# Research of Sequential Recommendation Algorithm Based on Contrastive Learning and Causal Learning

Tong Wang[1] Yaping Dai[1] Shuai Shao[1*]

School of Automation, Beijing Institute of Technology, Beijing, 100081, China
shaoshuai@bit.edu.cn

**Abstract.** In sequential recommender systems, two primary challenges are the long-tailed distribution of data and data distribution bias. To effectively address these issues, a Contrastive and Causal Learning Algorithm for Sequential Recommendation (C2ASeRec) has been proposed. The algorithm enhances the training efficacy of sequential recommendation models and boosts their performance by introducing environment partition and reweighting, regularization term constraint based on causal learning, and methods to enhance uniformity of representation. These innovations mitigate the performance degradation previously caused by data distribution bias. By concurrently incorporating causal learning-based regularization constraints and representation uniformity enhancement techniques, C2ASeRec demonstrates both universality and robustness across different environment partitioning principles, enabling superior performance in complex real-world scenarios. Experimental results indicate that C2ASeRec achieves outstanding outcomes in addressing data distribution bias. In terms of key performance metrics such as hit rate and normalized discounted cumulative gain, our algorithm significantly surpasses seven previous methods, showcasing exceptional advanced performance.

**Keywords:** Sequential Recommendation · Causal Learning · Data Distribution Bias.

## 1 Introduction

When the sequential recommendation system presents items to users, the display method will also affect the probability of user interaction with the items. This includes whether the item is exposed to the user, the way the item is exposed to the user, the position of the item on the display page, and the comparison with other items. These factors may confuse the user's feedback data, making it

Supported by the National Natural Science Foundation of China under Grant No.82201753.

more difficult to capture the user's preferences. In addition, the sequential recommendation algorithm itself may have some inherent tendencies. For example, algorithms based on popularity tend to recommend popular items and content. The sequential recommendation system is a feedback loop process. It trains the algorithm model from the feedback data, and then provides recommendations to users through the algorithm model. After the user interacts with the system, feedback data is generated. This continuous cycle process will also introduce various biases[1]. These biases may hinder the accurate capture of user preferences and have a significant impact on the performance of the sequential recommendation model, thereby reducing the effectiveness of the recommendation.

Therefore, a reasonable solution to the bias problem in the sequential recommendation system helps to capture user preferences more accurately. However, most current research focuses on using machine learning algorithms to fit user behavior history data, while ignoring the various biases in the observed data caused by factors such as user behavior habits, item display methods, and sequential recommendation algorithm settings. The existence of these biases limits the ability of algorithms that learn directly from data to accurately express user preferences, thereby reducing the performance of sequential recommendation systems. Therefore, how to mitigate the impact of these biases on algorithm learning and thus improve the performance of sequential recommendation systems has attracted widespread attention from academia and industry[2].

The sequential recommendation algorithm is divided into correlation-based recommendation algorithm and causal sequential recommendation algorithm. The sequential recommendation algorithms introduced above all belong to correlation sequential recommendation algorithms, which have the advantages of being simple and convenient. They only need to continuously fit the data to make the algorithm model better match the data. However, it is difficult to evaluate and solve the problem of data distribution deviation based on correlation sequential recommendation algorithms alone, because the correlation relationship cannot infer the causal relationship between input and output, and the deviation may originate from a certain causal relationship stage in the interaction between users and systems. The causal recommendation algorithm emphasizes considering causal relationships rather than just correlation relationships when processing observed data. It can learn invariant features, that is, the fundamental needs of each user's personalization, thereby alleviating the impact and interference caused by different data deviations. Therefore, causal recommendation is more suitable for solving data deviation problems, helping to more accurately locate and solve deviations to eliminate adverse effects and improve the interpretability of sequential recommendation algorithms.

Therefore, in order to solve the problem of data bias more comprehensively, The algorithm will further improve and expand the method in the previous algorithm. We propose a more general and flexible sequential recommendation algorithm (C2ASeRec) that introduces contrastive and causal learning. The algorithm does not rely on a specific model architecture, but starts from the perspective of environment division and model training, and reduces the impact of data bias by optimizing the distribution of data and the training strategy of the model. The algorithm is not only applicable to the long-tail distribution problem discussed in the previous algorithm, but can also deal with other types of data bias problems, and has better generalization and application prospects.

