Noise-Augmented Contrastive Learning for Sequential Recommendation

Kun He[1], Shunmei Meng[2,3(✉)], Qianmu Li[2], Xiao Liu[1], Amin Beheshti[4], Xiaoxiao Chi[4], and Xuyun Zhang[4]

[1] School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
hekun@njust.edu.cn
[2] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
mengshunmei@njust.edu.cn
[3] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[4] School of Computing, Macquarie University, Sydney, Australia

Abstract. Recently, contrastive learning has been widely used in the field of sequential recommendation to solve the data sparsity problem. CL4Rec augments data through simple random crop, mask, and reorder, while DuoRec proposes a model-level data augmentation method. However, these methods do not take into account the issue of noisy data in sequential recommendation, such as false clicks during browsing. The noise may lead to poor representations of learned sequences and negatively affect the augmented data. Current sequential recommendation methods tend to learn the user's intention from their original sequences, but these methods have certain limitations as the user's intention for the next interaction may change. Based on the above observations, we propose Noise-augmented Contrastive Learning for Sequential Recommendation (NCL4Rec). Our NCL4Rec proposes sequential noise probability-guided data augmentation. We introduce supervised noise recognition during training instead of obtaining it from original sequences. Moreover, we design positive and negative augmentations of the sequence and design unique noise loss function to train them. Through experiments, it is verified that our NCL4Rec consistently outperforms the current state-of-the-art models.

Keywords: Sequential Recommendation · Contrastive Learning

1 Introduction

Sequential recommendation predicts potentially interesting items based on the user's historical behavior. In the internet age, the amount of user behavior data and available items has grown exponentially [1]. The deep neural network learns item representation through a large amount of data, and many classic models

F. Zhang et al. (Eds.): WISE 2023, LNCS 14306, pp. 559–568, 2023.
https://doi.org/10.1007/978-981-99-7254-8_43

emerge. For example, Caser [11] employs a convolutional neural network (CNN) as the backbone network, and GRU4Rec [4] uses a recurrent neural network (RNN) as the backbone network. In particular, the transformer [12] structure shines in sequential recommendation, such as SASRec [5], BERT4Rec [9].

However, due to the sparseness of sequence data, deep neural network cannot learn accurate item representations. The emergence of contrastive learning [6] solves the problem of sparse sequence data to a certain extent. CL4Rec [15] augments data through random crop, mask and reorder. DuoRec [8] utilizes a Dropout based approach to enhance sequence representation at the model level. On the other hand, it mines positive and negative samples using sequences of similar target items. But due to the noise in the sequence data, the augmented data is still disturbed by the noise in the original sequence.

But contrastive learning methods do not solve the problem of noise in the sequence. Noise has always been a major difficulty in representation learning and is no exception in sequential recommendation [13,14]. For example, in real online shopping, the user's mistaken click may not be the user's real intention behavior. The augmented data generated by randomly cropping, masking and reordering the original sequence may lack robustness due to the presence of noise data. Poor quality data augmentation can have negative effects on model training. Furthermore, most of current methods obtain the user's intent from the user's original sequence [3,7,10]. And it is easy to think of the user's recent behavior as the user's intention or query vector, but it may not be accurate due to the changing existence of the user's intention.

Based on the above observation, we propose a Noise-augmented Contrastive Learning for Sequential Recommendation (NCL4Rec) to address the noise problem in sequential recommendation. In our method, we use noise probabilities to guide the data augmentation process and mitigate the impact of noise in the original sequence. We introduce supervised noise recognition during training instead of relying on the original sequence, thereby eliminating the influence of noise in the original data. The noise probability is dynamically updated online after a certain number of training epochs. During training, we calculate the noise probability and design positive and negative sample augmentations based on it. Positive samples are generated by processing items with low noise probability, while negative samples are generated by processing items with high noise probability. Additionally, we design positive and negative loss functions to minimize the distance between positive samples and maximize the distance between positive and negative samples.

Our contributions:

- We propose a Noise-augmented Contrastive Learning for Sequential Recommendation (NCL4Rec), which addresses noise issues and data sparsity by unifying sequential recommendation and self-supervised contrastive learning methods.
- We propose novel noise-guided data positive and negative augmentations to better discriminate noisy data by exploiting the relevance of items to user intent. And a noise loss function is designed to better distinguish noise items from normal items.

