Intent Contrastive Learning for Sequential Recommendation

Yongjun Chen
Salesforce Research
Palo Alto, CA, USA
yongjun.chen@salesforce.com

Zhiwei Liu
Salesforce Research
Palo Alto, CA, USA
zhiweiliu@salesforce.com

Jia Li
Salesforce Research
Palo Alto, CA, USA
jia.li@salesforce.com

Julian McAuley
UC San Diego
La Jolla, CA, USA
jmcauley@eng.ucsd.edu

Caiming Xiong
Salesforce Research
Palo Alto, CA, USA
cxiong@salesforce.com

ABSTRACT

Users' interactions with items are driven by various intents (e.g., preparing for holiday gifts, shopping for fishing equipment, etc.). However, users' underlying intents are often unobserved/latent, making it challenging to leverage such latent intents for *Sequential recommendation* (SR). To investigate the benefits of latent intents and leverage them effectively for recommendation, we propose ***I**ntent **C**ontrastive **L**earning* (ICL), a general learning paradigm that leverages a latent intent variable into SR. The core idea is to learn users' intent distribution functions from unlabeled user behavior sequences and optimize SR models with contrastive self-supervised learning (SSL) by considering the learnt intents to improve recommendation. Specifically, we introduce a latent variable to represent users' intents and learn the distribution function of the latent variable via clustering. We propose to leverage the learnt intents into SR models via contrastive SSL, which maximizes the agreement between a view of sequence and its corresponding intent. The training is alternated between intent representation learning and the SR model optimization steps within the generalized expectation-maximization (EM) framework. Fusing user intent information into SR also improves model robustness. Experiments conducted on four real-world datasets demonstrate the superiority of the proposed learning paradigm, which improves performance, and robustness against data sparsity and noisy interaction issues [1].

CCS CONCEPTS

• **Information systems** → **Personalization**; **Recommender systems**.

KEYWORDS

Latent Factor Modeling, Self-Supervised Learning, Contrastive Learning, Robustness, Sequential Recommendation

[1]Code is available at https://github.com/salesforce/ICLRec

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512090

ACM Reference Format:
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 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3485447.3512090

1 INTRODUCTION

Recommender systems have been widely used in many scenarios to provide personalized items to users over massive vocabularies of items. The core of an effective recommender system is to accurately predict users' interests toward items based on their historical interactions. With the success of deep learning, deep *Sequential Recommendation* (SR) [13, 35] models, which aims at dynamically characterizing the behaviors of users with different deep neural networks [43, 46], arguably represents the current state-of-the-art [6, 13, 19, 27, 35, 45, 51].

Figure 1: Users' purchasing behaviors can be driven by underlying intents that are not observed.

In general, a deep SR model is trained based on users' interaction behaviors via a deep neural network, assuming users' interests depending on historical behaviors. However, consuming behaviour of users can be affected by other latent factors, i.e., driven by their underlying intents. Consider the example illustrated in Figure 1. Two users purchased a series of different items on Amazon in the past. Given such distinct interaction behaviors, the system will recommend different items to them. However, both of them are fishing enthusiasts and are shopping for fishing activities. As a result, they both purchase 'fishing swivels' in the future. If the system is aware that these two users are shopping for fishing activities, then

commonly purchased items for fishing, such as 'fishing swivels' can be suggested. This motivates us to mine underlying intents that are shared across users and use the learnt intents to guide system providing recommendations.

Precisely discovering the intents of users, however, is under-explored. Most existing works [1, 37] of user intent modeling require side information. ASLI [37] leverages user action types (e.g., click, add-to-favorite, etc.) to capture users' intentions, whereas such information is not always available in system. CoCoRec [1] utilizes item category information. But we argue that categorical feature is unable to accurately represents users' intents. For example, intents like 'shopping for holiday gifts' may involve items from multiple different categories. DSSRec [27] proposes a seq2seq training strategy, which optimizes the intents in latent spaces. However, those intents in DSSRec are inferred solely based on individual sequence representation, while ignoring the underlying correlations of the intents from different users.