# 2 Design of sequential recommendation algorithm based on contrastive learning and causal learning

## 2.1 Sequential recommendation algorithm structure based on contrastive learning and causal learning

The structure of the C2ASeRec algorithm in The algorithm is shown in **Fig. 1**:

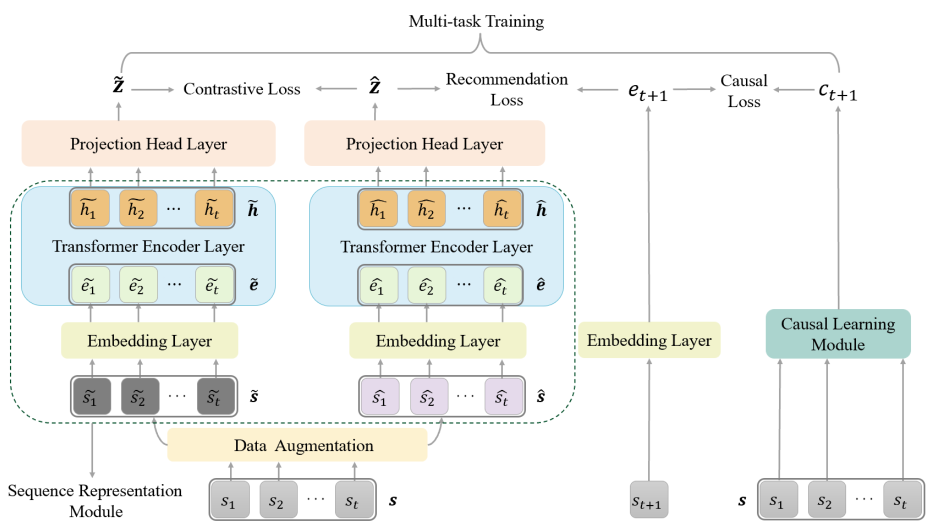


**Fig. 1.** C2ASeRec algorithm structure diagram

In order to solve the problem of data distribution bias in the sequential recommendation task, The algorithm proposes a sequential recommendation algorithm that introduces contrastive learning and causal learning, which is applied to the sequential recommendation task. First, since the long-tail problem of data cannot be ignored, the algorithm in The algorithm retains the structure of the contrastive learning module; secondly, in order for the sequential recommendation algorithm to learn the fundamental needs of users rather than environmental interference factors, The algorithm introduces the causal learning method as an invariant feature extractor; finally, through the multi-task training method, contrastive learning and causal learning are introduced into the sequential recommendation task to achieve loss synchronization optimization.

Specifically, the algorithm structure proposed in The algorithm is divided into three parts: sequential recommendation representation module, contrastive learning module and causal learning module. First, the sequential recommendation representation module consists of an embedding layer and a self-attention mechanism encoding layer to extract deep features of sequential recommendation; second, the contrastive learning module is mainly divided into three parts: adversarial training, projection mapping layer and sequence shuffle data enhancement method, which have been elaborated in detail in the previous algorithm and will not be repeated in this algorithm; finally, the causal learning module consists of three parts: the environment division reweighting method, the regular term constraint method based on causal learning and the representation uniformity enhancement method. The environment division reweighting method is used to optimize the loss function of the algorithm to prevent the algorithm from learning false related features. The regular term constraint method based on causal learning avoids the problem of too large differences in the effect of the algorithm in different environments by adding a variance constraint to the loss, and avoids the negative impact that may be caused by environment division. The uniformity of the representation is enhanced by the representation uniformity method, so that the algorithm can make more stable and consistent predictions when processing data from different distributions, and improve the generalization ability of the algorithm. The following mainly introduces the causal learning module and the corresponding detailed method design in detail.

### 2.2 Causal Learning Module

Since the contrastive learning method has been introduced in the previous algorithm, the following mainly elaborates on the proposed causal invariant learning method. This section first analyzes the data generation process caused by distribution deviation due to confounding factors in the sequential recommendation scenario from the perspective of causal graph, as shown in **Fig. 2**:

### 2.3 Causal Learning Method Process

Although previous studies on sequential recommendation algorithms have achieved remarkable results, these algorithms still have significant performance degradation when faced with data distribution bias. The algorithm proposes a new method to further improve the learning effect of sequence data representation. Specifically, The algorithm introduces three key technical innovations to address these challenges.

