300	-	-	-
	BPR	200	-	0.0001	1.0	-	-	-	-	30	-	-	1
	CDAE	200	0.0001	0.001	1.0	-	-	-	-	300	0.2	-	-
	NCE-PLRec	50	-	10000.0	1.0	-	-	-	10	-	-	1.3	-
	PLRec	400	-	10000.0	1.0	-	-	-	10	-	-	-	-
	PureSVD	50	-	-	-	-	-	-	10	-	-	-	-
	VAE-CF	50	0.0001	1e-05	1.0	-	-	0.2	-	-	0.5	-	-
	CE-VAE	200	0.0001	0.0001	1.0	0.01	0.01	0.001	-	600	0.2	-	-
	CE-VNCF	100	0.0005	0.0001	1.0	1.0	1.0	0.1	-	100	0.1	-	5
	M&Ms-VAE	400	0.00005	1e-12	2.0	-	-	0.8	-	300	-	-	-

Dataset	**Model**	h	LR	λ_{L2}
Beer		0.75	0.001	0
CDsVinyl	M&Ms-VAE $\xi(\cdot)$	3.0	0.001	1e-10
Yelp	(Critiquing)	2.0	0.001	0
Hotel		5.0	0.001	1e-10

D MULTI-STEP CRITIQUING ON THE WHOLE SET OF ITEMS

Here we replicate the experiment in Section 4.4, but we use instead all the available items (see Table 1 for the sizes). When the evaluation is conducted on 300 items (see Figure 4), we see that users indeed find a specific item using our technique with a high success rate (i.e., around 90%). However, in Figure 7 where thousands of items are available, the results show that current methods are not yet good enough to achieve similar results for such a large number. Nevertheless, M&Ms-VAE clearly outperforms on average other methods and still achieves an average success rate of 30%.

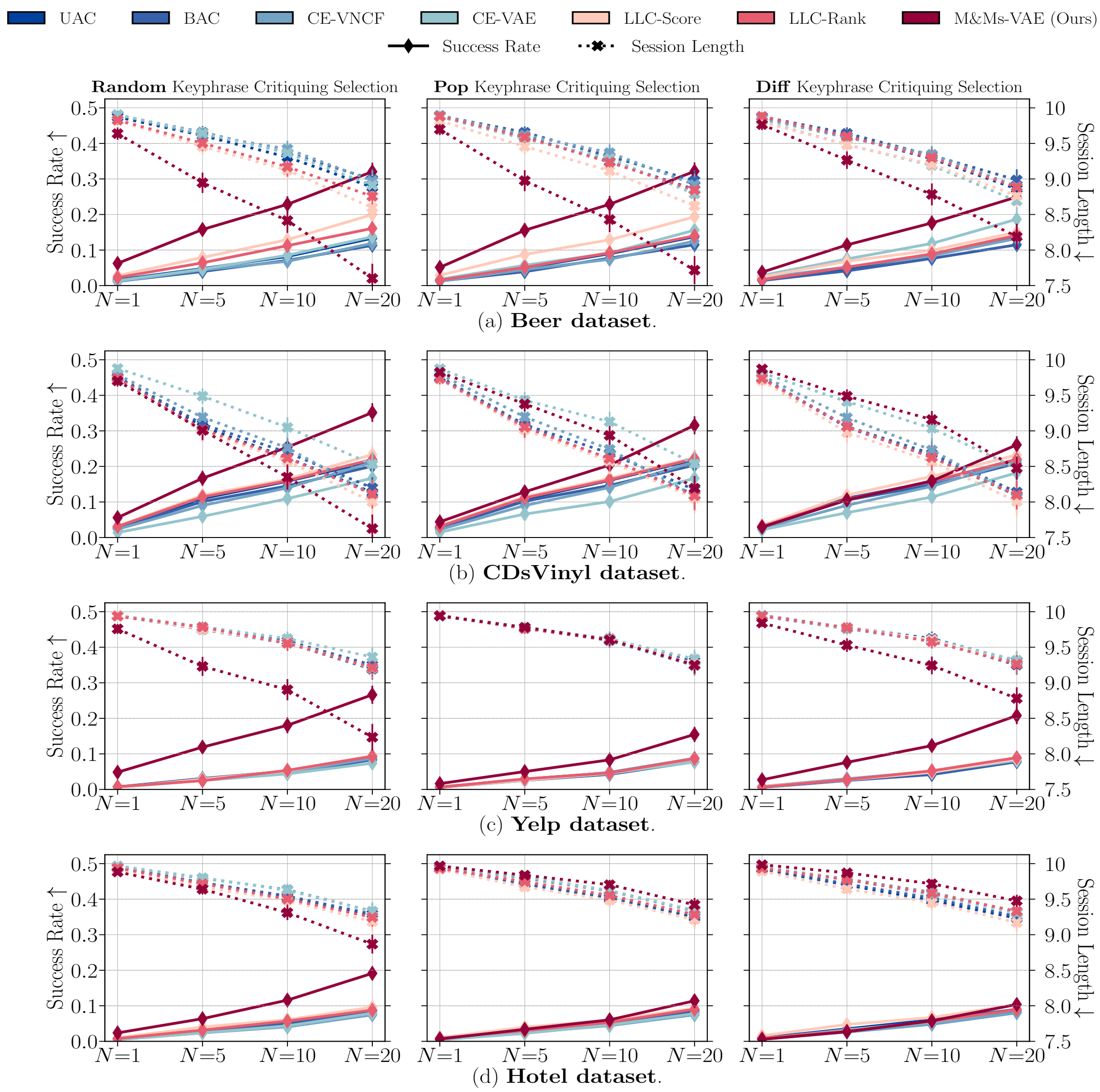


Fig. 7. Multi-step critiquing performance after 10 turns of conversation on all items. For each dataset and keyphrase critiquing selection method, we report the average success rate (left y-axis) and session length (right y-axis) at different Top-N with 95% confidence interval.