Effectively modeling latent intents from user behaviors poses two challenges. First, it is extremely difficult to learn latent intents accurately because we have no labelling data for intents. The only available supervision signals for intents are the user behavior data. Nevertheless, as aforementioned example indicates, distinct behaviors may reflect the same intent. Besides, effectively fusing intent information into a SR model is non-trivial. The target in SR is to predict next items in sequences, which is solved by encoding sequences. Leveraging latent intents of sequences into the model requires the intent factors to be orthogonal to the sequence embeddings, which otherwise would induce redundant information.

To discover the benefits of latent intents and address challenges, we propose the ***I**ntent **C**ontrastive **L**earning* (ICL), a general learning paradigm that leverages the latent intent factor into SR. It learns users' intent distributions from all user behavior sequences via clustering. And it leverages the learnt intents into the SR model via a new contrastive SSL, which maximizes the agreement between a view of sequence and its corresponding intent. The intent representation learning module and the contrastive SSL module are mutually reinforced to train a more expressive sequence encoder. We tackle the challenge of intent mining problem by introducing a latent variable to represent users' intents and learn them alternately along with the SR model optimization through an expectation-maximization (EM) framework to ensure convergence. We suggest fusing learnt intent information into SR via the proposed contrastive SSL, as it can improve model's performance as well as robustness. Extensive experiments conducted on four real-world datasets further verify the effectiveness of the proposed learning paradigm, which improves performance and robustness, even when recommender systems face heavy data sparsity issues.

2 RELATED WORK

2.1 Sequential Recommendation

Sequential recommendation aims to accurately characterize users' dynamic interests by modeling their past behavior sequences [5, 13, 21, 23, 32, 34]. Early works on SR usually model an item-to-item transaction pattern based on Markov Chains [10, 32]. FPMC [34] combines the advantages of Markov Chains and matrix factorization to fuse both sequential patterns and users' general interest. With the recent advances of deep learning, many deep sequential recommendation models are also developed [12, 13, 35, 36]. Such as Convolutional Neural Networks (CNN)-based [36] and RNN-based [12] models. The recent success of Transformer [40] also motivates the developments of pure Transformer-based SR models. SASRec [13] utilizes unidirectional Transformer to assign weights to each interacted item adaptively. BERT4Rec [35] improves that by utilizing a bidirectional Transformer with a *Cloze* task [38] to fuse user behaviors information from left and right directions into each item. LSAN [21] improves SASRec on reducing model size perspective. It proposes a temporal context-aware embedding and twin-attention network, which are light weighted. ASReP [24] further alleviates the data-sparsity issue by leveraging a pre-trained Transformer on the revised user behavior sequences to augment short sequences. In this paper, we study the potential of addressing data sparsity issues and improving SR via self-supervised learning.

2.2 User Intent for Recommendation

Recently, many approaches have been proposed to study users' intents for improving recommendations [3, 17, 18, 41]. MCPRN [41] designs mixture-channel purpose routing networks to adaptively learn users' different purchase purposes of each item under different channels (sub-sequences) for session-based recommendation. MITGNN[25] proposes a multi-intent translation graph neural network to mine users' multiple intents by considering the correlations of the intents. ICM-SR [31] designs an intent-guided neighbor detector to retrieve correct neighbor sessions for neighbor representation. Different from session-based recommendation, another line of works focus on modeling the sequential dynamics of users' interaction behaviors in a longer time span. DSSRec [27] proposes a seq2seq training strategy using multiple future interactions as supervision and introducing an intent variable from her historical and future behavior sequences. The intent variable is used to capture mutual information between an individual user's historical and future behavior sequences. Two users of similar intents might be far away in representation space. Unlike this work, our intent variable is learned over all users' sequences and is used to maximize mutual information across different users with similar learned intents. ASLI [37] captures intent via a temporal convolutional network with side information (e.g., user action types such as click, add-to-favorite, etc.), and then use the learned intents to guide SR model to predict the next item. Instead, our method can learn users' intents based on user interaction data only.