(1) Environment partitioning and reweighting method:

It can be noted that when processing sequence data, sequential recommendation algorithms often find it difficult to capture the information of the environment. In order to alleviate the problem of data distribution bias, The algorithm proposes an environment partitioning and reweighting method, and introduces

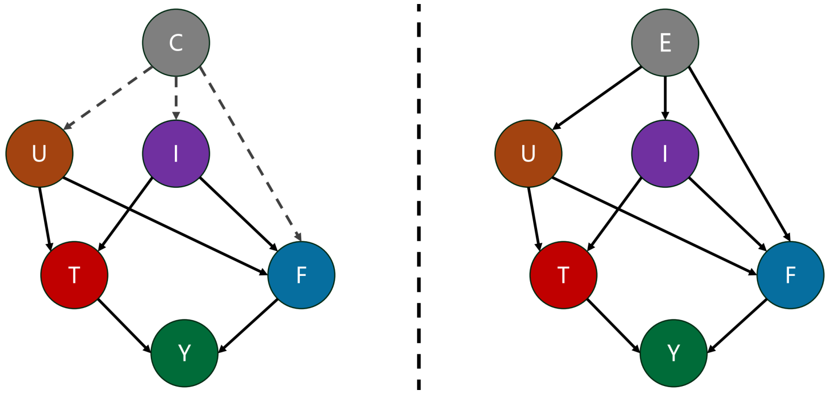


**Fig. 2.** Causal graph of data distribution bias in sequential recommendation

an environment-based loss function for the training process of environment partitioning and reweighting algorithms. The role of the environment loss is to emphasize the importance of learning environment information by the algorithm, thereby helping the algorithm to better capture the contextual information of the data. The introduction of this innovation enables the method in The algorithm to better adapt to different environments and data characteristics when processing sequence data, thereby improving the performance of sequential recommendation.

(2) Regularization term constraint based on causal learning:

Inspired by causal learning theory, The algorithm introduces a regularization term constraint based on causal learning to improve the sequence data representation learned by the algorithm model. The purpose of this constraint is to help the sequential recommendation algorithm better understand the causal relationship in the data, thereby improving its generalization ability. By incorporating causal relationships into the training process, the algorithm in The algorithm can better identify and utilize causal information in the data, further improving the performance of the algorithm.

(3) Enhanced uniformity of representation:

Although the constraint of loss variance helps to alleviate the problems introduced by environment partitioning to some extent, its actual effect is still affected by the design of the environment partitioning principle. Taking unpopular/hot as an example, the algorithm model tends to learn invariant features that are not affected by popularity. However, for other data distribution bias problems, such as class imbalance, random data loss, etc., this division may not be optimal, and the mitigation effect of loss variance constraint is relatively general.

This shows that different invariant features may be required for different types of data distribution bias, and simply based on loss variance constraints cannot completely solve this problem. This is mainly because in actual scenarios, the test distribution is usually unknown, and the demand for invariant features may be different under different data distribution bias problems.

In order to further enhance the uniformity of representation, the algorithm in The algorithm introduces a method of adding random noise to enhance the uniformity of representation to ensure a more uniform representation on the hypersphere. This helps to reduce the impact of the data distribution bias problem and improves the stability and generalization of the algorithm.

Therefore, in order to enhance the prediction effect of the algorithm when facing different data distribution deviation problems, the concept of representation uniformity is introduced. By emphasizing the uniform learning of the algorithm on the input, the algorithm is made more versatile and better adapted to unknown test distributions. The choice of this strategy is intended to enable the algorithm to more comprehensively deal with different types of data distribution deviations and improve the performance of sequence recommendation systems in practical applications.

$$U' = U + e_u, \quad I' = I + e_i \tag{1}$$

The added noise vectors $e_u$ and $e_i$ satisfy the first constraint: s$\|e_u\|_2 = \varepsilon, \|e_i\|_2 = \varepsilon$ and the second constraint: $e_u = \overline{e_u} \odot sign(e_u), e_i = \overline{e_i} \odot sign(e_i), \overline{e_i} \in R^d \sim U(0,1), \overline{e_u} \in R^d \sim U(0,1)$

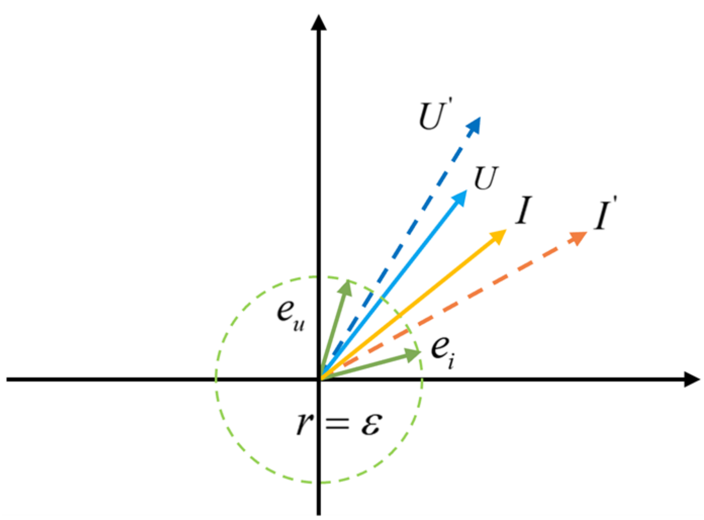


**Fig. 3.** Visualization diagram of the formula corresponding to the method for characterizing uniformity

As shown in the **Fig. 3**, the first constraint controls the size of $e_u$ and $e_i$, which are numerically equivalent to points on a hypersphere with a radius of $\varepsilon$.

