



A Novel Variational Autoencoder with Multi-position Latent Self-attention and Actor-Critic for Recommendation

Jiamei Feng, Mengchi Liu^(✉), Song Hong , and Shihao Song

School of Computer Science, South China Normal University, Guangzhou 510631, China

liumengchi@scnu.edu.cn

Abstract. Variational Autoencoder (VAE) has been extended as a representative nonlinear latent method for collaborative filtering recommendation. As a high-dimensional representation of data, latent vectors play a vital role in the transmission of important information in a VAE model. However, VAE-based models suffer from a common limitation that the transmission ability of the latent vectors' important information is limited, resulting in lower quality of global information representation. To address this, we present a novel VAE model with multi-position latent self-attention and reinforcement learning' actor-critic algorithm. We first build a multi-position latent self-attention model, which can learn richer and more complex latent vectors and strengthens the transmission of important information at different positions. At the same time, we use reinforcement learning to enhance the interactive learning process of collaborative filtering recommendation training. Specifically, our model is stable and can be easy applied in the recommendation. We observed significant improvements over the previous state-of-the-art baselines on three social media datasets, where the largest improvement can reach 26.10%.

Keywords: Recommendation · Variational autoencoder · Actor-Critic · Latent attention

1 Introduction

Collaborative filtering [2] models are the most commonly used and widely applied methods for recommendation. After decades of development, recommendation techniques have shifted from the latent linear models to deep nonlinear models for modeling latent features and feature interactions among sparse features. Variational AutoEncoder (VAE) [11] has been extended to Mult-VAE [12], a representative nonlinear method for collaborative filtering, and has received widespread attention in the recommender system community in recent years. VAE-based methods [1, 3, 13, 14, 16] are significantly outperform the classical latent variable models.

However, it remains a question whether these models can effectively learn representations of user preferences. Latent vectors, which serve as a high-dimensional representation of user-item scoring data, play a vital role in transmitting important information. Attention mechanism can solve the transmission of important information, but it is usually applied to sequence recommendation [21], which results depend on sequence with poorly parallelism. With the development of study, there are better applications in the non-sequenced data. Here the latent vector transmission of VAE model is independent of sequence. And as a deep learning model, VAE also lacks interactivity, which is its inherent question. Reinforcement learning is a good way to face this question, but only reinforcement learning is difficult to use in VAE directly.

In order to address the above problem, we propose a novel VAE-based model for collaborative filtering recommendation, which we call the VAE with multi-position latent self-attention and actor-critic or acMLAVAE for short. As the name suggests, our model consider multi-position latent self-attention, where each position is independent. Considering that different positions have different important information nodes, this paper designs a latent self-attention mechanism, which combines feedforward network and residual network to make it suitable for latent vectors at different positions. On the one hand, it can adaptively assign larger weight values to the nodes containing important information in the latent vector, and conversely, assign smaller weight values. On the other hand, the improved VAE latent function transfer does not consider sequence, and mining information through the similarity between vectors can achieve the attention of global information. Especially we add reinforcement learning of actor-critic [13] on the training process. This is a good way to solve the problem of poor interactivity of VAE. The main contributions of this work are as follows:

1. A novel model of VAE for collaborative filtering using multi-position and latent self-attention is proposed, which allows the model to learn richer and more complex latent vectors and strengthens the transmission of important information.
2. We use VAE with multi-position latent self-attention as the actor and a new neural network as the critic to enhance interaction in the reinforcement learning process. The model realizes the common advantages of deep learning and reinforcement learning.
3. Our model acMLAVAE is comprehensively evaluated on numerical experiments and shows that it consistently outperforms the eight existing baselines across all metrics. Especially, we demonstrate good stability of our model and trains latent self-attention’s weights automatically.