– We conduct extensive experiments on three benchmark datasets, and our method consistently outperforms currently existing state-of-the-art models, with performance gains ranging from 3.37% to 7.10%.

2 Problem Formulation

Formally, let $S_u = (s_u^1, s_u^2, \ldots, s_u^n)$ be a sequence of items, and let s_u^{n+1} be the next item in the sequence to be predicted. We define the problem of sequence recommendation as follows:

Given a set of training sequences $D = (S_u, s_u^{n+1})$, where each training sequence S_u consists of n items, and the corresponding next item s_u^{n+1}, the goal is to learn a function f that maps a user's historical sequence S_u to the next item s_u^{n+1}. More formally, we seek a function f such that:

$$s_u^{n+1} = f(S_u) \tag{1}$$

where f is learned from the training set D. The learned function f can then be used to make predictions on new, unseen sequences.

3 Methods

The emphasis of this paper is on effective data augmentation, and there is no detailed description of the sequence encoding model. Instead, we use the backbone network that is commonly used in contrastive learning-based sequential recommendation models. It's important to note that the purpose of contrastive learning methods is to address the problem of sparse training data and help us obtain a more effective encoding model.

In this section, we describe in detail our proposed **Noise-augmented Contrastive Learning for Sequential Recommendation** (**NCL4Rec**). The framework of our method is shown in Fig. 1. Our method mainly consists of four parts, **(1) the generation of sequence item noise probabilities; (2) data augmentation guided by noise probabilities, (3) user representation encoding model, (4) noise contrastive loss function**.

3.1 The Generation of Sequence Item Noise Probabilities

For our user sequence item, there are often a lot of noise data. Noise is an item that does not conform to the user's intention. Most current methods are based on the original sequence to enhance the data of the item. However, the user's sequence behavior will be transferred according to the user's next item, so we use the user's target item to calculate the user's noise probability. On the one hand, it can effectively eliminate the interference between the original sequences and grasp the user's intention more accurately. The user's intention transfer can be better learned. We define the user sequence as $S_u = s_u^1, s_u^2, s_u^3 ... s_u^n$, where n

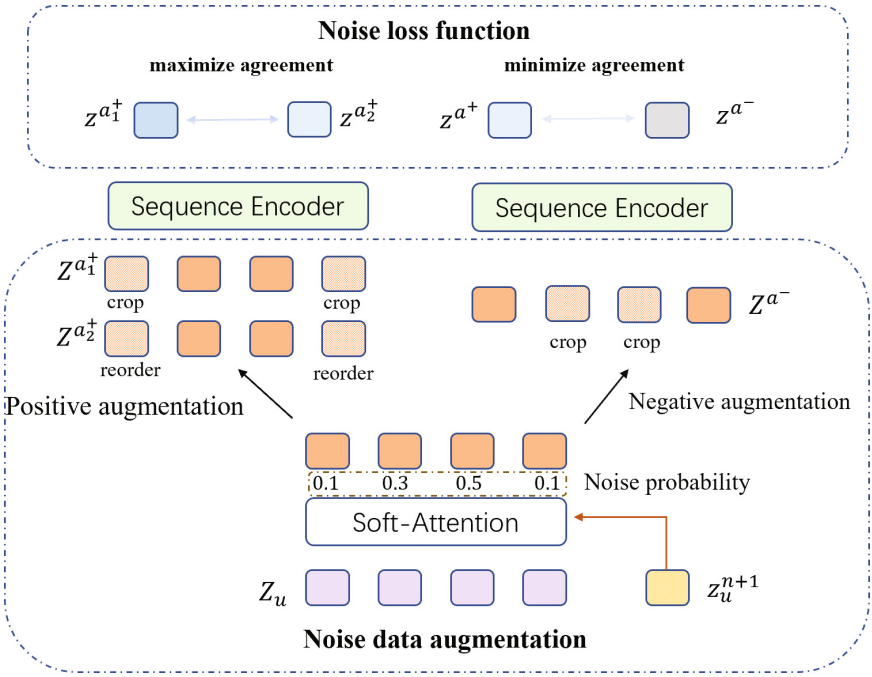

Fig. 1. Framework of NCL4Rec.

is the sequence length. s_u^{n+1} is the user's next interaction item, which is also the supervision signal in our training.