The second constraint ensures that $U$, $e_u$ and $e_i$ should be in the same hyperoctave, so that the noise vector and the original representation are in the same hyperquadrant, so that adding noise will not cause a large deviation in $U$, so as to avoid excessive semantic deviation caused by adding noise. The figure shows that by adding the scaled noise vector to the original representation, it can be regarded as rotating the original representation vector $U$ by two small angles $\theta 1$ and $\theta 2$ in space, each rotation corresponds to the deviation of $U$, and realizes the enhancement of the representations $e_u$ and $e_i$. Due to the small rotation angle, the enhanced representation retains most of the information of the original representation and brings semantic differences. And for each node representation, the added random noise is different.

### 2.4 Multi-task training

**Sequence Recommendation Loss Function** The negative log likelihood with sampled softmax is applied as the recommendation loss for each user $u$ at each time step $t+1$ as:

$$\mathcal{L}_{\text{rec}}(s_t) = -\log \frac{\exp\left(s_t^\top v_{t+1}^+\right)}{\exp\left(s_t^\top v_{t+1}^+\right) + \sum_{v_{t+1}^- \in \mathcal{V}^-} \exp\left(s_t^\top v_{t+1}^-\right)}, \tag{2}$$

where $s_{u,t}$represents the predicted user representation, $v_{t+1}^+$ is the item that user $u$ interacts with, and $v_{t+1}^-$ is the randomly sampled negative example item at time step $t+1$.

The algorithm clusters user behaviors across environments to optimize the sequential recommendation loss function and obtain:

$$L_{rec}^{Env} = \sum_{Env \in H} \frac{1}{|D_{Env}|_{(u,i) \in D_{Env}}} L_{rec}\left(\hat{\gamma}_{u,i,Env}, \nu_{u,i}\right) \tag{3}$$

where $H$ represents the divided environment, $D_{Env}$ represents the data from the environment $Env$, and $L_{rec}$ is the loss function of the sequence recommendation task. In this formula, this chapter minimizes the loss function of user and item representations to better fit the data of each environment respectively.

**Contrastive Learning Loss Function** To distinguish whether the two sequential representations come from the same user history sequence, contrastive learning loss is trained iteratively to minimize the differences between different views from the same user history sequence and to maximize the differences between augmentation sequences from different users. Then the data augmentation module is applied to each user's sequence and obtain the augmented sequence. For each user, $(\widetilde{s}_u, \widehat{s}_u)$ is treated as a positive sample pair, and consider the other $2(N-1)$ data augmented examples as negative samples, after which the dot product is utilized to represent the similarity between each sequence, $\operatorname{sim}(u, v) = u^\top v$.

Finally, the contrastive learning loss function is defined similarly to softmax cross entropy loss as:

$$\mathcal{L}_{\text{cl}}(\widetilde{s}, \widehat{s}) = -\log \frac{\exp\big(\text{sim}(\widetilde{s}, \widehat{s})\big)}{\exp\Big(\text{sim}(\widetilde{s}, \widehat{s})\big) + \sum_{s^- \in S^-} \exp\big(\text{sim}(\widetilde{s}, s^-)\Big)}, \tag{4}$$

where sim(.)represents the cosine similarity function, $\widetilde{s}, \widehat{s}$ indicates the hidden representation after data augmentation, $s^-$ is a randomly sampled negative hidden representation. Finally, all $2N$ in-batch losses are averaged to get the final contrastive loss.

**Total Loss Function** In order to combine contrastive learning and sequence recommendation effectively, a multi-task training method is adopted, which is jointly optimized by the sequence recommendation task and the additional contrastive learning task. The total loss is a linear weighted sum as follows:

$$\mathcal{L}_{\text{total}} = L_{rec}^{Env} + \gamma \mathcal{L}_{\text{cl}}, \tag{5}$$

where $\gamma$ is a weighting hyper-parameter.