2.3 Contrastive Self-Supervised Learning

Contrastive Self-Supervised Learning (SSL) has brought much attentions by different research communities including CV [2, 4, 9, 14, 20] and NLP [7, 8, 29, 50], as well as recommendation [44, 45, 47, 51]. The fundamental goal of contrastive SSL is to maximize mutual information among the positive transformations of the data itself while improving discrimination ability to the negatives. In recommendation, A two-tower DNN-based contrastive SSL model is proposed in [47]. It aims to improving collaborative filtering based recommendation leveraging item attributes. SGL [44] adopts a multi-task framework with contrastive SSL to improve the graph neural networks (GCN)-based collaborative filtering methods [11,

26, 42, 49] with only item IDs as features. Specific to SR, S^3-Rec [51] adopts a pre-training and fine-tuning strategy, and utilizes contrastive SSL during pre-training to incorporate correlations among items, sub-sequences, and attributes of a given user behavior sequence. However, the two-stage training strategy prevents the information sharing between next-item prediction and SSL tasks and restricts the performance improvement. CL4SRec [45] and CoSeRec [23] instead utilize a multi-task training framework with a contrastive objective to enhance user representations. Different from them, our work is aware of users' latent intent factor when leveraging contrastive SSL, which we show to be beneficial for improving recommendation performance and robustness.

3 PRELIMINARIES

3.1 Problem definition

Assume that a recommender system has a set of users and items denoted by $\mathcal{U}$ and $\mathcal{V}$ respectively. Each user $u \in \mathcal{U}$ has a sequence of interacted items sorted in chronological order $S^u = [s_1^u, \ldots, s_t^u, \ldots, s_{|S^u|}^u]$ where $|S^u|$ is the number of interacted items and s_t^u is the item u interacted at step t. We denote $\mathbf{S}^u$ as embedded representation of S^u, where $\mathbf{s}_t^u$ is the d-dimensional embedding of item s_t^u. In practice, sequences are truncated with maximum length T. If the sequence length is greater than T, the most recent T actions are considered. If the sequence length is less than T, 'padding' items will be added to the left until the length is T [12, 13, 36]. For each user u, the goal of next item prediction task is to predict the next item that the user u is most likely to interact with at the $|S_u| + 1$ step among the item set $\mathcal{V}$, given sequence $\mathbf{S}^u$.

3.2 Deep SR Models for Next Item Prediction

Modern sequential recommendation models commonly encode user behavior sequences with a deep neural network to model sequential patterns from (truncated) user historical behavior sequences. Without losing generality, we define a sequence encoder $f_\theta(\cdot)$ that encodes a sequence $\mathbf{S}^u$ and outputs user interest representations over all position steps $\mathbf{H}^u = f_\theta(\mathbf{S}^u)$. Specially, $\mathbf{h}_t^u$ represents user's interest at position t. The goal can be formulated as finding the optimal encoder parameter θ that maximizes the log-likelihood function of the expected next items of given N sequences on all positional steps:

$$\theta^* = \arg\max_\theta \sum_{u=1}^{N} \sum_{t=2}^{T} \ln P_\theta(s_t^u). \tag{1}$$

which is equivalent to minimizing the adapted binary cross-entropy loss as follows:

$$\mathcal{L}_{\text{NextItem}} = \sum_{u=1}^{N} \sum_{t=2}^{T} \mathcal{L}_{\text{NextItem}}(u, t), \tag{2}$$

$$\mathcal{L}_{\text{NextItem}}(u, t) = -\log(\sigma(\mathbf{h}_{t-1}^u \cdot \mathbf{s}_t^u)) - \sum_{neg} \log(1 - \sigma(\mathbf{h}_{t-1}^u \cdot \mathbf{s}_{neg}^u)), \tag{3}$$

where $\mathbf{s}_t^u$ and $\mathbf{s}_{neg}^u$ denote the embeddings of the target item s_t and all items not interacted by u. The sum operator in Eq. 3 is computationally expensive because $|V|$ is large. Thus we follow [3, 13, 51] to use a sampled softmax technique to randomly sample a negative item for each time step in each sequence. σ is the sigmoid function. And N is refers to the mini-batch size as the SR model.