2 Related Works

Actor-Critic. Reinforcement learning encompasses a large category of machine learning. Actor-critic is one main class of reinforcement learning that is based on estimating only the values of states. In actor-critic, the actor implements a

policy and the critic attempts to estimate the value of each state under this policy [15]. With the development of actor-critic, recent papers [9, 19] have added attention to actor-critic algorithm to improve reinforcement learning effects. But only reinforcement learning improvement cannot be directly applied to collaborative filtering recommendation. Some of the combination of actor-critic and VAE are more applied to the automatic driving collar [5, 6], so we need to explore the reinforcement field of collaborative filtering.

Attention. The attention mechanism has a relatively wide and mature application in the field of graph attention networks [18] and translation [17]. In text processing, the traditional attention is generally used in the encoder-decoder of the sequence model Long Short Term Memory (LSTM) [7]. Such as DA-LSTM-VAE [22] is a model, which attention is based sequence LSTM that the weight calculation needs target. However, those models of processing the sequence depend on the order between words and cannot be calculated in parallel. The transformer [17] greatly promotes the development of self-attention mechanism. Therefore, it provides us with a good idea to solve VAE's important latent information transmission problem.

3 Preliminaries

Under massive sparse data, VAE modeling has been investigated for collaborative filtering to achieve good performance. We employ the notation $u \in 1, \dots, U$ to denote users and $i \in 1, \dots, I$ to denote items. x_u is a bag-of-words vector from user u in matrix X .

3.1 Variational Autoencoder for Collaborative Filtering

VAE is a typical encoder-decoder architecture and have powerful generation capability. The model of VAE is shown in Fig. 1. VAE uses the variational reasoning framework to learn the approximate posterior of latent variables.

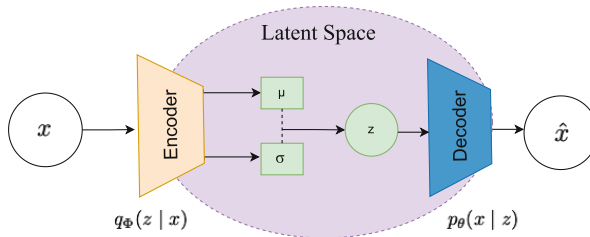


Fig. 1. VAE of Encoder and Decoder.

It is clear to see that VAE consists of two parts, the encoder and decoder. In encoder part, maps the input data x to a latent vector z in the latent space, and outputs the mean μ and variance σ^2 of the latent vector.

$$z = \mu + \epsilon \odot \sqrt{\exp(\log(\sigma^2))} = q_{\Phi}(z | x) \quad (1)$$

Here, the Φ -parametered neural network encoder is expressed as $q_{\Phi}(z | x)$. ϵ is a vector sampled from the standard normal distribution $N(0, 1)$, \odot denotes element-wise multiplication, and $\log(\sigma^2)$ represents the logarithm of the variance.

The decoder part generates the reconstructed data \hat{x} . Here, θ represents the parameters of the decoder. The θ -parametered neural network decoder is expressed as $p_{\theta}(x | z)$.

$$\hat{x} = Decoder(z; \theta) = p_{\theta}(x | z) \quad (2)$$

The loss function of VAE consists of two parts, the reconstruction error and the KL divergence.

$$L_{VAE} = L_{reconstruction}(x, \hat{x}) - KL(q(z|x)||p(z)) \quad (3)$$

Here, $L_{reconstruction}$ represents the reconstruction error [10], KL represents the KL divergence, $q(z|x)$ represents the posterior distribution, and $p(z)$ represents the prior distribution.

$$L_{reconstruction} = E(q_{\Phi}(z|x) [\log p_{\theta}(x | z)]) \quad (4)$$

In Eq. 5, k represents the dimension of the latent vector z , that is, the length of the latent vector.

$$KL(q(z|x)||p(z)) = -\frac{1}{2} \sum_k (1 + \log((\sigma_k)^2) - (\mu_k)^2 - (\sigma_k)^2) \quad (5)$$

A traditional VAE model typically utilizes Gaussian log-likelihood [4] for latent space collaborative filtering. Specifically, for each user u , the corresponding latent representation $z_u = 0$ follows a standard Gaussian prior distribution, i.e., $z_u \sim N(0, I)$. We cannot get distribution from z_u directly. In order to solve this problem, we use $q_{\Phi}(z_u | x_u)$ instead of $p_{\theta}(x_u | z_u)$ to realize the process of variational inference. VAE is trained using the backpropagation algorithm to minimize the loss function L_{VAE} to update the Φ neural network of the encoder and θ neural network of the decoder.

