Explanation Guided Contrastive Learning for Sequential Recommendation

Lei Wang
Ee-Peng Lim*
lei.wang.2019@phdcs.smu.edu.sg
eplim@smu.edu.sg
Singapore Management University
Singapore

Zhiwei Liu
Salesforce
USA
zhiweiliu@salesforce.com

Tianxiang Zhao
Penn State University
USA
tkz5084@psu.edu

ABSTRACT

Recently, contrastive learning has been applied to the sequential recommendation task to address data sparsity caused by users with few item interactions and items with few user adoptions. Nevertheless, the existing contrastive learning-based methods fail to ensure that the positive (or negative) sequence obtained by some random augmentation (or sequence sampling) on a given anchor user sequence remains to be semantically similar (or different). When the positive and negative sequences turn out to be false positive and false negative respectively, it may lead to degraded recommendation performance. In this work, we address the above problem by proposing **Explanation Guided Augmentations (EGA)** and **Explanation Guided Contrastive Learning for Sequential Recommendation (EC4SRec)** model framework. The key idea behind EGA is to utilize explanation method(s) to determine items' importance in a user sequence and derive the positive and negative sequences accordingly. EC4SRec then combines both self-supervised and supervised contrastive learning over the positive and negative sequences generated by EGA operations to improve sequence representation learning for more accurate recommendation results. Extensive experiments on four real-world benchmark datasets demonstrate that EC4SRec outperforms the state-of-the-art sequential recommendation methods and two recent contrastive learning-based sequential recommendation methods, CL4SRec and DuoRec. Our experiments also show that EC4SRec can be easily adapted for different sequence encoder backbones (e.g., GRU4Rec and Caser), and improve their recommendation performance.[1]

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

[1]Code is available at https://github.com/demoleiwang/EC4SRec.

*Corresponding Author.

KEYWORDS

Sequential Recommendation, Contrastive Learning, Explanation

ACM Reference Format:
Lei Wang, Ee-Peng Lim*, Zhiwei Liu, and Tianxiang Zhao. 2022. Explanation Guided Contrastive Learning for Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3511808.3557317

1 INTRODUCTION

Background. Recommender systems have played an important role in today's online services [5, 10, 24] to help users navigate the overwhelming amount of information and discover interesting items. Since sequential patterns of user-item interactions change with time, researchers thus [13, 16, 34, 35, 40] pay much attention to sequential recommendation which focuses on characterizing dynamics in user sequences to predict next user-item interaction(s).

For a sequential recommendation method to yield accurate results, it has to learn a high-quality user representation from the user's historical sequence and match the user representation against candidate items. Traditional methods model low-order dependencies between users and items via Markov Chain and Matrix Factorization [12, 29]. Recently, researchers have developed deep learning-based sequential recommendation methods using deep neural networks (such as recurrent neural networks [13], convolutional neural networks [35], transformer [16], and graph neural networks [1]) which learn higher-order dependencies to predict the next items. However, data sparsity is still a major challenge due to limited data about users and items in the long tail. The former refers to many users having very short item sequences. The latter refers to many items having very few user interactions. To cope with these challenges, contrastive learning-based (CL-based) sequential recommendation works [22, 43, 53] incorporate positive and negative views of original user sequences by augmentations and sampling so as to learn more robust user sequence representations, thus more accurately matching candidate items to improve recommendation performance.

Motivating Example. Figure 1 shows an example of the contrastive learning approach to sequential recommendation. From a given user sequence shown in Figure 1(a), we obtain two positive views of the user sequence using some augmentation operator(s), and select the sequence of another user as a negative view. For the positive views, we randomly mask as few items in the given user sequence as shown in Figure 1(b). To learn user sequence

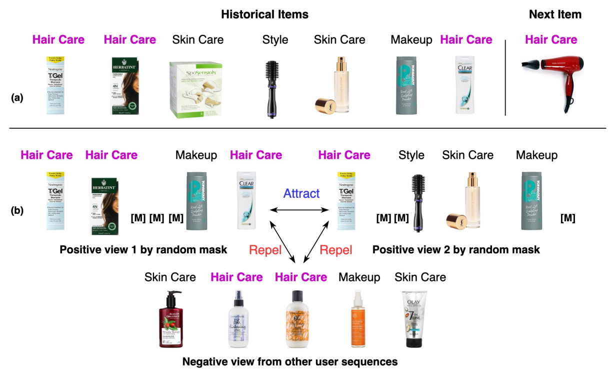

Figure 1: Motivation example: (a) A given user sequence with seven items and a red hair-dryer as the next item; (b) Two positive views generated by random mask operations on the given sequence and a negative view which is the sequence of another user. [M] represents a masked item.

representations, contrastive loss(es) is introduced to make the representations of positive views to be close to each other, but far from that of the negative view [22, 43, 53].

Note that even as CL approach has been shown to improve sequential recommendation performance, its user sequence augmentation and sampling methods are performed with randomness (e.g., random crop, random mask, and random reorder) and is thus prone to produce for a given user sequence positive views that look very different and negative views that look quite similar. As a result, the learned sequence representations are non-ideal reducing the recommendation accuracy. For example, if we were to know that the red hair dryer is the next item, the hair care items in the original sequence will be considered to be more relevant (or important). The positive view 2 in Figure 1(b) which has two hair care items masked however looks quite different from positive view 1. Attracting the representations of positive views 1 and 2 to be closer to each other is therefore inappropriate and may degrade the recommendation performance. By the same reasoning, the negative view may be inappropriately sampled if it shares many hair care items with the two positive views.

Proposed idea. The above motivating example suggests that we need to carefully choose positive and negative views in order to learn good high-quality user sequence representations. To begin this research, we thus conduct a small experiment to show that items important to the next-item of the predicted sequence should be treated differently from non-important items for CL-based sequential recommendation to achieve high accuracy. While this result is interesting, it is infeasible to know which items are important in a user sequence as the next-item is not given during model training. To determine the elusive "important items", we therefore propose **explanation guided augmentations (EGA)** to infer the important items of a given user sequence using explanation methods and consider item importance in augmentation operations. This way, better positive views and negative views can be derived for contrastive learning. We also propose the **Explanation Guided Contrastive Learning for Sequential Recommendation (EC4SRec)** framework to utilise the positive and negative views for self-supervised and supervised learning of user sequence representations, combined with recommendation loss function. EGA and EC4SRec are also designed to accommodate different sequential recommendation backbones. In other words, they can be readily applied to existing self-supervised and supervised CL methods to improve their recommendation performance.

Our contributions. In summary, our contribution is three-fold:

- We propose a model-agnostic Explanation Guided Contrastive Learning for Sequential Recommendation (EC4SRec) framework that incorporates explanation methods into user sequence augmentations for generating positive and negative user sequences for both self-supervised and supervised contrastive learning. EC4SRec can be seen as an integration of CL4SRec and DuoRec, with an additional sampling of negative views for contrastive learning to more effectively separate the representations of positive views from that of the negative views. To our knowledge, EC4SRec is also the first that utilizes explanation methods to improve sequential recommendation.
- We propose several explanation guided augmentation operations to generate both positive and negative user sequences using importance score derived from explanation methods. With these operations, EC4SRec can effectively utilize augmented positive and negative user sequences in contrastive learning to obtain better sequence representations for recommendation.
- We evaluate the proposed augmentation method over two types of contrastive learning frameworks, with three different base sequential recommendation models, on four real-world datasets. The experiment results show that EC4SRec significantly outperforms the vanilla CL4SRec and DuoRec, and other state-of-the-art sequential recommendation methods . We also demonstrate the generalizability of EC4SRec using different sequence encoders and combinations of explanation methods with consistent performance improvement by 4.2% ~23.0%.