3.3 Contrastive SSL in SR

Recent advances in contrastive SSL have inspired the recommendation community to leverage contrastive SSL to fuse correlations among different views of one sequence [4, 44, 47], following the mutual information maximization (MIM) principle. Existing approaches in SR can be seen as instance discrimination tasks that optimize a lower bound of MIM, such as InfoNCE [4, 9, 20, 30]. It aims to optimize the proportion of gap of positive pairs and negative pairs [22]. In such an instance discrimination task, sequence augmentations such as 'mask', 'crop', or 'reorder' are required to create different views of the unlabeled data in SR [35, 45, 51, 52]. Formally, given a sequence S^u, and a pre-defined data transformation function set $\mathcal{G}$, we can create two positive views of S^u as follows:

$$\tilde{S}_1^u = g_1^u(S^u), \tilde{S}_2^u = g_2^u(S^u), \text{ s.t. } g_1^u, g_2^u \sim \mathcal{G}, \tag{4}$$

where g_1^u and g_2^u are transformation functions sampled from $\mathcal{G}$ to create a different view of sequence s_u. Commonly, views created from the same sequence are treated as positive pairs, and the views of any different sequences are considered as negative pairs. The augmented views are first encoded with the sequence encoder $f_\theta(\cdot)$ to $\tilde{\mathbf{H}}_1^u$ and $\tilde{\mathbf{H}}_2^u$, and then be fed into an 'Aggregation' layer to get vector representations of sequences, denoted as $\tilde{\mathbf{h}}_1^u$ and $\tilde{\mathbf{h}}_2^u$. In this paper, we 'concatenate' users' interest representations over time steps for simplicity. Note that sequences are prepossessed to have the same length (See Sec. 3.1), thus their vector representations after concatenation have the same length too. After that, we can optimize θ via InfoNCE loss:

$$\mathcal{L}_{\text{SeqCL}} = \mathcal{L}_{\text{SeqCL}}(\tilde{\mathbf{h}}_1^u, \tilde{\mathbf{h}}_2^u) + \mathcal{L}_{\text{SeqCL}}(\tilde{\mathbf{h}}_2^u, \tilde{\mathbf{h}}_1^u), \tag{5}$$

and

$$\mathcal{L}_{\text{SeqCL}}(\tilde{\mathbf{h}}_1^u, \tilde{\mathbf{h}}_2^u) = -\log \frac{\exp(\text{sim}(\tilde{\mathbf{h}}_1^u, \tilde{\mathbf{h}}_2^u))}{\sum_{neg} \exp(\text{sim}(\tilde{\mathbf{h}}_1^u, \tilde{\mathbf{h}}_{neg}))}, \tag{6}$$

where $sim(\cdot)$ is dot product and $\tilde{\mathbf{h}}_{neg}$ are negative views' representations of sequence S^u. Figure 2 (a) illustrates how SeqCL works.

3.4 Latent Factor Modeling in SR

The main goal of next item prediction task is to optimize Eq. (1). Assume that there are also K different user intents (e.g., purchasing holiday gifts, preparing for fishing activity, etc.) in a recommender system that forms the intent variable $c = \{c_i\}_{i=1}^K$, then the probability of a user interacting with a certain item can be rewritten as follows:

$$P_\theta(s^u) = \mathbb{E}_{(c)} \left[P_\theta(s^u, c) \right]. \tag{7}$$

However, users intents are latent by definition. Because of the missing observation of variable c, we are in a 'chicken-and-eggs' situation that without c, we cannot estimate parameter θ, and without θ we cannot infer what the value of c might be.