First our sequence passes through the embedding layer,

$$Z_u = Embedding(S_u) \tag{2}$$

$$Z_u = z_u^1, z_u^2, z_u^3 ... z_u^n \tag{3}$$

where z_u^i is the embedding space representation of the i-th item of user u. We calculate the similarity between the target item and the sequence item through the soft attention mechanism, which represents the noise probability of the item.

$$prob(z_u^i) = 1 - \frac{\exp(cor_i)}{\sum_{j=1}^{n} \exp(cor_j)} \tag{4}$$

where $cor_i = sim(z_u^i, z_u^{n+1})$, sim is our correlation calculation method. In this article, we use cosine similarity.

From the above method, we get the noise probability of each item of the user $Porb(Z_u) = prob(z_u^1), prob(z_u^2), prob(z_u^3), ..., prob(z_u^n)$, unlike all previous methods, we use the supervision signal to directly calculate the noise probability, because we only need to calculate the noise probability in the training set, and the supervision signal will not work in the test set.

Noise Update Strategy for Sequential Items. As our noise probabilities are calculated based on the embedding representation of items, after a certain number of training epochs, the noise probabilities of items may not be accurate enough and require updating. Our update interval epoch is a hyperparameter t, and every t epochs we recompute our noise probabilities for each item. Assuming that our total training round N is 50 and t is 20, we will update the sequence item noise update in the 20th and 40th epoch of training.

3.2 Data Augmentation Based on Noise Probability

According to the noise probabilities of the items in the sequence calculated in the previous section, we perform corresponding data augmentation. In this section, we design 5 sequence data augment methods. We perform positive data augmentation and negative data augmentation on the crop and mask in CL4Rec according to the noise probability. Our reorder operation will not change the element, so we only take positive data augmentation for it.

- **Crop or Mask for Noise reduction.** In order to reduce the noise data of the user behavior sequence, we select k items with the highest noise probability to crop or mask, so that the similarity between the behavior items in the sequence and the user's intention is higher, where k is calculated by our crop or mask coefficient α, $k = \alpha|Z_u|, 0 < \alpha < 1$.

$$Z_u^{crop+} = [\hat{v}_1, \hat{v}_2, ..., \hat{v}_{|Z_u|}] \tag{5}$$

$$\hat{v}_i = \begin{cases} z_u^i, prob(z_u^i) < Porb(Z_u).sort()[k] \\ \emptyset \ or \ [mask], prob(z_u^i) >= Porb(Z_u).sort()[k] \end{cases} \tag{6}$$

- **Crop or Mask for Noise augmentation.** In order to augment the noise data of the user behavior sequence, we select k items with the smallest noise probability to crop or mask, so that the items in the sequence are contrary to our user intentions as much as possible, where k is calculated by our crop or mask coefficient β, $k = \beta|Z_u|, 0 < \beta < 1$. The formulaic expression is as before
- **Reorder for Noise reduction.** In order to minimize the impact of noise items in users on user sequence intentions, we select k subsequences with the highest noise probability for random reorder. where k is calculated by our reorder coefficient γ, $k = \gamma|Z_u|, 0 < \gamma < 1$.

3.3 Sequence Encoder

Transformer has a good encoding ability for sequence data, and can overcapture the internal relationship between sequences through the self-attention mechanism. It is also widely used as the backbone network for sequential recommendation. Moreover, other sequence encoders are also valid, similar to GRU4Rec, Caser, BERT4Rec.

$$\hat{Z}_u = TranfomerEncoder(Z_u) \tag{7}$$

We follow the common approach of sequential recommendation models and use the last item representation z_u as the representation of the whole sequence.

$$z_u = \hat{Z}_u[-1] \tag{8}$$

3.4 Noise Contrastive Loss

In our data augmentation method, we differ from CL4Rec or DuoRec in that we introduce unique negative data augmentation, which is similar to our idea of contrastive learning by maximizing the difference between positive and negative samples.