2 RELATED WORK

2.1 Sequential Recommendation

Sequential recommendation aims to learn high-quality user and item representations to predict the next item of a given user sequence. Early works focus on modeling low-order transition relationships between items via Markov Chains as item-item features to be used for recommendation [12, 29, 44]. With the advances in neural networks, sequential recommendation research turns to using neural networks [13, 16, 18, 19, 23, 33, 35, 38, 48, 49], such as RNN [13], CNN [35], Transformer [16], and GNN [1] to model high-order sequential dependencies hidden in historical user-item interactions. GRU4Rec [13], for example, incorporates GRU to model sequence-level patterns. This is further improved by replacing GRU by hierarchical RNN [27]. Caser [35] on the other hand uses CNN to model high-order item-item relationships. Inspired by the effectiveness of self-attention in NLP communities [39], Kang, et al. [16] apply self-attention in sequential recommendation named SASRec. GNN based models [22, 42] are also proposed to capture more complex patterns than sequential patterns. To improve sequential recommendation by both performance and interpretability, various works [3, 14, 46] began to determine items contributing to the next-item prediction. Explanation methods, such as attention weights [39], gradient-based methods [32, 47], and Occlusion [30] have been designed to determine features that explain the prediction labels. In our research, we explore the use of explanation methods to determine specific earlier items in the user sequence that explain the predicted next-item and in turn improve sequential recommendation accuracy under the EC4SRec framework.

2.2 Contrastive Learning

Contrastive learning has recently achieved great successes in various research domains including computer vision [2, 9, 11, 25], NLP [6, 7], recommendation [4, 4, 20, 21, 37, 41, 41, 43, 45, 52, 53], etc.. It aims to obtain high-quality representations by pulling positive views of the same instance closer while pushing the positive views and their negative views apart in the representation space. S^{3}Rec [53] pre-trains sequential recommendation by contrastive learning with four self-supervised tasks defined on historical items and their attributes. CL4SRec [43] combines recommendation loss with contrastive loss of self-supervised tasks to optimize the sequential recommendation model. CoSeRec [22] introduces two new augmentation operations, insert and replace, to train robust sequence representations. DuoRec [26] retrieves the positive view of a given user sequence by finding another user's sequence which shares the same next-item in its proposed supervised contrastive learning. In Section 3, we will further elaborate CL4SRec and DuoRec. The above contrastive learning-based sequential recommendation methods, nevertheless, suffer the same pitfalls mentioned in our motivating example. In this research, we therefore seek to address these pitfalls by explanation-guided augmentations and explanation-guided contrastive learning framework.

2.3 Explanation Methods

While there are several works on explainable recommendation [8, 36, 50], they are designed to explain why items are recommended by algorithms. This work mainly focuses on general explanation methods [30, 32, 47] originally designed to determine features that explain the prediction results. By applying these methods to sequential recommendation methods, we can determine historical items in a user sequence that explain the predicted next-item, and assign importance scores to these historical items. For example, Saliency [47] derives an input feature's attribution score by returning the gradient with respect to the input feature. Integrated Gradient [32] takes derivatives of the value for the predicted label with respect to the input features. It outperforms Saliency but is less efficient. Models with attention mechanism provide attention weights as the relative importance of items. However, attention as explanation is controversial [15, 39] since different attention distributions can produce the same model predictions. Occlusion [30] is a perturbation based explanation method which computes input features' attribution scores by the difference between outputs of the original and perturbed input features.

3 PRELIMINARIES

3.1 Problem Formulation

Suppose that we have a set of users $\mathcal{U}$ and items $\mathcal{V}$. For the sequential recommendation task, each user $u \in \mathcal{U}$ has a sequence of items the user has interacted with in the past. We denote this sequence by $s_u = [v_1^u, v_2^u, \ldots, v_{|s_u|}^u]$ where $v_i^u \in \mathcal{V}$ and $|s_u|$ denotes the sequence length. The items in the sequence are ordered by time. The goal of sequential recommendation is to predict the next item at time step, i.e., v_*^u, using the observed historical sequence s_u. Suppose $P(v|s)$ is a model that returns the probability of v being the next item given a sequence s. The sequential recommendation task can be formulated as:

$$v_*^u = \arg\max_{v \in \mathcal{V}} P\Big(v_{|s_u|+1}^u = v \mid s_u\Big).$$

3.2 Contrastive Learning for Sequential Recommendation

In this section, we describe a set of basic augmentation operations to determine positive views of a given user sequence.

Basic Augmentation Operations. There are four basic augmentation operations [26, 43, 53] to generate positive views from an original user sequence, $s_u = [v_1^u, v_2^u, \ldots, v_{|s_u|}^u]$.

- **Random Crop** (crop)**:** It randomly selects a continuous sub-sequence from positions i to $i + l_c$ from s_u and removes it. l_c is defined by $l_c = i + \lfloor \mu_c \cdot |s_u| \rfloor$ where μ_c $(0 < \mu_c \leq 1)$ is a hyper-parameter. The cropped sequence is defined by: $s_u^{\text{c}} = [v_i^u, v_{i+1}^u, \ldots, v_{i+l_c}^u]$.
- **Random Mask** (mask)**:** It randomly selects a proportion μ_m of items from s_u to be masked. Let $g^m(1), g^m(2), \cdots, g^m(n_u^m)$ be the indexes of the items to be masked where $n_u^m = \lfloor \mu_m \cdot |s_u| \rfloor$ and $g^m(x) \in [1, |s_u|]$. An item v_i is replaced with the mask item [m] if selected to be masked. The masked sequence is thus: $s_u^{\text{mask}} = [v_1^u, \cdots, v_{g^m(1)-1}^u, [\text{m}], v_{g^m(1)+1}^u, \cdots, v_{g^m(n_u^m)-1}^u, [\text{m}], v_{g^m(n_u^m)+1}^u, \ldots, v_{|s_u|}^u]$.
- **Random Reorder** (rord)**:** It first randomly selects a continuous sub-sequence $[v_i^u, v_{i+1}^u, \ldots, v_{i+l_r}^u]$ of length $l_r = \lfloor \mu_r * |s_u| \rfloor$ $(0 \leq \mu_r \leq 1)$. It then randomly shuffles the items in the sub-sequence. Suppose the reordered items, sorted by new positions, are $[\tilde{v}_i^u, \ldots, \tilde{v}_{i+l_r}^u]$. The reordered sequence is thus: $s_u^{\text{rord}} = [v_1^u, \cdots, v_{i-1}^u, \tilde{v}_i^u, \tilde{v}_{i+1}^u, \cdots, \tilde{v}_{i+l_r}^u, v_{i+l_r+1}^u, \cdots v_{|s_u|}^u]$.
- **Random Retrieval** (rtrl)**:** This operation randomly selects another user sequence $s_{u'}$ that shares the same target (or next) item as the input sequence s_u, i.e., $v_*^u = v_*^{u'}$. The retrieved sequence is thus: $s_u^{\text{rtrl}} = s_{u'}, s.t.\ v_*^u = v_*^{u'}$