$$L_{VAE}(x_u; \Phi, \theta) = E(q_{\Phi}(z_u|x_u) [\log p_{\theta}(x_u | z_u)]) - KL(q_{\Phi}(z_u | x_u) || p_{\theta}(z_u)). \quad (6)$$

3.2 Self-attention

The transformer [17] greatly promotes the development of self-attention. It consists self-attention and multi-head self-attention. Multi-head self-attention is

only a distributed and parallel expression of self-attention, which does not change the internal structure of attention, but improves the training efficiency. So this paper uses multi-head self-attention showing as self-attention, because parallel does not affect the final result. A self-attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. From the definition of attention [17], we know that the queries, keys and values are packed together into matrices Q , K and V . Here, d is the dimension of queries and keys. M is the masks. We know the matrix of outputs as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}} + M)V \tag{7}$$

From Eq. 8, we know that the calculation of self-attention is not dependent on the sequence of words, but on the similarity of words to information mining.

4 Proposed Model

4.1 The Framework

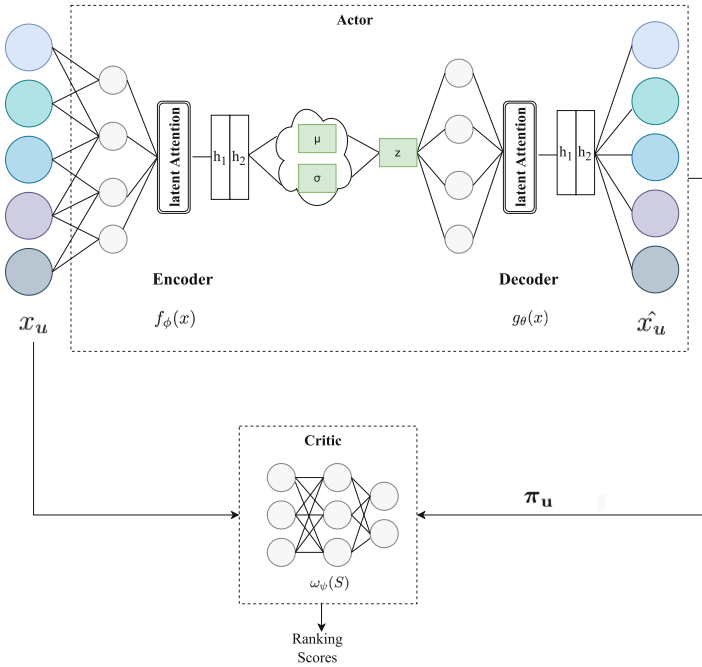


Fig. 2. The Overall Framework of acMLVAE.

VAE is a deep learning model that can represent latent vectors friendly, but lacks interactivity. The reinforcement learning algorithm actor-critic learns the optimal strategy through interaction. Through the joint learning of the two models, the deep relationship representation of the latent vector is obtained to complete the final recommendation task. We set Φ -parameterized neural network encoder as $g_{\Phi}(x)$ and θ -parameterized neural network decoder as $f_{\theta}(x)$ as actor part. We propose multi-position latent self-attention that is used in the actor process. The location of latent attention in Fig. 2 corresponds to the position of the multi-position latent self-attention in the encoder and decoder. Critic process uses Ψ -parameterized neural network function ω , which takes in the prediction π to compare with the ground-truth x . The critic process uses Ψ -parameterized neural network function $\omega: \{\pi; x\} \rightarrow y \in [0, 1]$, and outputs a scalar y to rate the prediction quality. The overall framework is shown in Fig. 2.