Traditional Sequential Recommendation Loss Function. In this paper we adopt cross-entropy [2] as our supervised learning loss function.

$$\mathcal{L}_{seq}\left(s_u\right) = -\log \frac{\exp\left(\operatorname{sim}\left(z_u, z_u^{n+1}\right)\right)}{\sum_{i=1}^{||V||} \exp\left(\operatorname{sim}\left(z_u, z^{v_i}\right)\right)} \tag{9}$$

where z_u is the representation of the user sequence, z_u^{n+1} is the representation of our next item, z^{v_i} is the embedding of all candidate item sets, $||V||$ is the size of the item set.

Positive Contrastive Loss Function. We use a contrastive loss function [6] to calculate whether two positive samples come from the same user history sequence. We minimize positive samples from the same sequence with different augmentations, and maximize the difference between different sequences.

$$\mathcal{L}_{cl}^{+}\left(s_u\right) = -\log \frac{\exp\left(\operatorname{sim}\left(z_u^{a_i}, z_u^{a_j}\right)/\tau\right)}{\exp\left(\operatorname{sim}\left(z_u^{a_i}, z_u^{a_j}\right)/\tau\right) + \sum_{s^- \in S^-} \exp\left(\operatorname{sim}\left(z_u^{a_i}, z^{s^-}\right)/\tau\right)} \tag{10}$$

where $z_u^{a_i}, z_u^{a_j}$ is the representation of user sequence from two noise reduction methods, S^- is the set of negative samples. This negative sample refers to a sample that is augmented from other sequences relative to the current sequence within the same batch.z^{s^-} is the negative sample. τ is temperature coefficient.

Negative Contrastive Loss Function. Our negative samples are the samples we generated by noise augmentations. Our goal is to make noise-augmented samples that are close to each other, and noise-augmented samples that are far from noise-reduced samples.

$$\mathcal{L}_{cl}^{-}\left(s_u\right) = -\frac{1}{|A^-|} \sum_{s_{u'}^{a} \in A^-} \log \frac{\exp\left(\operatorname{sim}\left(z_u^{a^-}, z_{u'}^{a}\right)/\tau\right)}{\exp\left(\operatorname{sim}\left(z_u^{a^-}, z_{u'}^{a}\right)/\tau\right) + \sum_{s \in A^+} \exp\left(\operatorname{sim}\left(z_u^{a^-}, z\right)/\tau\right)} \tag{11}$$

where A^- is the set of sample generated by noise augmentations. A^+ is the set of sample generated by noise reduction. $z_u^{a^-}$ is the representation of user sequence from a noise augmentation method. $z_{u'}^{a}$ is a sample from noise augmentation and z is a sample from noise reduction.

Joint Training. Finally, the loss function of NCL4Rec is to jointly train the cross entropy with the positive loss function and the negative loss function.

$$\mathcal{L}_{NCL4Rec} = \mathcal{L}_{seq} + \lambda_{cl^+}\mathcal{L}^{+}_{\mathrm{cl}^+} + \lambda_{cl^-}\mathcal{L}^{-}_{\mathrm{cl}} \tag{12}$$

where λ_{cl^+} is the coefficient of positive loss function and λ_{cl^-} is the coefficient of negative loss function.

4 Experiment

In order to better compare our experiments, we mainly focus on the following questions.

Q1: How does our NCL4Rec perform compared to other sequential recommendation models?
Q2: How does our NCL4Rec compare to other models in terms of representation learning?

4.1 Setup

Dataset. The datasets we use for sequential recommendation are widely used datasets, namely the Amazon and the MovieLens.

Baselines. The following methods are used for comparison:

- Sequential recommendation model: We use GRU4Rec [4] based on RNN, Caser [11] based on CNN, SASRec [5] based on Transformer.
- Contrastive learning model for sequential recommendation: We use CL4Rec [15] and DuoRec [8].

Metrics. We use top-K Hit Ratio (HR@K) and top-K Normalized Discounted Cumulative Gain (NDCG@K), where K is selected from 5, 10.