CL4SRec Method. Consider a set of users in a batch $U_B = \{u_1, u_2, \ldots, u_{|U_B|}\}$. The loss function of CL4SRec is:

$$\mathcal{L}_{CL4SRec} = \sum_{u \in U_B} \mathcal{L}_{rec}(s_u) + \lambda \mathcal{L}_{cl}(s_u^{a_i}, s_u^{a_j}). \qquad (1)$$

where $\mathcal{L}_{rec}(s_u)$ and $\mathcal{L}_{cl}(s_u^{a_i}, s_u^{a_j})$ are the recommendation loss and self-supervised contrastive loss respectively. $s_u^{a_i}$ and $s_u^{a_j}$ are positive views of original user sequence s_u after applying augmentations a_i and a_j respectively. a_i and a_j are sampled from $\{\text{crop}, \text{mask}, \text{rord}\}$. We denote the positive view pairs for the users in the batch B as $S_B = \{s_{u_1}^{a_1}, s_{u_1}^{a_2}, s_{u_2}^{a_1}, s_{u_2}^{a2}, \cdots, s_{u_{|B|}}^{a_1}, s_{u_{|B|}}^{a_2}\}$. Thus, the recommendation loss for the user u can be formulated as:

$$\mathcal{L}_{rec}(s_u) = -\log \frac{\exp(sim(h_u, h_{v_*^u}))}{\exp(sim(h_u, h_{v_*^u})) + \sum_{v^- \in V^-} \exp(sim(h_u, h_{v^-}))}, \tag{2}$$

where $V^- = V - \{v_*^u\}$, and h_{v^-} are the representations of the sequence s_u, the next item v_u^*, and a negative item v^- respectively. The contrastive loss is:

$$\mathcal{L}_{cl}(s_u^{a_i}, s_u^{a_j}) = -\log \frac{\exp(sim(h_u^{a_i}, h_u^{a_j}))}{\exp(sim(h_u^{a_i}, h_u^{a_j})) + \sum_{s^- \in S_u^-} \exp(sim(h_u^{a_i}, h^-))}, \tag{3}$$

where $h_u^{a_i}$ and $h_u^{a_j}$ are the representations of s_u after augmentations a_i and a_j respectively. S_u^- denotes a set of negative sequences defined by $S_u^- = S_B - \{s_u^{a1}, s_u^{a2}\}$. s^- and h^- denote a sequence that does not belong to u in the batch B and its representation respectively.

DuoRec Method. Given a user sequence s_u, we randomly sample a *retrieved-positive view* from other users' sequences that share the same next item v_*^u. We denote all user sequences and their corresponding retrieved-positive views by $S = \{s_{u_1}, s_{u_1}^{\mathrm{rtrl}}, s_{u_2}, s_{u_2}^{\mathrm{rtrl}}, \cdots, s_{u_{|B|}}, s_{u_{|U|}}^{\mathrm{rtrl}}\}$. In DuoRec, the representations of each user sequence s_u and its retrieved-positive view s_u^{rtrl} are learned to be close to each other but far from other user sequences and their retrieved-positive views denoted by $S_u^- = S - \{s_u, s_u^{\mathrm{rtrl}}\}$.

The loss function of DuoRec consists of both recommendation loss and supervised contrastive loss functions:

$$\mathcal{L}_{DuoRec} = \sum_{u \in U_B} \mathcal{L}_{rec}(s_u) + \lambda \mathcal{L}_{sl}(s_u) \tag{4}$$

$$\mathcal{L}_{sl}(s_u) = -\Big(\log \frac{\exp(sim(h_u, h_u^{\mathrm{rtrl}})/\tau)}{\exp(sim(h_u, h_u^{\mathrm{rtrl}})/\tau) + \sum_{s^- \in S_u^-} \exp(sim(h_u, h^-)/\tau)} + \log \frac{\exp(sim(h_u^{\mathrm{rtrl}}, h_u)/\tau)}{\exp(sim(h_u^{\mathrm{rtrl}}, h_u)/\tau) + \sum_{s^- \in S_u^-} \exp(sim(h_u^{\mathrm{rtrl}}, h^-)/\tau)}\Big)$$

where τ is the temperature ratio.

3.3 Experiment for Important Item Evaluation

As shown in Figure 1, random augmentation is prone to generate false positive pairs that possibly degrade the quality of learned representations. To evaluate this claim, we conduct an experiment comparing CL4SRec using the vanilla random augmentation operations and augmentation operations that are aware of important items. Our goal is to evaluate if the latter can contribute to better recommendation performance, suggesting that the item importance-aware approach generates higher quality representations.

To verify this assumption empirically, we construct a synthetic dataset[2], which provides ground truth of important items in every user sequence. Specifically, the dataset consists of 500 user sequences each with 10 historical items and 3 additional items at the end serving as the next-items. Among the historical items are 3 important items shared by the 3 next-items to be used for training, validation and test respectively.

We then experiment CL4SRec on this synthetic dataset with two types of mask operations to generate positive views. Each of them masks 4 historical items as follows: (i) *random masking* that randomly masks 4 historical items (4 is empirically chosen); and (ii) *oracle based masking* that masks only unimportant items of the user sequence.

[2]Details of the synthetic data is available at https://github.com/demoleiwang/EC4SRec.

Table 1: Results of CL4SRec on synthetic dataset with ground truth important items.

Masking Op.	Random	Oracle-based
HR@3	0.3560	0.5180
NDCG@3	0.2573	0.3645

As shown in Table 1, CL4SRec using oracle-based masking substantially outperforms that using random masking by both HitRate@3 and NDCG@3. The former achieves more than 40% higher NDCG@3 than the latter. This motivates us to determine important items for effective augmentation and contrastive learning in sequential recommendation.

4 EXPLANATION-GUIDED CONTRASTIVE LEARNING APPROACH

4.1 Proposed Framework

Our proposed **Explanation guided Contrastive Learning Framework for Sequential Recommendation (EC4SRec)**, as shown in Figure 2, consists of a **sequence encoder** to represent a given user's sequence of historical items s_u into a vector representation h_u which is in turn matched with items from a common pool by a **next-item predictor** which returns the next recommended item.

Unlike the existing contrastive learning methods to sequential recommendation (e.g., CL4SRec, DuoRec), EC4SRec utilizes an **explanation method** at scheduled epoch(es) to determine for a user sequence with next-item returned by the sequence encoder and next-item predictor the importance of each s_u's items. Next, the **explanation guided augmentation** will utilize the item importance scores to generate positive and negative views of user sequences for further training the sequence encoder and next-item predictor. The right of Figure 2 shows the different loss and recommendation loss functions that are used to train the models under different explanation-guided contrastive learning methods.

The schedule of explanation method updating the item importance scores, also known as *update schedule*, is controlled by a hyperparameter p. For a model training with a total of N epoches, we schedule the updates to be at epoch $l \cdot \lfloor \frac{N}{p+1} \rfloor$ for $1 \le l \le p$. For example, for $p = 3$ and $N = 100$, updates will scheduled at epochs 25, 50, and 75. For epochs before the first scheduled update (i.e., 1 to $\lfloor \frac{N}{p+1} \rfloor - 1$), EC4SRec can adopt any reasonably good sequential recommendation model (e.g., CL4SRec or DuoRec) to train the initial sequence encoder and next-item predictor. In our experiments, we combine the losses of CL4SRec and DuoRec, i.e., $\sum_{u \in U} \mathcal{L}_{rec}(s_u) + \lambda \mathcal{L}_{cl}(s_u) + \lambda \mathcal{L}_{sl}(s_u)$, to train the initial model. During inference, we only need to feed the input user sequence to the sequence encoder which generates the sequence representation for next-item predictor to recommend the next-item.