## 3 Experimental design and results analysis

### 3.1 Introduction to datasets and baseline algorithms

The algorithm conducts experiments on Amazon Beauty, Diginetica, MovieLens-1m, and Yelp public representative datasets, which have significant differences in domain and sparsity. In order to verify the advancement and effectiveness of the algorithm proposed in this algorithm, we conduct a detailed comparison and analysis with a series of representative baseline algorithms.

- **SASRec** [3]. This method stands as one of the leading baselines for addressing sequential recommendation tasks. It leverages self-attention modules to model user sequences and capture their dynamic interests.
- **BERT4Rec** [4]. This approach uses a mask term training method similar to the Bert[5] models in NLP. The backbone is a bidirectional self-attention mechanism
- **CL4SRec** [6]. This technique applies item cropping, masking, and reordering as augmentations for contrastive learning. It is notable for being the first to introduce contrastive learning to sequential recommendation.
- **DuoRec** [7]. This method leverages contrastive learning to tackle the representation degeneration issue in . It integrates contrastive regularization with dropout-based augmentation and supervised positive sampling to create contrastive samples.
- **CT4Rec** [8]. This approach proposes a consistency training method for sequential recommendation tasks, featuring two bidirectional losses. It introduces regularization in the output space by minimizing the bidirectional loss between two different outputs.

- **ICSRec** [9]. This method uniquely extracts coarse-grained intent supervision signals from the historical interaction sequences of all users. These signals are then used to construct two auxiliary learning objectives aimed at enhancing intent representation learning.

### 3.2 Evaluation Metrics

In this paper, the "leave one out" strategy is used to evaluate the performance of each algorithm. For each user, the last item interacted with is retained as the test data, and the items before the last item are used as the validation data. The remaining items are used for training. Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) are common metrics used to evaluate the performance of algorithms proposed in related research. Hit Ratio HR@k can be used to represent the prediction accuracy of items, while Normalized Discounted Cumulative Gain NDCG@k further considers the information of ranking position. The detailed definitions of HR@k and NDCG@k have been explained in detail in the previous algorithm and will not be repeated here.

In this work, k=5, 10 are used to calculate HR@k and NDCG@k. For these metrics, higher values indicate better algorithm performance.

### 3.3 Experimental settings

This paper uses the Adam optimizer to train the algorithm model, with a learning rate of 0.001. During the training process, the batch size is set to 2048 and the number of training epochs is set to 1500.

### 3.4 Experimental results and analysis

In Table 1, the performance of C2ASeRec is compared on four data sets with seven algorithms. The best and second best scores for the metric in each data set are shown in bold and underlined, respectively. The last column shows the percentage improvement of the metric relative to the best baseline algorithm. The table shows that C2ASeRec achieves the best performance on all datasets and performs better than the CFSeRec[10] and the previous state-of-the-art algorithm ICSRec[9]. Although BERT4Rec[4]applies a bidirectional self-attention architecture in sequential recommendation tasks, the performance improvement of BERT4Rec[4] relative to SASRec[3] is not significant and stable. This finding shows that simply optimizing the entire algorithm by the final recommendation goal cannot fully utilize temporal information. For sequential recommendation algorithms using contrastive learning, C2ASeRec generally performs better than previous algorithms. Both C2ASeRec adopts a multi-task learning strategy to utilize contrastive learning self-supervised signals, and the experimental results on the four data sets are always better than the previous algorithms, proving the effectiveness of contrastive learning in improving the performance of sequential recommendation systems. sex. At the same time, C2ASeRec not only adds