4.2 Variational Autoencoder with Multi-position Latent Self-attention (MLAVAE)

A novel model that we extend a VAE-based model [12] by adding multi-position latent self-attention in latent space, which is denoted as MLAVAE. The traditional VAE models only result in activation of a few latent vectors which will become more serious as the depth of the network deepens. As a high-dimensional representation of user-item scoring data, latent vectors play an important role in reconstructing data. For the problem that the inactive latent vector affects the quality of the reconstructed data, this paper designs a multi-position latent self-attention in Fig. 2's encode part that conforms to the latent vector, assigns a larger weight value to the nodes containing important information in the latent vector, enhances the role of important information in the reconstructed data, automatically gives a smaller weight value to the unimportant information, weakens its impact on the reconstructed data, and reduces noise. The general idea of our multi-position latent self-attention is to first utilize self-attention to generate an attention-weight matrix and then obtain output by applying element wise multiplication between the attention-weight matrix and the original input. Multi-position latent self-attention can quantify the importance of different information through the weight of latent vector in the process of data learning, and the weight is automatically adjusted during the learning process by random initialization without manual modification of parameters. This dynamic adjustment of the learning process makes it possible to select important information from different datasets, finally improving the accuracy of recommendation. In the following, we shall explain this process step by step.

Step 2 to step 7 is the process of VAE with multi-position latent self-attention in Algorithm 1. In step 2, when the data enters the encoder, it becomes a vector in the latent space, where h takes the latent vector. In order to strengthen the influence of key latent vectors on the collaborative filtering recommendation, and taking into account the different key information carried by latent vectors at different positions, we propose a multi-position method, which can be encoder, decoder, or encoder+decoder. In step 3, we assign (h, h, h) to (K, Q, V) of Eq. 8 in

encoder. Here, M is the masks of 10 percent. First, we do not need to adjust the parameters manually. Second, the weights of MLVAE are adjusted adaptively, and the nodes containing important information in the latent vectors are given larger weights. We get multi-position latent self-attention denoted as MLA .

$$MLA = softmax(\frac{hh^\top}{\sqrt{d}} + M)h \quad (8)$$

Step 4 is feedforward network that W_1 , W_2 , b_1 and b_2 are randomly initialized matrices to ensure weight independent learning. Step 5 is residual network to avoid gradient disappearance and network degradation, so as to reduce the training difficulty of deep neural network. Here $f_u(\cdot)$ is full connection layer, which weight is static. In step 6, the latent vector is passed to the decoder. Then, we add the above multi-position latent self-attention to Eq. 8, and get Eq. 9.

$$L_{MLVAE}(x_u; \Phi, \theta) = E(q_\Phi(z_u | x_u) [\log p_\theta(x_u | z_u, MLA)] - KL(q_\Phi(z_u | x_u) || p_\theta(z_u, MLA))). \quad (9)$$

Finally, we use L_{MLVAE} to update the Φ and θ neural network.

4.3 The Design of MLVAE with Actor-Critic(acMLVAE)

Algorithm 1. acMLVAE

Initialization: Randomly initialize actor network parameters Φ , θ and critic parameters Ψ , $f_u(\cdot)$ is full connection layer, time steps T .

Input: State interaction matrix' vector x_u into actor network, W_1 , W_2 , b_1 and b_2 are randomly initialized matrices, probability distribution prediction π_u is obtained as action.

- 1: **for** $t = 1$ to T **do**
 - 2: $h \leftarrow encoder : f_u(x_u)$;
 - 3: $h_0 \leftarrow Attention(h, h, h)$;
 - 4: $h_1 \leftarrow (W_1 h_0 + b_1)W_2 + b_2$;
 - 5: $h_2 \leftarrow h_1 + f_u(h_1)$;
 - 6: $decoder \leftarrow h_2$;
 - 7: Update Φ , θ with $\frac{\partial L_{MLVAE}}{\partial \Phi}$ and $\frac{\partial L_{MLVAE}}{\partial \theta}$ in Eq. 9;
 - 8: $\pi_u \leftarrow decoder$;
 - 9: Decompose π_u into S ;
 - 10: Compute $L_{critic} = \|\omega_\psi(S) - y\|^2$;
 - 11: Update ψ with $\frac{\partial L_{critic}}{\partial \psi}$.
 - 12: **end for**
-