4.2 Explanation Guided Importance Scores

General explanation methods, such as Saliency Maps [47], Integrated Gradient [32], and Occlusion [30], are agnostic to sequential

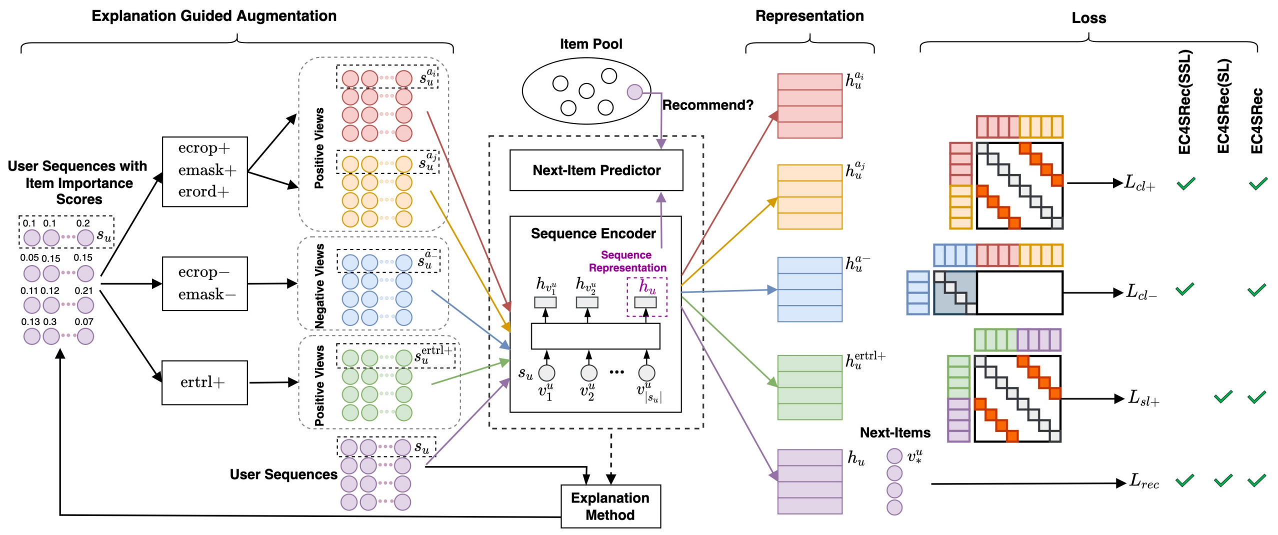

Figure 2: Proposed EC4SRec Framework.

recommendation algorithms, such as GRU4Rec [34], Caser [35], and SASRec [16]. To obtain explanation-guided importance scores for each item in the user sequence, we feed the input user sequence $s_u = [v_1^u, v_2^u, \cdots, v_{|s_u|}^u]$, the sequential encoder *SeqRec*, and its prediction probability for any next item y_u into any model-agnostic explanation method $Expl(\cdot)$, which determine the **importance scores** of items in s_u as $score(s_u) = Expl(y_u, s_u, SeqRec)$, where $score(s_u) = [score(v_1^u), score(v_2^u), \ldots, score(v_{|s_u|}^u)]$ and $score(v_i^u)$ denotes the importance score of item v_i^u.

While any explanation method (e.g., Saliency, Occlusion, and Integrated Gradient) can be used as $Expl(\cdot)$, we use Saliency to illustrate how importance score of each item is derived. Assume that there is an item embedding matrix $E \in \mathbb{R}^{|V| \times d}$, where d is the embedding dimension. The embedding vector $e_{v_i^u} \in \mathbb{R}^d$ of item v_i^u can be derived from E. The importance score of dimension j of $e_{v_i^u}$ can be defined by: $score(e_{v_{i,j}^u}) = \| \frac{\partial y_u}{\partial e_{v_{i,j}^u}} \|$.

By adding and normalizing the importance scores of d dimensions, we obtain the importance score of $e_{v_i^u}$, or $score(e_{v_i^u})$:

$$score(v_i^u) = \frac{\sum_{j=1}^{d} score(e_{v_{i,j}^u})}{\sum_{i'=1}^{|s_u|} \sum_{j=1}^{d} score(e_{v_{i',j}^u})}. \tag{5}$$

$score(v_i^u)$ returns a value in $[0, 1]$ indicating how important is v_i^u in the sequence s_u for a specific given sequential recommendation algorithm. As $\sum_i score(v_i^u) = 1$, the importance score is relative and comparable only among items of the same sequence.

4.3 Explanation Guided Augmentation

We propose five explanation guided augmentation operations, three for generating positive views and two for generating negative views. While some of these operations are extensions of random augmentation, the operations for generating negative views (that is, ecrop− and emask−) are unique to EC4SRec as both CL4SRec and DuoRec consider augmentations for positive views only. Our experiment results in Section 5.3.2 also show that these negative views can substantially improve recommendation accuracy.

- **Explanation Guided Crop for Positive and Negative View** (ecrop+, ecrop−)**:** To perform ecrop+ (or ecrop−) on s_u, we select the k (or $|s_u| - k$) items with the lowest (or highest) by importance score to be removed to generate the positive (or negative) view. k is defined by $\lfloor \mu_e \cdot |s_u| \rfloor$ where μ_e $(0 < \mu_e \leq 1)$. Let $[v_{i_1}^u, \cdots, v_{i_k}^u]$ denote the sub-sequence of k items in s_u with the lowest importance scores. The explanation guided cropped positive and negative views are defined as:

$$s_u^{\text{ecrop+}} = s_u - [v_{i_1}^u, \cdots, v_{i_k}^u], \qquad s_u^{\text{ecrop−}} = [v_{i_1}^u, \cdots, v_{i_k}^u].$$

- **Explanation Guided Mask for Positive or Negative View** (emask+,emask−)**:** To perform emask+ on s_u, we select the k items with the lowest importance scores to be masked. Let $[v_{i_1}^u, \cdots, v_{i_k}^u]$ denote the sub-sequence of k items in s_u with the lowest importance scores. The explanation guided masked positive view is then defined as:

$$s_u^{\text{emask+}} = [v_1^u, \cdots, v_{i_1-1}^u, [\text{m}], v_{i_1+1}^u, \cdots, v_{i_k-1}^u, [\text{m}], v_{i_k+1}^u, \cdots, v_{|s_u|}^u]$$

The explanation guided masked negative view $s_u^{\text{emask−}}$ is defined in a similar way except that the k items to be masked are those with highest importance scores.

- **Explanation Guided Reorder for Positive View** (erord+)**:** Let $[v_{i_1}^u, \cdots, v_{i_k}^u]$ denote the sub-sequence of k items in s_u with the lowest importance scores $(i_1 < i_2 < \cdots < i_k)$. We randomly reorder these items. Suppose the reordered items, sorted by new positions, are $[\tilde{v}_{i_1}^u, \cdots, \tilde{v}_{i_k}^u]$. The reordered positive view can be formulated as:

$$s_u^{\text{erord+}} = [v_1^u, \cdots, v_{i_1-1}^u, \tilde{v}_{i_1}^u, v_{i_1+1}^u, \cdots, v_{i_k-1}^u \tilde{v}_{i_k}^u, v_{i_k+1}^u, \cdots, v_{|s_u|}^u].$$

We leave out explanation guided reorder operation for negative views as it is not likely to generate discriminative negative views.

- **Explanation Guided Retrieval for Positive View** (ertrl+)**:** Like random retrieval, we first define the candidate sequences that share the same target (or next) item as the original user sequence s_u as: $S_u^c = \{s_{u_1}, s_{u_2}, \ldots, s_{u_{|S_u^c|}}\}$. That is, $v_*^{u_k} = v_*^u$, s_{u_k} & $u_k \neq u$. Next, we compute the probability for each sequence in S_u^c using importance scores:

$$P(s_{u_k}) = \frac{util(s_{u_k})}{\sum_{s_{u_j} \in S_u^c} util(s_{u_j})}.$$

where

$$util(s_{u_k}) = \frac{|s_u \cap s_{u_k}|}{|s_u \cup s_{u_k}|} \sum_{v \in s_u \cap s_{u_k}} score(v)$$

We then sample the explanation guided retrieved sequence user sequence $s_u^{\text{ertrl+}}$ from S_u^c with the probability distribution.