Later, we will show that a generalized Expectation-Maximization framework provides a direction to address above problem with a convergence guarantee. The basic idea of optimizing Eq. (7) via

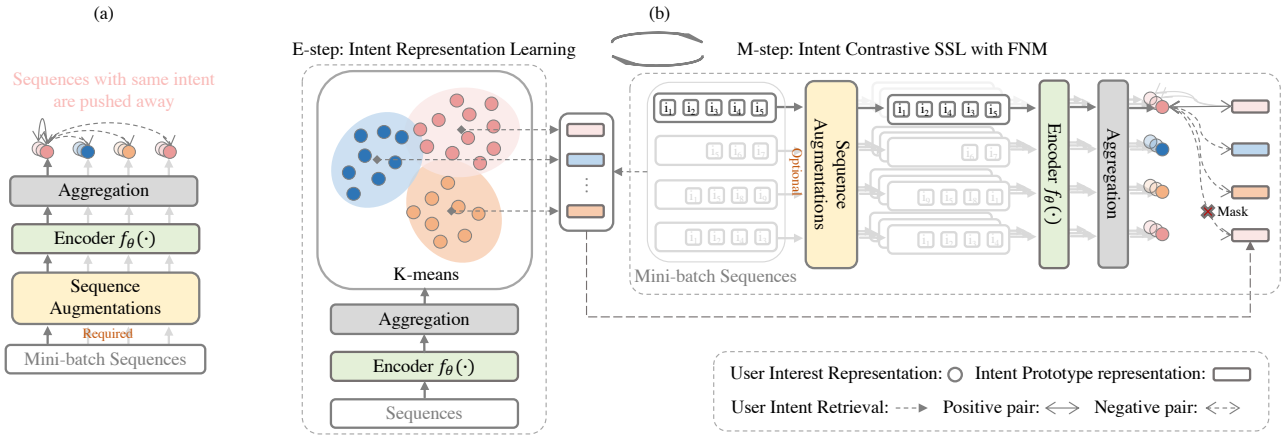

Figure 2: Overview of ICL. (a) An individual sequence level SSL for SR. (b) The proposed ICL for SR. It alternately performs intent representation learning and intent contrastive SSL with FNM within the generalized EM framework to maximizes mutual information (MIM) between a behavior sequence and its corresponding intent prototype.

EM is to start with an initial guess of the model parameter θ and estimate the expected values of the missing variable c, i.e., the E-step. And once we have the values of c, we can maximize the Eq. (7) w.r.t the parameter θ, i.e., the M step. We can repeat this iterative process until the likelihood cannot increase anymore.

4 METHOD

The overview of the proposed ICL within EM framework is presented in Figure 2 (b). It performs E-step and M-step alternately to estimate the distribution function $Q(c)$ over the intent variable c and optimize the model parameters θ. In E-step, it estimates $Q(c)$ via clustering. In M-step, it optimizes θ with considering the estimated $Q(c)$ via mini-batch gradient descent. In each iteration, $Q(c)$ and θ are updated.

In the following sections, we first derive the objective function in order to model the latent intent variable c into an SR model, and how to alternately optimize the objective function w.r.t. θ and estimate the distribution of c under a generalized EM framework in Section 4.1. Then we describe the overall training strategy in Section 4.2. We provide detailed analyses in Section 4.3 followed by experimental studies in Section 5.

4.1 Intent Contrastive Learning

4.1.1 ***Modeling Latent Intent for SR.*** Assuming that there are K latent intent prototypes $\{c_i\}_{i=1}^K$ that affect users' decisions to interact with items, then based on Eq. (1) and (7), we can rewrite objective as follows:

$$\theta^* = \arg\max_\theta \sum_{u=1}^{N}\sum_{t=1}^{T} \ln \mathbb{E}_{(c)}\left[P_\theta(s_t^u, c_i)\right], \tag{8}$$

which is however hard to optimize. Instead, we construct a lower-bound function of Eq. (8) and maximize the lower-bound. Formally, assume intent c follows distribution $Q(c)$, where $\sum_c Q(c_i) = 1$ and $Q(c_i) \geq 0$. Then we have

$$\begin{aligned}\sum_{u=1}^{N}\sum_{t=1}^{T} \ln \mathbb{E}_{(c)}\left[P_\theta(s_t^u, c_i)\right] &= \sum_{u=1}^{N}\sum_{t=1}^{T} \ln \sum_{i=1}^{K} P_\theta(s_t^u, c_i) \\ &= \sum_{u=1}^{N}\sum_{t=1}^{T} \ln \sum_{i=1}^{K} Q(c_i)\frac{P_\theta(s_t^u, c_i)}{Q(c_i)}.\end{aligned} \tag{9}$$