Finally the overall framework learning process is provided in Algorithm 1. The process of step 2 to step 7 is the actor training process, and step 8 to step 11 is the critic training process. Critic part decomposes probability π_u into S , and computes the mean square error $\|\omega_\psi(S) - y\|^2$ to get L_{critic} . In the mass,

Φ -parameteried neural network in encoder and θ -parameteried neural network in decoder obtain more useful features, and Ψ -parameteried neural network in critic minimizes the mean square error to complete the recommendation. Specifically, the above three neural network architectures are as follows. Encoder: linear \rightarrow tanh \rightarrow latent self-attention \rightarrow feedforward network \rightarrow residual network \rightarrow Linear. Decoder: linear \rightarrow tanh \rightarrow latent self-attention \rightarrow residual network \rightarrow feedforward network \rightarrow residual network \rightarrow softmax. Critic: batch normalization \rightarrow relu \rightarrow sigmoid. Tanh, relu and sigmoid are activation function. The sigmoid function is used to rescale the result to $[0, 1]$, which is more suitable for latent variables.

5 Experiments

5.1 Experiment Platform and Datasets

We develop the proposed algorithm acMLVAE with tensorflow architecture in a CentOS 7 system (PowerLeader server with an Intel CPU and 1T memory). Batch size is 500. The dimension of the word embedding is 600, and the number of attention heads is 8.

We conducted our empirical evaluations on three real world social media and publicly available datasets from Mult-VAE [12]: ML-20M, Netflix and MSD. The ML-20M dataset is a user-movie rating dataset. Each user left their rating score on a scale of 5, which has been binarized for our study. The Netflix dataset comprises of user-movie ratings and was originally introduced for the Netflix Prize competition. The rating scale is the same as the ML-20M. The MSD dataset used in this study contains user-song play count information. Users who interacted with less than 5/20 movies/songs were removed, and the data were converted to binary as in the case of implicit feedback.

5.2 Evaluation Indicators

Two standard metrics are utilized for evaluating the quality of recommendation, including normalized discounted cumulative gain (NDCG@ R) and Recall@ R . Recall@ R and NDCG@ R are the universal evaluation metrics based on ranking, where R is a parameter. Recall@ R measures the fraction of the positive items in the test data. The R in Recall@ R keeps the first R items equally important. A higher NDCG@ R represents the positive items in the test data are ranked higher in the ranking list. The R in NDCG@ R supports multi-level similarity, which makes retrieval more reasonable when designing multi-table data. $m(r)$ is the item of rank r . H_0 is hold-out items that user u clicks on interaction. Recall@ R for a given user u can be defined as Eq. 10.

$$\text{Recall@}R(u, m(r)) = \frac{\sum_{r=1}^R \delta [m(r) \in H_0]}{\min(R, |H_0|)}. \quad (10)$$

Discounted cumulative gain ($\text{DCG}@R$) for user u is defined as Eq. 11. $\text{NDCG}@R$ linearly normalizes $\text{DCG}@R$ to $[0, 1]$.

$$\text{DCG}@R(u, m(r)) = \sum_{r=1}^R \frac{2^{\delta[m(r) \in H_0]} - 1}{\log(r+1)}. \quad (11)$$

5.3 Experimental Results and Analysis

Overall Performance of AcMLVAE. The results in Table 1 clearly show that the performance of acMLVAE on each dataset surpasses other baselines.

Table 1. acMLVAE compare with eight kinds of models on three datasets. Bold indicates the overall best.