4.4 Explanation Guided Contrastive Learning

Based on the EC4SRec framework, we can derive different proposed models depending the type of explanation guided contrastive learning used for model training. In the following, we introduce three proposed models based on explanation guided self-supervised contrastive learning, explanation guided supervised contrastive learning, and combined explanation guided contrastive learning.

*4.4.1 Explanation Guided Self-Supervised Learning (**EC4SRec(SSL)**).* This model can be seen as an extension of CL4SRec with explanation guided augmentation operations generating both positive and negative views for contrastive learning. The loss function consists of three components: (i) *recommendation loss*, (ii) *contrastive loss for explanation guided positive views*, and (iii) *contrastive loss for explanation guided negative views*:

$$\mathcal{L}_{EC4SRec(SSL)} = \sum_{u \in U_B} \mathcal{L}_{rec}(s_u) + \lambda_{cl+}(\mathcal{L}_{cl+}(s_u) + \lambda_{cl-}\mathcal{L}_{cl-}(s_u)). \tag{6}$$

The $\mathcal{L}_{rec}(s_u)$ here has been defined earlier in Equation 2. Let $A^+ = \{a_{\text{ecrop+}}, a_{\text{emask+}}, a_{\text{erord+}}\}$ and $A^- = \{a_{\text{ecrop}-}, a_{\text{emask}-}\}$. To obtain $\mathcal{L}_{cl+}(s_u)$, we generate two positive views $s_u^{a_i}$ and $s_u^{a_j}$ by sampling a_i and a_j ($a_i \neq a_j$) from A^+ and applying them on s_u. We repeat this for all other users and obtain a set of a set of positive views from *all* users denoted as S^+. Let S_u^+ be $\{s_u^{a_i}, s_u^{a_j}\}$. To get the representations of $s_u^{a_i}$ and $s_u^{a_j}$ closer to each other but farther away from other users' positive views, we define:

$$\mathcal{L}_{cl+}(s_u) = -\log \frac{\exp(sim(h_u^{a_i}, h_u^{a_j}))}{\exp(sim(h_u^{a_i}, h_u^{a_j})) + \sum_{s_{u'}^a \in S^+ - S_u^+} \exp(sim(h_u^{a_i}, h_{u'}^a))}$$

To obtain $\mathcal{L}_{cl-}(s_u)$, we generate a negative view s_u^{a-} by applying an augmentation operator $a-$, sampled from A^-, on s_u. Here, we would like this negative view to be closer to other users' negative views (as we do not need distinctive representations for these negative views) and farther away from the all users' positive views, borrowing a similar idea from [17]. Let S^- denote the set of negative views after repeating the above on *all* users. We define:

$$\mathcal{L}_{cl-}(s_u) = -\frac{1}{|S^-| - 1} \sum_{s_{u'}^a \in S^- - \{s_u^{a-}\}} \log \frac{\exp(sim(h_u^{a-}, h_{u'}^a))}{\sum_{s \in S^+ \cup \{s_{u'}^a\}} \exp(sim(h_u^{a-}, h))},$$

where h is the representation of the sequence s. By setting $\beta = 0$, we can obtain a model variant that considers positive views only.

Table 2: Dataset Statistics After Preprocessing.

Dataset	Beauty	Clothing	Sports	ML-1M
Users	22,363	39,387	35,598	6,041
Items	12,101	23,033	18,357	3,417
User-item Interactions	198,502	278,677	296,337	999,611
Avg Sequence Length	8.9	7.1	8.3	165.5
Sparsity	99.93%	99.97%	99.95%	95.16%

*4.4.2 Explanation Guided Supervised Contrastive Learning (**EC4SRec(SL)**).* This model extends DuoRec to use explanation guided augmentation. The loss function is:

$$\mathcal{L}_{EC4SRec(SL)} = \sum_{u \in U_B} \mathcal{L}_{rec}(s_u) + \lambda \mathcal{L}_{sl+}(s_u), \tag{7}$$

where

$$\mathcal{L}_{sl+}(s_u) = - \Bigg(\log \frac{\exp(sim(h_u, h_u^{ertrl+})/\tau)}{\exp(sim(h_u, h_u^{ertrl+})/\tau) + \sum_{s^- \in S_u^-} \exp(sim(h_u, h^-)/\tau)} + \log \frac{\exp(sim(h_u^{ertrl+}, h_u)/\tau)}{\exp(sim(h_u^{ertrl+}, h_u)/\tau) + \sum_{s^- \in S_u^-} \exp(sim(h_u^{ertrl+}, h^-)/\tau)} \Bigg). \tag{8}$$

h_u^{ertrl+} is the representation of the augmented sequence for the user u generated by explanation guided retrieval operation, i.e., ertrl+.

*4.4.3 Combined Explanation Guided Contrastive Learning (**EC4SRec**).* To leverage both self-supervised contrastive learning and supervised contrastive learning, we combine two contrastive learning losses as:

$$\mathcal{L}_{EC4SRec} = \sum_{u \in U_B} \mathcal{L}_{rec}(s_u) + \lambda_{cl+}\mathcal{L}_{cl+}(s_u) + \lambda_{cl-}\mathcal{L}_{cl-}(s_u) + \lambda_{sl+}\mathcal{L}_{sl+}(s_u) \tag{9}$$

5 EXPERIMENT

5.1 Experimental Settings

5.1.1 Datasets and Data Preprocessing. We conduct experiments on four widely used real-world datasets, i.e., Beauty, Clothing, Sports, and ML-1M (Movielens 1M). The first three are from Amazon[3] [24], and ML-1M[4] [10] is a very large benchmark dataset for movie recommendation. Following previous works [16, 26, 43, 53], we remove repeated items, and preprocess four datasets with the 5-core strategy (i.e., removing users and items with fewer than 5 interactions). The dataset statistics are summarized in Table 2. The datasets are very sparse as their sparsity indices (defined by $1 - \frac{\text{num. of interactions}}{\text{num. of users} \cdot \text{num. of items}}$) are very high.

5.1.2 Evaluation Protocols. Following [26, 43], we use the last interacted item of each user sequence for test, the second last item for validation, and all the earlier items for training. The predicted next-item come from the pool of *all* items without any candidate filter. We employ two performance metrics, **Hit Ratio at k (HR@k)** and **Normalized Discounted Cumulative Gain at k (NDCG@k)**,

[3]http://jmcauley.ucsd.edu/data/amazon/

[4]https://grouplens.org/datasets/movielens/1m/

Table 3: Overall Results. (The best and second best results are boldfaced and underlined. *: significant improvement of EC4SRec(SSL) over CL4SRec with p-value= 0.05. **: significant improvement of EC4SRec(SL) over DuoRec with p-value= 0.01.)