Based on the Jensen's inequality, the term in Eq. (9) is

$$\begin{aligned}&\geq \sum_{u=1}^{N}\sum_{t=1}^{T}\sum_{i=1}^{K} Q(c_i) \ln \frac{P_\theta(s_t^u, c_i)}{Q(c_i)} \\ &\propto \sum_{u=1}^{N}\sum_{t=1}^{T}\sum_{i=1}^{K} Q(c_i) \cdot \ln P_\theta(s_t^u, c_i),\end{aligned} \tag{10}$$

where the $\propto$ stands for 'proportional to' (i.e. up to a multiplicative constant). The inequality will hold with equality when $Q(c_i) = P_\theta(c_i|s_t^u)$. For simplicity, we only focus on last positional step when optimize the lower-bound, which is defined as:

$$\sum_{u=1}^{N}\sum_{i=1}^{K} Q(c_i) \cdot \ln P_\theta(S^u, c_i), \tag{11}$$

where $Q(c_i) = P_\theta(c_i|S^u)$.

So far, we have found a lower-bound of Eq. (8). However, we cannot directly optimize Eq. (11) because $Q(c)$ is unknown. Instead, we alternately optimize the model between the Intent Representation Learning (E-step) and the Intent Contrastive SSL with FNM (M-step), which follows a generalized EM framework. We term the whole processes Intent Contrastive Learning (ICL). In each iteration, $Q(c)$ and the model parameter θ are updated.

4.1.2 ***Intent Representation Learning.*** To learn the intent distribution function $Q(c)$, we encode all the sequences $\{S^u\}_{u=1}^{|\mathcal{U}|}$ with the encoder θ followed by an 'aggregation layer', and then we perform K-means clustering over all sequence representations $\{\mathbf{h}^u\}_{u=1}^{|\mathcal{U}|}$

to obtain K clusters. After that, we can define the distribution function $Q(c_i)$ as follows:

$$Q(c_i) = P_\theta(c_i|S^u) = \begin{cases} 1 & \text{if } S^u \text{ in cluster } i \\ 0 & \text{else.} \end{cases} \tag{12}$$

We denote $\mathbf{c_i}$ as the vector representation of intent c_i, which is the centroid representation of the i^{th} cluster. In this paper, we use 'aggregation layer' to denote the the mean pooling operation over all position steps for simplicity. We leave other advanced aggregation methods such as attention-based methods for future work studies. Figure 2 (b) illustrates how the E-step works.

4.1.3 ***Intent Contrastive SSL with FNM.*** We have estimated the distribution function $Q(c)$. To maximize Eq. (11), we also need to define $P_\theta(S^u, c_i)$. Assuming that the prior over intents follow the uniform distribution and the conditional distribution of S^u given c is isotropic Gaussian with $L2$ normalization, then we can rewrite $P_\theta(S^u, c_i)$ as follows:

$$\begin{aligned} P_\theta(S^u, c_i) &= P_\theta(c_i)P_\theta(S^u|c_i) = \frac{1}{K} \cdot P_\theta(S^u|c_i) \\ &\propto \frac{1}{K} \cdot \frac{\exp(-(\mathbf{h}^u - \mathbf{c}_i)^2)}{\sum_{j=1}^K \exp(-(\mathbf{h}_i^u - \mathbf{c}_j)^2)} \\ &\propto \frac{1}{K} \cdot \frac{\exp(\mathbf{h}^u \cdot \mathbf{c}_i)}{\sum_{j=1}^K \exp(\mathbf{h}^u \cdot \mathbf{c}_j)}, \end{aligned} \tag{13}$$

where $\mathbf{h}^u$ and $\mathbf{c}_u$ are vector representations of S^u and c_i, respectively. Based on Eq. (11), (12), (13), maximizing Eq. (11) is equivalent to minimize the following loss function:

$$-\sum_{v=1}^{N} \log \frac{\exp(\text{sim}(\mathbf{h}^u, \mathbf{c}_i))}{\sum_{j=1}^K \exp(\text{sim}(\mathbf{h}^u, \mathbf{c}_j))}, \tag{14}$$

where $\text{sim}(\cdot)$ is a dot product. We can see that Eq. (14) has a similar form as Eq. (6), where Eq. (6) tries to maximize mutual information between two individual sequences. While Eq. (14) maximizes mutual information between one individual sequence and its corresponding intent. Note that, sequence augmentations are required in SeqCL to create positive views for Eq. (6). While in ICL, sequence augmentations are optional, as the view of a given sequence is its corresponding intent that learnt from original dataset. In this paper, we apply sequence augmentations for enlarging training set purpose and optimize model w.r.t θ based on Eq. (14). Formally, given a batch of training sequences $\{s_u\}_{u=1}^N$, we first create two positive views of a sequence via Eq. (4), and then optimize the following loss function:

$$\mathcal{L}_{\text{ICL}} = \mathcal{L}_{\text{ICL}}(\tilde{\mathbf{h}}_1^u, \mathbf{c}_u) + \mathcal{L}_{\text{ICL}}(\tilde{\mathbf{h}}_2^u, \mathbf{c}_u), \tag{15}$$

and

$$\mathcal{L}_{\text{ICL}}(\tilde{\mathbf{h}}_1^u, \mathbf{c}_u) = -\log \frac{\exp(\text{sim}(\tilde{\mathbf{h}}_1^u, \mathbf{c}_u))}{\sum_{neg} \exp(\text{sim}(\tilde{\mathbf{h}}_1^u, \mathbf{c}_{neg}))}, \tag{16}$$

where c_{neg} are all the intents in the given batch. However, directly optimizing Eq. (16) can introduce false-negative samples since users in a batch can have same intent. To mitigate the effects of false-negatives, we propose a simple strategy to mitigate the effects by not contrasting against them:

$$\mathcal{L}_{\text{ICL}}(\tilde{\mathbf{h}}_1^u, \mathbf{c}_u) = -\log \frac{\exp(\text{sim}(\tilde{\mathbf{h}}_1^u, \mathbf{c}_u))}{\sum_{v=1}^N \mathbb{1}_{v \notin \mathcal{F}} \exp(\text{sim}(\tilde{\mathbf{h}}_1, \mathbf{c}_v))}, \tag{17}$$

where $\mathcal{F}$ is a set of users that have same intent as u in the mini-batch. We term this **False-Negative Mitigation (FNM)**. Figure 2 (b) illustrates how the M-step works.

4.2 Multi-Task Learning

We train the SR model with a multi-task training strategy to jointly optimize ICL via Eq. (17), the main next-item prediction task via Eq. (2) and a sequence level SSL task via Eq. (5). Formally, we jointly train the SR model f_θ as follows:

$$\mathcal{L} = \mathcal{L}_{\text{NextItem}} + \lambda \cdot \mathcal{L}_{\text{ICL}} + \beta \cdot \mathcal{L}_{\text{SeqCL}}, \tag{18}$$

where λ and β control the strengths of the ICL task and sequence level SSL tasks, respectively. Appendix A provides the pseudo-code of the entire learning pipeline. Specially, we build the learning paradigm on Transformer [13, 40] encoder to form the model ICLRec.

ICL is a model-agnostic objective, so we also apply it to S^3-Rec [51] model, which is pre-trained with several $\mathcal{L}_{\text{SeqCL}}$ objectives to capture correlations among items, associated attributes, and subsequences in a sequence and fine-tuned with the $\mathcal{L}_{\text{NextItem}}$ objective, to further verify its effectiveness (see 5.4 for details).

4.3 Discussion

4.3.1 ***Connections with Contrastive SSL in SR.*** Recent methods [45, 51] in SR follow standard contrastive SSL to maximize mutual information between two positive views of sequences. For example, CL4SRec encodes sequences with Transformer and maximizes mutual information between two sequences that are augmented (cropping, masking, or reordering) from the original sequence. However, if the item relationships of a sequence are vulnerable to random perturbation, two views of this sequence may not reveal the original sequence correlations. ICLRec maximizes mutual information between a sequence and its corresponding intent prototype. Since the intent prototype can be considered as a positive view of a given sequence that learnt by considering the semantic structures of all sequences, which reflects true sequence correlations, the ICLRec can outperform CL4SRec consistently.