**Table 1.** Performance comparison of C2ASeRec and baseline algorithms

| Dataset | Metrics | SASRec | BERT4Rec | CL4SRec | DuoRec | CT4Rec | ICSRec | CFSeRec | C2ASeRec | Improv. |
|---|---|---|---|---|---|---|---|---|---|---|
| Beauty | HR@5 | 0.2109 | 0.2007 | 0.2393 | 0.2436 | 0.2556 | 0.2591 | 0.2757 | **0.2846** | 3.23% |
| | HR@10 | 0.2759 | 0.2727 | 0.2912 | 0.3021 | 0.3200 | 0.3286 | 0.3613 | **0.3771** | 4.37% |
| | NDCG@5 | 0.1523 | 0.1486 | 0.1687 | 0.1733 | 0.1924 | 0.1987 | 0.2015 | **0.2203** | 9.33% |
| | NDCG@10 | 0.1733 | 0.1681 | 0.1854 | 0.1901 | 0.2132 | 0.2212 | 0.2448 | **0.2633** | 7.56% |
| Diginetica | HR@5 | 0.1398 | 0.1573 | 0.1762 | 0.2180 | 0.2572 | 0.2631 | 0.2944 | **0.3061** | 3.97% |
| | HR@10 | 0.2731 | 0.2936 | 0.3216 | 0.3365 | 0.3774 | 0.3882 | 0.4201 | **0.4413** | 5.04% |
| | NDCG@5 | 0.1285 | 0.1327 | 0.1450 | 0.1518 | 0.1729 | 0.1796 | 0.1994 | **0.2101** | 5.37% |
| | NDCG@10 | 0.1416 | 0.1609 | 0.1653 | 0.1735 | 0.1984 | 0.2015 | 0.2376 | **0.2519** | 6.02% |
| ML-1M | HR@5 | 0.1087 | 0.1003 | 0.1147 | 0.2038 | 0.2045 | 0.2081 | 0.2141 | **0.2346** | 9.57% |
| | HR@10 | 0.1593 | 0.1504 | 0.1975 | 0.2946 | 0.2981 | 0.3019 | 0.3187 | **0.3371** | 5.77% |
| | NDCG@5 | 0.0638 | 0.0616 | 0.0662 | 0.1390 | 0.1402 | 0.1428 | 0.1455 | **0.1603** | 10.17% |
| | NDCG@10 | 0.0724 | 0.0701 | 0.0928 | 0.1680 | 0.1699 | 0.1706 | 0.1781 | **0.1933** | 8.53% |
| Yelp | HR@5 | 0.0156 | 0.0161 | 0.0186 | 0.0173 | 0.0216 | 0.0221 | 0.0241 | **0.0266** | 10.37% |
| | HR@10 | 0.0252 | 0.0265 | 0.0291 | 0.0282 | 0.0352 | 0.0379 | 0.0431 | **0.0491** | 13.92% |
| | NDCG@5 | 0.0096 | 0.0102 | 0.0118 | 0.0114 | 0.0130 | 0.0196 | 0.0225 | **0.0286** | 27.11% |
| | NDCG@10 | 0.0129 | 0.0134 | 0.0171 | 0.0163 | 0.0185 | 0.0219 | 0.0286 | **0.0351** | 22.73% |

contrastive learning methods but also causal learning methods, and the experimental results on the four data sets are better than CFSeRec[10]that only adds contrastive learning, reflecting the effectiveness of causal learning methods in improving the performance of sequential recommendation systems. sex. By comparing C2ASeRec with the baseline algorithms introduced in this article, The algorithm finds that C2ASeRec has the best performance on these data sets.

### 3.5 Ablation experiment

**Table 2.** Ablation experiment results of C2ASeRec (NDCG@10)

| Architecture | Beauty | Diginetica | ML-1M | Yelp |
|---|---|---|---|---|
| (0) Default | 0.2633 | 0.2519 | 0.1933 | 0.0299 |
| (1) Remove Reg | 0.2617 | 0.2503 | 0.1912 | 0.0342 |
| (2) Remove Env | 0.2604 | 0.2483 | 0.1886 | 0.0331 |
| (3) Remove Uniformity | 0.2586 | 0.2449 | 0.1850 | 0.0312 |
| (4) Remove Reg and Env | 0.2571 | 0.2436 | 0.1839 | 0.0304 |
| (5) Remove Reg and Uniformity | 0.2536 | 0.2421 | 0.1821 | 0.0296 |
| (6) Remove Env and Uniformity | 0.2481 | 0.2398 | 0.1802 | 0.0289 |

In the causal learning module, the representation uniformity enhancement method, the environment-based reweighting method, and the regularization term constraint method based on causal learning are used. In order to verify the effectiveness of each method proposed in this algorithm, ablation experiments were conducted on four datasets, and the results are shown in Table 2 (0) The default is the complete C2ASeRec algorithm proposed in this paper. (1)(2)(3) respectively represent the experimental results after using two of the causal learning methods, and (4)(5)(6) respectively represent the results after using only one causal learning method. It can be seen from the results in the table that when these components are removed, the performance deteriorates. For the effectiveness ranking of the three causal learning methods in improving the final indicators in the experiment, enhanced uniformity of representation > environment partitioning and reweighting method > regularization term constraint based on causal learning. And the indicators using all three causal learning methods are better than using only one or two methods. The effectiveness of each method proposed in The algorithm is proved by this ablation experiment.