Datasets Models	ML-20M			Netflix			MSD		
	Recall@20	Recall@50	NDCG@100	Recall@20	Recall@50	NDCG@100	Recall@20	Recall@50	NDCG@100
acMLVAE	0.4448	0.5729	0.5018	0.4098	0.4797	0.4667	0.3017	0.3872	0.3909
RaCT	0.3960	0.5368	0.4274	0.3578	0.4490	0.3906	0.2587	0.3560	0.3190
SE-VAE	0.4178	0.5555	0.4471	0.3771	0.4631	0.4090	NULL	NULL	NULL
RecVAE	0.4113	0.5502	0.4399	0.3597	0.4481	0.3930	0.2711	0.3719	0.3195
MacridVAE	0.3964	0.5290	0.4249	0.3458	0.4347	0.3798	NULL	NULL	NULL
Multi-VAE	0.3912	0.5323	0.4228	0.3509	0.4428	0.3857	0.2600	0.3560	0.3100
Multi-DAE	0.3870	0.5240	0.4190	0.3440	0.4380	0.3800	0.2660	0.3630	0.3130
CDAE	0.391	0.523	0.418	0.343	0.428	0.376	0.188	0.283	0.237
WMF	0.360	0.498	0.386	0.316	0.404	0.351	0.211	0.312	0.257

It also proves that the addition of multi-position latent self-attention with actor-critic can improve the performance of VAE generally, because multi-position latent self-attention with actor-critic integrates multiple layers to achieve latent vector to transmit important information. Multi-VAE [12] is currently the most advanced collaborative filtering model based on encoder and decoder network. The experiment evaluated the model in this paper from three indicators: Recall@20, Recall@50 and NDCG@100, and the indicators increased by [13.70%, 7.63%, 18.68%] on ML-20M, [16.79%, 8.33%, 21.00%] on Netflix, and [16.04%, 8.76%, 26.10%] on MSD. Especially in NDCG@100 's performance is excellent in three datasets. RaCT [13], MacridVAE [14], SE-VAE [3] and RecVAE [16] are all improved, and those four models are based on Multi-VAE. MacridVAE [14] adds auxiliary information based on Multi-VAE. SE-VAE [3] focuses on sampling based on Multi-VAE. RecVAE [16] changes the prior distribution based on Multi-VAE. Compared with Multi-VAE, although their Recall@20, Recall@50 and NDCG@100 values have been improved, but not fully utilized the latent vectors containing important information. RaCT [13] adds actor-critic based on Multi-VAE. Although reinforcement learning can improve interaction, it ignores the internal connection of the model. Compared with WMF [8] and CDAE [20] models, the evaluation indicators on the ML-20M, Netflix and MSD datasets have significantly improved.

Improvement on Different R. The other baselines take Multi-VAE [12] as the baseline, so we mainly focus on the comparison with Multi-VAE. Table 2 shows under different numbers of R (1, 3, 5, 20, 50, 100, 200) can be improved

significantly, which indicates acMLVAE’s stability of neural network training in recommendation.

Table 2. Improvement on different R.

model	acMLVAE	Mult-VAE
Recall@50	0.5729	0.5323
Recall@20	0.4448	0.3912
NDCG@200	0.5362	0.4606
NGCG@100	0.5018	0.4228
NDCG@5	0.4449	0.3129
NDCG@3	0.4901	0.3271
NDCG@1	0.6131	0.3602

In collaborative filtering, we are interested in predicting the top-N items to the user in a sense that top-N items are more carefully observed in recommender systems. Recall@ R is the proportion of relevant (clicked) items predicted in the top R items. This metric becomes useful considering that the online users focus on top-N recommended items on first page or upper area without scrolling down. However, Recall@ R indicates the proportion of items brought in the first R position that were actually in the target subset, and does not consider the order. NDCG overcomes this issue by using the monotonically increasing discount. It emphasizes the importance of higher ranking than lower ones. Therefore, our model improves steadily under different R recommended, especially the NDCG@1 up to 61.31%.

Influence of Multi-position. As shown in Fig. 2, we known latent attention can be in the encoder or in the decoder or in the encoder and decoder (short as double) at the same time. Remove all other impression factors, and here only verify the influence of attention position. Table 3 shows the influence of multi-position.

On ML-20M, the effect of latent attention is the best in encoder at most cases, and the situation of using latent attention in both encoder and decoder is the second best. On Netflix, latent attention in encoder is the best, and in decoder is the second best. On MSD, second best appears in decoder and double. In datasets with different fields, sizes and sparsity, the effect of latent attention is different in different locations. So each location reflects different important information about latent vectors. On the whole, the effect of latent attention is best when the position in encoder.