4.2 Overall Performance (Q1)

In general, NCL4Rec performs the best on all metrics and datasets. On ML-100K, it outperforms other algorithms by a significant margin in HR@5, HR@10, and NDCG@10, and achieves the highest NDCG@5. Similarly, on Beauty and Sports, NCL4Rec consistently achieves the best performance across all metrics, with improvements ranging from 3.47SASRec and DuoRec also show competitive performance on all datasets. SASRec performs well in HR@5 and NDCG@5 on ML-100K, while DuoRec excels in HR@10 and NDCG@10. Both algorithms perform well on Beauty and Sports. Caser and CL4Rec, however, exhibit relatively suboptimal performance compared to other algorithms across all datasets. Caser consistently performs poorly across all metrics, while CL4Rec has low rankings in HR@5, HR@10, and NDCG@10 on ML-100K and Beauty and Sports.

Overall, these results indicate that NCL4Rec is a promising recommendation algorithm that achieves superior performance across multiple datasets and evaluation metrics (Table 1).

Table 1. Overall performance. (The best results are bolded and the suboptimal ones are underlined. The last column represents the percentage improvement of our results compared to the best results.)

Dataset	Metrics	GRU4Rec	Caser	SASRec	CL4Rec	DuoRec	NCL4Rec	Improv.
ML-100K	HR@5	0.0710	0.0551	0.0764	0.0753	0.0742	**0.0806**	5.50%
	HR@10	0.1295	0.1007	0.1304	0.1326	0.1400	**0.1495**	6.79%
	NDCG@5	0.0384	0.0319	0.0431	0.0450	0.0453	**0.0471**	3.97%
	NDCG@10	0.0574	0.0463	0.0600	0.0631	0.0663	**0.0690**	4.07%
Beauty	HR@5	0.1640	0.0191	0.0365	0.0493	0.0546	**0.0569**	4.21%
	HR@10	0.0365	0.0335	0.0627	0.0807	0.0831	**0.0859**	3.37%
	NDCG@5	0.0086	0.0114	0.0236	0.0166	0.0345	**0.0357**	3.47%
	NDCG@10	0.0142	0.0160	0.0281	0.0312	0.0436	**0.0453**	3.90%
Sports	HR@5	0.0137	0.0121	0.0218	0.0280	0.0310	**0.0332**	7.10%
	HR@10	0.0274	0.0204	0.0336	0.0455	0.0480	**0.0505**	5.21%
	NDCG@5	0.0096	0.0076	0.0087	0.0167	0.0190	**0.0201**	5.79%
	NDCG@10	0.0137	0.0103	0.0224	0.0225	0.0241	**0.0254**	5.39%

4.3 Study of Ablation

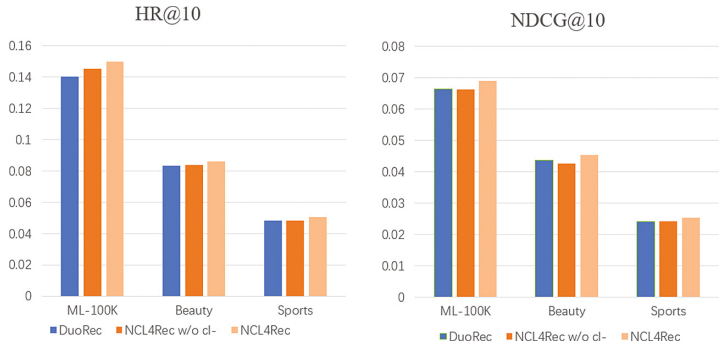

Fig. 2. Performance comparison on DuoRec, NCL4Rec w/o $\mathcal{L}_{cl}^{-}$, NCL4Rec on HR@10, NDCG@10.

To verify the effectiveness of our proposed method, we test the performance of NCL4Rec with different loss functions on three datasets. Additionally, we include DuoRec as a comparison for better observation. Figure 2 shows our results, and it can be seen that when we only use the positive loss function, our method outperforms DuoRec in terms of HR@10 on all three datasets. When using the full loss function, our method shows further improvement. However, on ML-100K, where only positive contrast is used, our method's performance is slightly lower than DuoRec.