		Non-Seq.	Seq. Rec. w/o Contrastive Learning					Seq. Rec. with Contrastive Learning				
Dataset	Metric	BPR-MF	GRU4Rec	Caser	SASRec	BERT4Rec	S^3Rec_{MIP}	CL4SRec	EC4SRec(SSL)**	DuoRec	EC4SRec(SL)*	**EC4SRec**
Beauty	HR@5	0.0120	0.0164	0.0191	0.0365	0.0193	0.0327	0.0495	0.0569	0.0548	**0.0585**	<u>0.0569</u>
	HR@10	0.0299	0.0365	0.0335	0.0627	0.0401	0.0591	0.0810	0.0853	0.0832	**0.0867**	<u>0.0862</u>
	NDCG@5	0.0065	0.0086	0.0114	0.0236	0.0187	0.0175	0.0299	0.0358	0.0345	<u>0.0361</u>	**0.0364**
	NDCG@10	0.0122	0.0142	0.0160	0.0281	0.0254	0.0268	0.0401	0.0450	0.0436	<u>0.0455</u>	**0.0458**
Clothing	HR@5	0.0067	0.0095	0.0049	0.0168	0.0125	0.0163	0.0187	0.0201	0.0196	<u>0.0205</u>	**0.0209**
	HR@10	0.0094	0.0165	0.0092	0.0272	0.0208	0.0237	0.0305	<u>0.0314</u>	0.0296	0.0311	**0.0320**
	NDCG@5	0.0052	0.0061	0.0029	0.0091	0.0075	0.0101	0.0104	0.0113	0.0112	<u>0.0115</u>	**0.0119**
	NDCG@10	0.0069	0.0083	0.0043	0.0124	0.0102	0.0132	0.0142	<u>0.0149</u>	0.0144	<u>0.0149</u>	**0.0155**
Sports	HR@5	0.0092	0.0137	0.0121	0.0218	0.0176	0.0157	0.0277	<u>0.0323</u>	0.0310	0.0317	**0.0331**
	HR@10	0.0188	0.0274	0.0204	0.0336	0.0326	0.0265	0.0455	<u>0.0497</u>	0.0480	0.0491	**0.0514**
	NDCG@5	0.0053	0.0096	0.0076	0.0127	0.0105	0.0098	0.0167	<u>0.0201</u>	0.0191	0.0194	**0.0203**
	NDCG@10	0.0085	0.0137	0.0103	0.0169	0.0153	0.0135	0.0224	<u>0.0256</u>	0.0246	0.0249	**0.0262**
ML-1M	HR@5	0.0164	0.0763	0.0816	0.1087	0.0733	0.1078	0.1583	**0.1699**	0.1672	<u>0.1682</u>	0.1672
	HR@10	0.0354	0.1658	0.1593	0.1904	0.1323	0.1952	0.2423	**0.2543**	0.2507	0.2526	<u>0.2533</u>
	NDCG@5	0.0097	0.0385	0.0372	0.0638	0.0432	0.0616	0.0996	0.1095	0.1076	**0.1104**	<u>0.1102</u>
	NDCG@10	0.0158	0.0671	0.0624	0.0910	0.0619	0.0917	0.1267	0.1368	0.1345	<u>0.1375</u>	**0.1380**

which are widely used in previous work [16, 26, 43, 53]. We report the average of results with running 3 times with 3 random seeds.

5.1.3 Baselines. We compare EC4SRec(SSL), EC4SRec(SL) and EC4SRec with the following three groups of baseline methods:

- **Non-sequential recommendation method**: We use BRP-MF [28], a popular matrix factorization model.
- **Sequential recommendation methods without contrastive learning:** GRU4rec [13], a RNN-based method; Caser [35], a CNN-based method; two self-attention based methods SASRec [16] and BERT4Rec [31]; and a self-supervised learning method S^3Rec_{MIP} [53].
- **Sequential recommendation methods with contrasitve learning**: CL4SRec [43] and DuoRec [26].

5.1.4 Implementation Details. For BPR-MF and S^3Rec_{MIP}, we use results reported by CL4SRec [43]. We implemented the baselines GRU4Rec, Caser, SASRec, and BERT4Rec using the RecBole library [5] [51]. For CL4SRec and DuoRec, we made some changes to the codes provided by the authors of DuoRec to mainly correct some bugs. Our CL4SRec and DuoRec results are generally similar to that reported in the original works. For each baseline, we set the embedding dimension to be 64 and keep all other hyper-parameter settings the same as those reported in their original papers. For EC4SRec and its variants, we use SASRec and Occlusion as the default backbone sequential recommendation method and explanation method respectively. The hyper-parameter settings, such as batch size, embedding dimension, number of layers, number of attention heads, follow those reported in [43]. We tune μ_e, a hyperparameter to control the proportion of important items in augmentation from 0.1 to 0.9 with step size = 0.1. We also tune the temperature τ within $[0.5, 1.0, 1.5]$, and the coefficients λ_{cl+}, λ_{cl-}, and λ_{sl+} within $[0.1, 0.2, 0.3, 0.4, 0.5]$.

5.2 Overall Results

5.2.1 EC4SRec versus Baselines. As shown in Table 3, the overall experiment results show that:

- Our proposed EC4SRec and its variants consistently outperform the state-of-the-art methods, including the latest contrastive learning-based models CL4SRec and DuoRec, for all datasets by all metrics. Specifically, EC4SRec achieves 12.4% (4.9%) improvement over CL4SRec (DuoRec) on average across all datasets by all metrics. EC4SRec generally performs better than EC4SRec(SSL) and EC4SRec(SL). The above findings as well as the significant improvement of EC4SRec(SSL) over CL4SRec and EC4SRec(SL) over DuoRec demonstrate that self-supervised and supervised contrastive learning benefit substantially from explanation guided augmentation. Higher-quality positive views and negative views for contrastive leaning have clearly resulted in better user sequence representations.
- Among the baselines, non-sequential recommendation recommendation methods (i.e., BPR-MF) unexpectedly yield the worst performance. It indicates that the sequential patterns are important in this task. Among the sequential recommendation methods, SASRec and BERT4Rec consistently outperform GRU4Rec and Caser. It shows that self attention can model more complex patterns than left-to-right patterns.
- Consistent with earlier results in [26, 43], contrastive learning methods CL4SRec and DuoRec outperform S^3Rec_{MIP} and SASRec. Our experiment also shows that the former also outperform BERT4Rec. The above demonstrates the the strength of contrastive learning. With supervised contrastive learning, DuoRec performs better than CL4SRec but the gap is reduced between EC4SRec(SL) and EC4SRec(SSL). This could be explained by the additional loss $\mathcal{L}_{cl-}$ added to EC4SRec(SSL).

5.2.2 EC4SRec with Different Backbone Sequential Recommendation Methods. Instead of using the default self-attention based backbone SASRec, we evaluate EC4SRec, its variants, CL4SRec and DuoRec using other backbones, namely RNN-based GRU4Rec and CNN-based Caser to study the impact of explanation guided augmentation and contrastive learning. Due to space constraint, we only show the result on three datasets. As shown in Table 4, the relative performance ranking between EC4SRec, EC4SRec(SSL), EC4SRec(SL), CL4SRec and DuoRec remains unchanged when using GRU4Rec and

[5]https://github.com/RUCAIBox/RecBole

Table 4: Results of EC4SRec with different Sequential Recommendation Backbones.