Ablation Studies. In order to figure out the contribution of different components to the performance of our acMALVAE, we conduct some ablation studies.

Table 3. The different positions in encoder, decoder and double. Bold indictes the overall best, and underline indicates second best.

Dataset	ML-20M			Netflix			MSD		
	Decoder	Encoder	Double	Decoder	Encoder	Double	Decoder	Encoder	Double
NGCG@100	0.4243	0.4301	<u>0.4255</u>	<u>0.3781</u>	0.3831	0.3667	<u>0.3065</u>	0.3117	0.3057
Recall@50	0.5376	<u>0.5391</u>	0.5405	<u>0.4381</u>	0.4405	0.4258	<u>0.3570</u>	0.3578	0.3560
Recall@20	0.3962	0.4021	<u>0.3992</u>	<u>0.3448</u>	0.3495	0.3326	<u>0.2588</u>	0.2615	0.2581
NDCG@200	0.4610	0.4672	<u>0.4625</u>	<u>0.4254</u>	0.4301	0.4141	0.3247	0.3396	<u>0.3336</u>
NDCG@5	0.3092	0.3239	<u>0.3110</u>	<u>0.3199</u>	0.3326	0.3066	0.2248	0.2369	<u>0.2250</u>
NDCG@3	0.3187	0.3375	<u>0.3226</u>	<u>0.3298</u>	0.3457	0.3168	<u>0.2440</u>	0.2586	0.2438
NDCG@1	0.3489	0.3763	<u>0.3552</u>	<u>0.3522</u>	0.3761	0.3417	0.2764	0.2985	<u>0.2782</u>

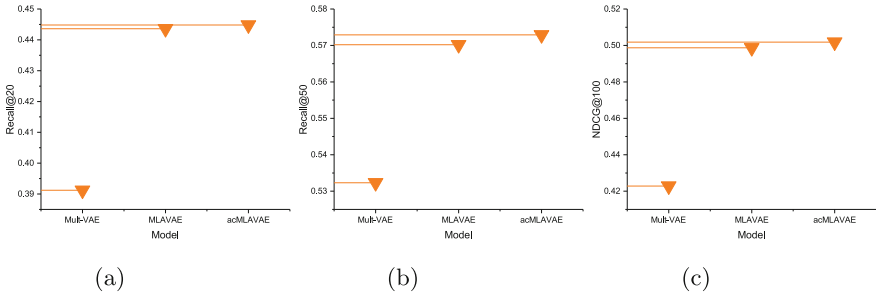


Fig. 3. Ablation studies on ML-20M dataset. (a), (b) and (c) correspond to Recall@20, Recall@50 and NDCG@100 respectively.

Notice that we only present the results on Recall@20, Recall@50 and NDCG@100 because the trends on other metrics are similar to theirs. Specifically, we remove the target representation enhancement model. Figure 3 shows ablation studies on ML-20M. acMLVAE is a novel model proposed by this paper. When acMLVAE does not use actor-critic, it is MLVAE. This reflects the interaction of the combination of intensive learning and deep learning. When MLVAE does not use multi-position latent self-attention, it is Multi-VAE. Which demonstrates that multi-position latent self-attention learns richer and complex latent vectors and strengthens the transmission of important information. From the comparison of the Recall@20, Recall@50 and NDCG@100, we can know that the addition of multi-position latent self-attention and actor-critic is meaningful. We only show the results on ML-20M because the results on Netflix and MSD are similar to Fig. 3. Therefore, the validation of three datasets can further illustrate two points. First, our model has good stability. Second, our model has good performance in collaborative filtering recommendation.

6 Conclusion

The acMLVAE model can increase interaction, and enhance the training process to learn richer and complex latent vectors and strengthens the transmission

of important information in collaborative filtering recommendation. Compared to other baselines, acMLVAE shows superior performance and exhibits strong stability. The research in this paper is more inclined to strengthen the importance of the hidden vector itself. In future work, more complex data forms and data relationships will be incorporated, such as knowledge graph, graph attention networks to explore the impact on the recommendation effect.