### 3.6 Experiment on the influence and sensitivity of the temperature hyperparameter of the loss function on the algorithm index

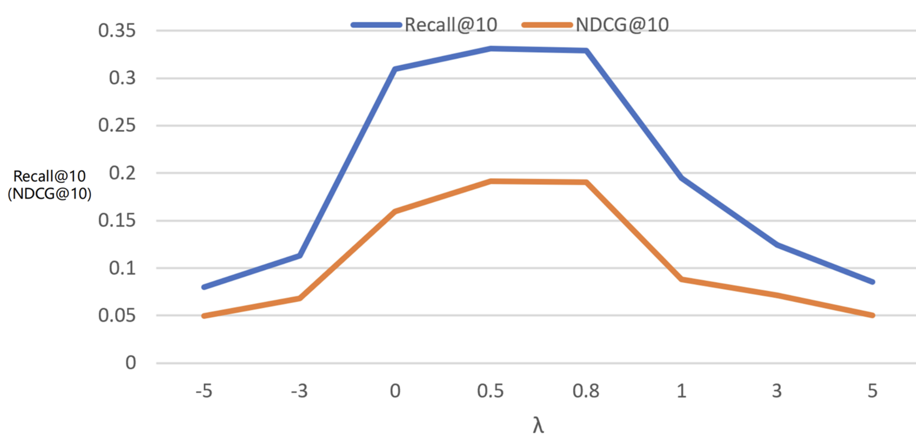


**Fig. 4.** Graph of algorithm index changing with local hyperparameter

This chapter also conducts experiments on the influence and sensitivity of hyperparameters on algorithm indicators. As shown in **Fig. 4** and **Fig. 5**, it is observed that with the increase of hyperparameter $\lambda$, the performance of the algorithm shows a trend of gradually increasing at the beginning and then gradually reaching a peak. Subsequently, the performance begins to decline, indicating that by adjusting $\lambda$, more detailed performance tuning can be achieved. The algorithm shows the best performance when the size of $\lambda$ is between 0 and 1, and the algorithm is relatively insensitive to changes in hyperparameters within an appropriate range.

This trend may be explained by the fact that fine-tuning of hyperparameters can balance the complexity of the algorithm to a certain extent. At the beginning, increasing $\lambda$ may help introduce some constraints and improve the generalization

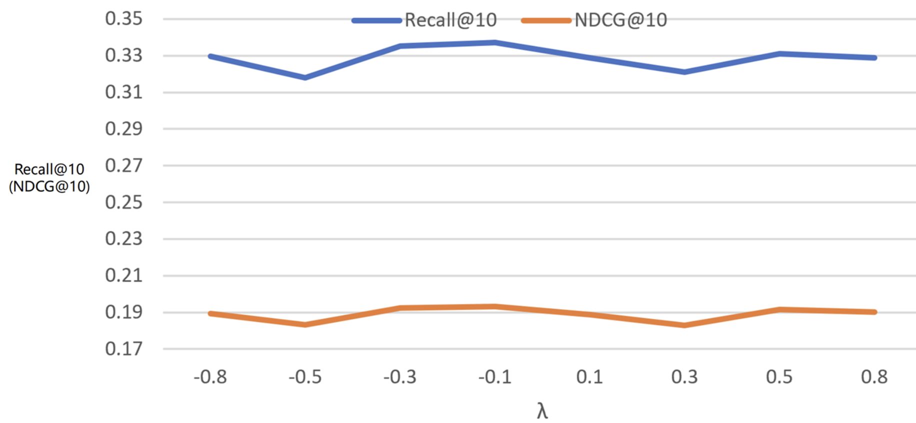


**Fig. 5.** Graph of algorithm indicators changing with global hyperparameters

ability of the algorithm, thereby improving performance. However, when $\lambda$ is too large, it may cause the algorithm to be over-constrained and lose some of its ability to fit the training data, resulting in performance degradation. In the appropriate range of $\lambda$, the algorithm can find a balance that takes into account the fitting of the training data and avoids overfitting to a certain extent.

It is worth noting that the algorithm performs best when the size of the hyperparameter $\lambda$ is between 0 and 1. This may be because within this range, the algorithm can better adapt to the characteristics of the training data while maintaining the ability to generalize to unseen data. The choice of hyperparameters plays a key role in the performance of the algorithm, and the algorithm's insensitivity to $\lambda$ within a certain range shows that the choice within this range is relatively robust.

Overall, this trend in hyperparameter adjustment reveals that fine-tuning the performance of the algorithm needs to take into account the balance between complexity and generalization, and choosing an appropriate hyperparameter range is a key step in improving the performance of the algorithm. Such an adjustment process provides the algorithm with better adaptability, enabling it to perform better in different tasks and data distributions.