		Beauty				Clothing				Sports			
Backbone		HR@5	HR@10	NDCG@5	NDCG@10	HR@5	HR@10	NDCG@5	NDCG@10	HR@5	HR@10	NDCG@5	NDCG@10
GRU4Rec	CL4SRec	0.0420	0.0640	0.0270	0.0341	0.0104	0.0180	0.0065	0.0089	0.0244	0.0389	0.0154	0.0200
	EC4SRec(SSL)	0.0461	0.0674	0.0314	0.0382	0.0128	0.0213	0.0082	0.0109	0.0253	0.0396	0.0167	0.0213
	DuoRec	0.0471	0.0689	0.0318	0.0388	0.0118	0.0193	0.0078	0.0102	0.0259	0.0396	0.0163	0.0207
	EC4SRec(SL)	0.0490	0.0717	0.0327	0.0401	0.0130	0.0201	0.0086	0.0108	0.0273	0.0414	0.0173	0.0218
	EC4SRec	**0.0495**	**0.0745**	**0.0332**	**0.0412**	**0.0138**	**0.0218**	**0.0089**	**0.0115**	**0.0276**	**0.0437**	**0.0182**	**0.0233**
Caser	CL4SRec	0.0185	0.0335	0.0108	0.0157	0.0058	0.0100	0.0036	0.0049	0.0113	0.0191	0.0071	0.0096
	EC4SRec(SSL)	0.0228	0.0390	0.0137	0.0189	0.0064	0.0113	0.0039	0.0055	0.014	0.0244	0.0088	0.0121
	DuoRec	0.0207	0.0375	0.0129	0.0183	0.0053	0.0100	0.0031	0.0046	0.0127	0.0215	0.0082	0.0110
	EC4SRec(SL)	0.0262	0.0439	0.0161	0.0218	0.0064	0.0117	0.0039	0.0056	0.0146	0.0240	0.0097	0.0127
	EC4SRec	**0.0269**	**0.0456**	**0.0172**	**0.0232**	**0.0065**	**0.0124**	**0.0041**	**0.0060**	**0.0152**	**0.0266**	**0.0105**	**0.0139**

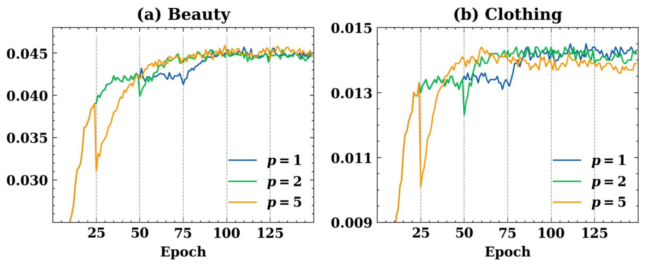

Figure 3: Changes of NDCG@5 for EC4SRec using different update schedules over 150 training epoches (p: number of importance score updates in training)

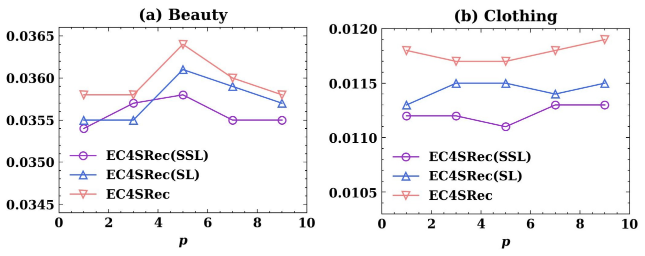

Figure 4: NDCG@5 Results with different update schedule settings (p: number of updates in model training).

Caser backbones. EC4SRec still achieves the best performance using different backbones. EC4SRec(SSL) and EC4SRec(SL) outperforms CL4SRec and DuoRec respectively. These encouraging results indicate the generalizability of the effectiveness of explanation guided approach.

5.3 Detailed Analysis

In this section, we conduct detailed analysis of EC4SRec and its variants. We show the results on Beauty and Clothing datasets only due to space constraint.

5.3.1 Effect of Update Schedule of Important Scores. As mentioned in Section 4.1, the parameter p controls the number of item importance updates scheduled during the training of EC4SRec and its variants. First, we want to study how the updates affect the performance of these models during the training epoches. For illustration, we plot the NDCG@5 of EC4SRec only on validation data in Figure 3. With 150 training epoches, the update occurs only at epoch 75 for $p = 1$, at epoches 50 and 100 for $p = 2$, and at epoches 25, 50, 75, 100 and 125 for $p = 5$. The figure shows that EC4SRec experiences drops of NDCG@5 at the first update. This is because EC4SRec switches from random augmentation and a loss function combining that of CL4SRec and DuoRec to explanation guided augmentation and explanation guided contrastive loss at the first update. EC4SRec however recovers quickly and continues to improve until it converges. Interestingly, the drop in performance is not noticeable for subsequent scheduled updates.

We also show the effect of update schedule on the trained EC4SRec and variants when evaluated against test data in Figure 4. Generally, the NDCG@5 performance does not change much for different p settings. $p = 5$ and $= 9$ yield best results for Beauty and Clothing respectively. As every update incurs additional overheads, there is clearly a trade-off between performance and efficiency when choosing the update schedule which we shall leave to future research.

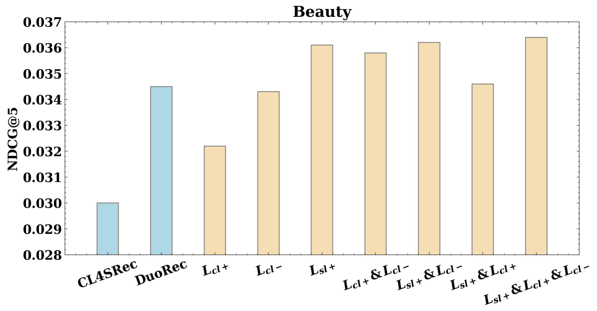

Figure 5: Ablation study of EC4SRec with different combinations of loss functions on Beauty dataset. (EC4SRec results are shown in yellow bars. As $\mathcal{L}_{rec}$ is included by default, EC4SRec(SSL) = EC4SRec with $\mathcal{L}_{cl+} + \mathcal{L}_{cl-}$; EC4SRec(SL) = EC4SRec with $\mathcal{L}_{sl+}$, and EC4SRec = one with all three losses.)

5.3.2 Ablation of Loss Functions. We study the effect of different contrastive losses on EC4SRec performance by evaluating the model using different combinations of losses as shown in Figure 5. The figure shows the NDCG@5 of EC4SRec using recommendation loss $\mathcal{L}_{rec}$ and seven combinations of the three contrastive losses, $\mathcal{L}_{cl+}$, $\mathcal{L}_{cl-}$ and $\mathcal{L}_{sl+}$. For illustration, we conduct this ablation study on Beauty and include CL4SRec and DuoRec for comparison.

Figure 5 shows that EC4SRec with $\mathcal{L}_{cl+}$ and EC4SRec with $\mathcal{L}_{cl-}$ outperform CL4SRec. EC4SRec with $\mathcal{L}_{sl+}$ also outperforms DuoRec. These indicate that each of the 3 explanation guided contrastive losses can effectively improve performance. Moreover, combining

Table 5: NDCG@5 Results of EC4SRec(SSL), abbreviated by E(SSL), with the removal of augmentation operation on Beauty, Clothing and Sports.

	Beauty		Clothing		Sports	
	CL4SRec	E(SSL)	CL4SRec	E(SSL)	CL4SRec	E(SSL)
None	0.0299	0.0358	0.0104	0.0113	0.0167	0.0201
−rord	0.0307	0.0344	0.0103	0.0110	0.0169	0.0181
−mask	0.0311	0.0350	0.0101	0.0116	0.0169	0.0200
−crop	0.0282	0.0353	0.0086	0.0116	0.0147	0.0200

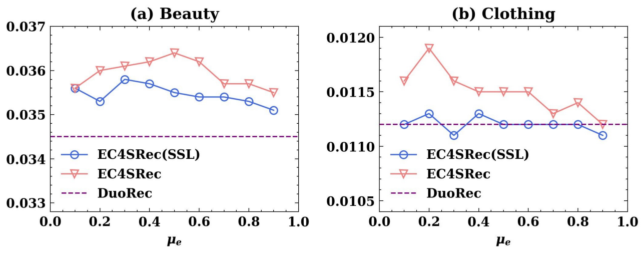

Figure 6: NDCG@5 of EC4SRec with different μ_e settings.

them together can yield even higher performance with the exception of $\mathcal{L}_{sl+} + \mathcal{L}_{cl+}$ which could be explained by having only $\mathcal{L}_{cl+}$ without $\mathcal{L}_{cl-}$ does not help to improve representations much and may conflict with the supervised contrastive learning loss using retrieved positive views.