Acknowledgements. This work was partly supported by the Guangzhou Key Laboratory of Big Data and Intelligent Education (No.201905010009) and National Natural Science Foundation of China (No.61672389).

References

1. Askari, B., Szlichta, J., Salehi-Abari, A.: Variational autoencoders for top-k recommendation with implicit feedback. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2061–2065. ACM (2021)
2. Chen, J., Lian, D., Jin, B., Huang, X., Zheng, K., Chen, E.: Fast variational autoencoder with inverted multi-index for collaborative filtering. In: WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25–29, 2022, pp. 1944–1954. ACM (2022)
3. Cho, Y., Oh, M.: Stochastic-expert variational autoencoder for collaborative filtering. In: WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25–29, 2022, pp. 2482–2490. ACM (2022)
4. Gopalan, P., Hofman, J.M., Blei, D.M.: Scalable recommendation with hierarchical poisson factorization. In: UAI, pp. 326–335 (2015)
5. Gupta, A., Khwaja, A.S., Anpalagan, A., Guan, L.: Safe driving of autonomous vehicles through state representation learning. In: 17th International Wireless Communications and Mobile Computing, IWCMC 2021, Harbin City, China, June 28 - July 2, 2021, pp. 260–265. IEEE (2021)
6. Gupta, A., Khwaja, A.S., Anpalagan, A., Guan, L., Venkatesh, B.: Policy-gradient and actor-critic based state representation learning for safe driving of autonomous vehicles. *Sensors* **20**(21), 5991 (2020)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
8. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proceedings of 8th IEEE International Conference on Data Mining, pp. 263–272. IEEE (2008)
9. Iqbal, S., Sha, F.: Actor-attention-critic for multi-agent reinforcement learning. arXiv (2018)
10. Karamanolakis, G., Cherian, K.R., Narayan, A.R., Yuan, J., Tang, D., Jebara, T.: Item recommendation with variational autoencoders and heterogeneous priors. In: Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems (DLRS), pp. 10–14 (2018)
11. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings (2014)

12. Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T.: Variational autoencoders for collaborative filtering. In: Proceedings of the 2018 World Wide Web Conference (WWW), pp. 689–698 (2018)
13. Lobel, S., Li, C., Gao, J., Carin, L.: Towards amortized ranking-critical training for collaborative filtering. In: International Conference on Learning Representation (ICLR) (2020)
14. Ma, J., Zhou, C., Cui, P., Yang, H., Zhu, W.: Learning disentangled representations for recommendation. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), pp. 5712–5723. Vancouver, BC, Canada (2019)
15. Mustapha, S.M., Lachiver, G.: A modified actor-critic reinforcement learning algorithm. In: Conference on Electrical and Computer Engineering (2000)
16. Shenbin, I., Alekseev, A., Tutubalina, E., Malykh, V., Nikolenko, S.I.: Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In: Proceedings of the Thirteenth ACM International Conference on Web Search and Data Mining (WSDM), pp. 528–536. ACM, Houston, TX, USA (2020)
17. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA, pp. 5998–6008 (2017)
18. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018)
19. Yang, N., Lu, Q., Xu, K., Ding, B., Gao, Z.: Multi-actor-attention-critic reinforcement learning for central place foraging swarms. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–6 (2021)
20. Yang, S.H., Long, B., Smola, A.J., Zha, H., Zheng, Z.: Collaborative competitive filtering: learning recommender using context of user choice. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 295–304 (2011)
21. Zhao, J., Zhao, P., Zhao, L., Liu, Y., Sheng, V.S., Zhou, X.: Variational self-attention network for sequential recommendation. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp. 1559–1570. IEEE (2021)
22. Zhao, Y., Zhang, X., Shang, Z., Cao, Z.: Da-LSTM-VAE: Dual-stage attention-based LSTM-VAE for KPI anomaly detection. *Entropy* **24**(11), 1613 (2022)