### 3.7 Representation uniformity experiment

This experiment deeply explores the reasons why the algorithm proposed in this chapter is superior to the previous algorithms. It can be seen in Figure6 that the visualization result of the representation after uniform enhancement and PCA dimensionality reduction on the right is more uniform and smoother than the visualization result of the representation without uniform enhancement on the left. By uniformly enhancing the representation learned by the algorithm model, the robustness of the algorithm is actually improved. When the representation becomes uniform, the algorithm is more able to capture the key core features

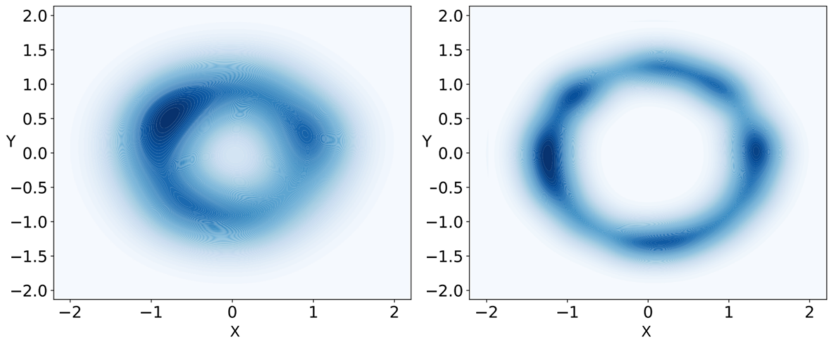


**Fig. 6.** Representation distribution visualization diagram

in the data instead of over-relying on specific data points that may be noisy or abnormal. This uniform representation makes the algorithm more general and more adaptable to different data distributions and input changes. When faced with unseen data or slightly different inputs, the algorithm is less likely to overreact to individual exceptions. Therefore, by enhancing the uniformity of representation, the robustness of the algorithm is effectively improved, making it more adaptable to complex and changeable situations in the real world.

## 4 Conclusion

This paper proposes a new sequential recommendation algorithm that introduces contrastive learning and causal invariant learning, named C2ASeRec, which aims to effectively solve the data distribution deviation problem of sequential recommendation systems existing in previous research. The algorithm successfully improves the effect of sequential recommendation algorithm model training and improves the performance of the sequential recommendation algorithm by introducing environment partitioning and reweighting method, regularization term constraint based on causal learning, and enhanced uniformity of representation. These innovations help solve the performance degradation problem caused by data distribution deviation in previous work. By simultaneously introducing regular term constraints and representation uniformity enhancement methods based on causal learning, the versatility and robustness of the algorithm can be taken into account under different environment division principles, thereby performing better in complex scenes in the real world. Experimental results show that C2ASeRec achieves remarkable results in resolving data distribution deviations. In terms of two key performance indicators, such as hit rate and normalized loss cumulative gain, the algorithm in The algorithm is significantly better than a series of the latest and classic sequential recommendation algorithm baseline algorithms, showing better performance.

## References

1. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant risk minimization. In: arXiv preprint arXiv:1907.02893 (2019)
2. Chang, S., Zhang, Y., Yu, M., Jaakkola, T.: Invariant rationalization. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 1448–1458. PMLR (2020)
3. Kang, W.-C., McAuley, J.: Self-attentive sequential recommendation. In: Proceedings of the IEEE International Conference on Data Mining (ICDM), pp. 197–206. IEEE, Singapore (2018)
4. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM), pp. 1441–1450. ACM, New York (2019)
5. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: arXiv preprint arXiv:1810.04805 (2018)
6. Xie, X., Sun, F., Liu, Z., et al.: Contrastive learning for sequential recommendation. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 1259–1273. IEEE (2022)
7. Qiu, R., Huang, Z., Yin, H., et al.: 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. ACM (2022)
8. Chong, L., Liu, X., Zheng, R., Zhang, L., Liang, X., Li, J., Wu, L., Zhang, M., Lin, L.: CT4Rec: Simple yet Effective Consistency Training for Sequential Recommendation. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 3901–3913. (2023)
9. Qin, X., Yuan, H., Zhao, P., Liu, G., Zhuang, F., Sheng, V. S.: Intent Contrastive Learning with Cross Subsequences for Sequential Recommendation. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM), pp. 548–556. (2024)
10. Wang, T., Dai, Y., Shao, S.: CFSeRec: A Contrastive Framework for Sequential Recommendation. In: 2023 42nd Chinese Control Conference (CCC), pp. 8211–8216. IEEE (2023)