5.3.3 Influence of Different Augmentation. Our approach consists of four explanation guided augmentations: ecrop, emask, erord, and ertrl. We earlier show that EC4SRec(SL) using explanation guided retrieval (i.e., ertrl) significantly outperforms DuoRec as shown in Table 3. In this section, we evaluate how EC4SRec performs when not using one of the three augmentation operations to investigate the effect of each augmentation operation.

As shown in Table 5, the recommendation accuracy drops substantially when any one of augmentations is removed. Besides, EC4SRec(SSL) consistently achieves better performance than CL4SRec even with one augmentation operation removed. It indicates the effectiveness of each proposed operation. Interestingly, for Clothing dataset, the removal of some augmentation operation can slightly improve the EC4SRec(SSL) performance.

5.3.4 Study of Hyper-Parameter μ_e. The hyper-parameter μ_e determines the number of items with highest scores would be augmented for positive views and negative views under explanation guided augmentation. In this study, we vary μ_e from 0.1 to 0.9 and show the NDCG@5 of EC4SRec and EC4SRec(SSL) on Beauty and Clothing datasets as shown in Figure 6. We observe that μ_e substantially affects the performance of EC4SRec and EC4SRec(SSL). For Beauty dataset, the NDCG@5 of EC4SRec changes from 0.0364 when μ_e = 0.5 to 0.0355 when μ_e = 0.9. Second, EC4SRec and EC4SRec(SSL) perform best on Beauty when μ_e = 0.5 and μ_e = 0.3 respectively. For Clothing dataset, the NDCG@5 of EC4SRec changes from 0.0119 when μ_e = 0.2 to 0.0112 when μ_e = 0.9. And both EC4SRec and EC4SRec(SSL) perform best on Clothing when μ_e = 0.2. These findings indicate that EC4Srec and its variants have different optimal value μ_e on different datasets. Besides, even EC4SRec with the worst performing μ_e still outperforms DuoRec.

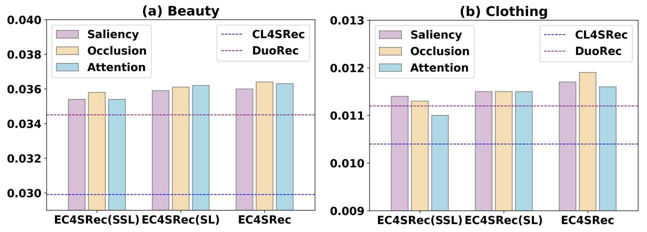

Figure 7: NDCG@5 using different explanation methods.

5.3.5 Effect of Different Explanation Methods. In our earlier results, we use Occlusion as the default explanation method. In this experiment, we evaluate EC4SRec and its variants using other explanation methods for comparison. Figure 7 shows the NDCG@5 results of the above models using Saliency, Occlusion, and Attention based explanation methods on Beauty and Clothing datasets. The NDCG@5 of CL4SRec and DuoRec are also shown as reference.

Figure 7 shows that Occlusion performs well in most cases. The three explanation methods generally help EC4SRec and variants outperform CL4SRec and DuoRec except Attention which could not help EC4SRec(SSL) outperforms DuoRec. Considering robustness, performance, and efficiency, we prefer to use occlusion as the explanation method to get better views for contrastive learning.

6 CONCLUSION

In this paper, we study how to utilize explanation methods to produce high-quality views for contrastive learning in sequential recommendation task. We propose a model-agnostic Explanation Guided Contrastive Learning for Sequential Recommendation (EC4SRec) framework. We introduce several proposed explanation-guided augmentations to generate positive and negative views of given user sequences and propose both self-supervised and supervised contrastive learning. Our extensive experiments on four real-world benchmark datasets demonstrate the effectiveness, generality, and flexibility of our proposed explanation guided approach. Our results also outperform the state-of-the-art contrastive learning based models. To our knowledge, this work represents the first that combine sequential recommendation with explanation methods. For our future research, we will conduct more extensive analysis of the importance score functions and training efficiency. Explanation guided supervised contrastive learning in particular could be slow as it involves selection of retrieved positive views using importance score function. One future research direction is thus to address such overheads by developing appropriate indexing or hashing techniques. One another direction is to meet the challenge of designing augmentations for very long sequential recommendations in contrastive learning.

ACKNOWLEDGEMENT

This research is supported by the National Research Foundation, Singapore under its Strategic Capabilities Research Centres Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

REFERENCES

[1] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 378–387.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[3] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 108–116.

[4] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2022*. 2172–2182.

[5] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[6] Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766* (2020).

[7] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[8] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. 2014. How should I explain? A comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies* 72, 4 (2014), 367–382.

[9] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems* 33 (2020), 21271–21284.

[10] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[12] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 191–200.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[14] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 505–514.

[15] Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186* (2019).

[16] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[17] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems* 33 (2020), 18661–18673.

[18] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[19] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*. 322–330.

[20] Guanyu Lin, Chen Gao, Yinfeng Li, Yu Zheng, Zhiheng Li, Depeng Jin, and Yong Li. 2022. Dual Contrastive Network for Sequential Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2686–2691.

[21] Zhiwei Liu, Yongjun Chen, Jia Li, Man Luo, S Yu Philip, and Caiming Xiong. 2021. Self-supervised Learning for Sequential Recommendation with Model Augmentation. (2021).

[22] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. 2021. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479* (2021).

[23] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S Yu. 2021. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer. In *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*. 1608–1612.

[24] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[25] Xiangyu Peng, Kai Wang, Zheng Zhu, Mang Wang, and Yang You. 2022. Crafting better contrastive views for siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16031–16040.

[26] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 813–823.

[27] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *proceedings of the Eleventh ACM Conference on Recommender Systems*. 130–137.

[28] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[29] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[30] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).

[31] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[32] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR, 3319–3328.

[33] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. 2021. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 598–606.

[34] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 17–22.

[35] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

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

[37] Chenyang Wang, Weizhi Ma, and Chong Chen. 2022. Sequential Recommendation with Multiple Contrast Signals. *ACM Transactions on Information Systems (TOIS)* (2022).

[38] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1101–1110.

[39] Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. *arXiv preprint arXiv:1908.04626* (2019).

[40] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. 495–503.

[41] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 726–735.

[42] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 346–353.

[43] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive learning for sequential recommendation. *arXiv preprint arXiv:2010.14395* (2020).

[44] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation.. In *IJCAI*, Vol. 19. 3940–3946.

[45] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised Learning for Large-scale Item Recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4321–4330.

[46] Jiahao Yuan, Zihan Song, Mingyou Sun, Xiaoling Wang, and Wayne Xin Zhao. 2021. Dual Sparse Attention Network For Session-based Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4635–4643.

[47] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*. Springer, 818–833.

[48] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2021. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 367–377.

[49] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation.. In *IJCAI*. 4320–4326.

[50] Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101.

[51] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4653–4664.

[52] Yu Zheng, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2022. Disentangling Long and Short-Term Interests for Recommendation. In *Proceedings of the ACM Web Conference 2022*. 2256–2267.

[53] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1893–1902.