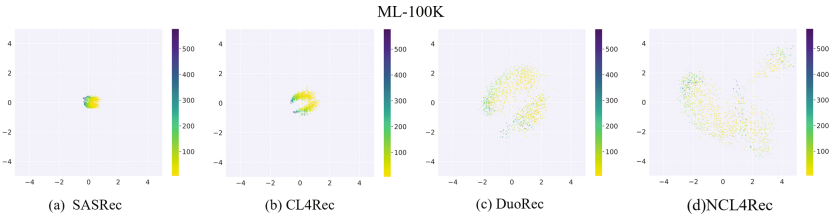

Fig. 3. Item embeddings on ML-100K dataset.

4.4 Discussion About Item Representation (Q3)

Representation learning is always the focus of deep recommendation systems. The embedded representation of items directly determines the performance of recommendation models. Figure 3 show the item embedding representations learned by the four methods of SASRec, CL4Rec, DuoRec, and NCL4Rec on the datasets ML-100K. These four methods all use the transformer as the backbone network. It can be seen that the embedded representations of SASRec are very clustered, followed by CL4Rec. DuoRec uses a contrast regularization method to enhance the uniformity of sequence representation distribution, which has a greater improvement compared to CL4Rec. Our NLC4Rec constructs positive and negative data augmentation to make it easier to distinguish the noise and normal items in the sequence. NCL4Rec can make the embedded representation of items more uniform and more discriminative, and our embedded representation is further improved.

5 Conclusion

In this paper, we investigate how to address the inherently noisy data present in sequence data to optimize our recommendation performance. We introduce supervisory signals to identify noise in raw sequence data, and then design positive and negative augmentations. By pulling in the distance between the positive sample data and widening the distance between the positive sample and the negative sample, we can better learn the representation of the item. Experiments demonstrate that NCL4Rec outperforms state-of-the-art sequence recommendation models on multiple datasets. In future research, we will explore more accurate noise identification methods, so that the inherent noise data in the sequence can be better identified, and the generated samples have better representation capabilities.

Acknowledgement. This work is supported in part by National Natural Science Foundation of China (61702264), the Open Research Project of State Key Laboratory of Novel Software Technology (Nanjing University, No. KFKT2022B28), the National Key R&D Program of China (No. 2020YFB1805503) and the Postdoctoral Science Foundation of China (2019M651835). Dr. Xuyun Zhang is supported only by ARC DECRA Grant DE210101458. Key Technologies and Industrialization of Industrial Internet Terminal Threat Detection and Response System.

References

1. Covington, P., Adams, J., Sargin, E.: Deep neural networks for YouTube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 191–198 (2016)
2. De Boer, P.T., Kroese, D.P., Mannor, S., Rubinstein, R.Y.: A tutorial on the cross-entropy method. Ann. Oper. Res. **134**, 19–67 (2005). https://doi.org/10.1007/s10479-005-5724-z
3. Duan, J., Zhang, P.F., Qiu, R., Huang, Z.: Long short-term enhanced memory for sequential recommendation. World Wide Web **26**(2), 561–583 (2023). https://doi.org/10.1007/s11280-022-01056-9
4. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2016)
5. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 197–206. IEEE (2018)
6. Khosla, P., et al.: Supervised contrastive learning. In: Advances in Neural Information Processing Systems, vol. 33, pp. 18661–18673 (2020)
7. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1419–1428 (2017)
8. Qiu, R., Huang, Z., Yin, H., Wang, Z.: Contrastive learning for representation degeneration problem in sequential recommendation. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp. 813–823 (2022)
9. Sun, F., et al.: BERT4Rec: sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 1441–1450 (2019)
10. Sun, K., Qian, T., Zhong, M., Li, X.: Towards more effective encoders in pre-training for sequential recommendation. World Wide Web 1–32 (2023). https://doi.org/10.1007/s11280-023-01163-1
11. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding, pp. 565–573 (2018)
12. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
13. Wang, G., Wang, H., Liu, J., Yang, Y.: Leveraging the fine-grained user preferences with graph neural networks for recommendation. World Wide Web **26**, 1371–1393 (2023). https://doi.org/10.1007/s11280-022-01099-y
14. Wang, W., Feng, F., He, X., Nie, L., Chua, T.S.: Denoising implicit feedback for recommendation. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 373–381 (2021)
15. Xie, X., et al.: Contrastive learning for sequential recommendation. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 1259–1273. IEEE (2022)