4.3.2 ***Time Complexity and Convergence Analysis.*** In every iteration of the training phase, the computation costs of our proposed method are mainly from the E-step estimation of $Q(\cdot)$ and M-step optimization of θ with multi-tasks training. For the E-step, the time complexity is $O(|U|mKd)$ from clustering, where d is the dimensionality of the embedding and m is the maximum iteration number in clustering ($m = 20$ in this paper). For the M-step, since we have three objectives to optimize the network $f_\theta(\cdot)$, the time complexity is $O(3 \cdot (|U|^2 d + |U|d^2)$. The overall complexity is dominated by the term $O(3 \cdot (|U|^2 d))$, which is 3 times of Transformer-based SR with only next item prediction objective, e.g., SASRec. Fortunately, the model can be effectively parallelized because f_θ is Transformer and we leave it in future work. In the testing phase, the proposed ICL as well as the SeqCL objectives are no longer needed, which yields the model to have the same time complexity as SASRec ($O(d|V|)$). The empirical time spending comparisons are reported in Sec. 5.2. The convergence of ICL is guaranteed under the generalized EM framework. Proof is provided in Appendix B.

Table 1: Performance comparisons of different methods. The best score is bolded in each row, and the second best is underlined. The last two columns are the relative improvements compared with the best baseline results.

Dataset	Metric	BPR	GRU4Rec	Caser	SASRec	DSSRec	BERT4Rec	S^3-Rec$_{ISP}$	CL4SRec	ICLRec	Improv.
Sports	HR@5	0.0141	0.0162	0.0154	0.0206	0.0214	0.0217	0.0121	0.0217±0.0021	**0.0283**±0.0006	30.48%
	HR@20	0.0323	0.0421	0.0399	0.0497	0.0495	0.0604	0.0344	0.0540±0.0024	**0.0638**±0.0023	18.15%
	NDCG@5	0.0091	0.0103	0.0114	0.0135	0.0142	0.0143	0.0084	0.0137±0.0013	**0.0182**±0.0001	33.33%
	NDCG@20	0.0142	0.0186	0.0178	0.0216	0.0220	0.0251	0.0146	0.0227±0.0016	**0.0284**±0.0008	24.89%
Beauty	HR@5	0.0212	0.0111	0.0251	0.0374	0.0410	0.0360	0.0189	0.0423±0.0031	**0.0493**±0.0013	16.43%
	HR@20	0.0589	0.0478	0.0643	0.0901	0.0914	0.0984	0.0487	0.0994±0.0028	**0.1076**±0.0001	8.30%
	NDCG@5	0.0130	0.0058	0.0145	0.0241	0.0261	0.0216	0.0115	0.0281±0.0018	**0.0324**±0.0017	15.51%
	NDCG@20	0.0236	0.0104	0.0298	0.0387	0.0403	0.0391	0.0198	0.0441±0.0018	**0.0489**±0.0013	10.90%
Toys	HR@5	0.0120	0.0097	0.0166	0.0463	0.0502	0.0274	0.0143	0.0526±0.0034	**0.0590**±0.0012	12.07%
	HR@20	0.0312	0.0301	0.0420	0.0941	0.0975	0.0688	0.0235	0.1038±0.0041	**0.1150**±0.0016	10.74%
	NDCG@5	0.0082	0.0059	0.0107	0.0306	0.0337	0.0174	0.0123	0.0362±0.0025	**0.0403**±0.0002	11.34%
	NDCG@20	0.0136	0.0116	0.0179	0.0441	0.0471	0.0291	0.0162	0.0506±0.0025	**0.0560**±0.0004	10.57%
Yelp	HR@5	0.0127	0.0152	0.0142	0.0160	0.0171	0.0196	0.0101	0.0229±0.0003	**0.0257**±0.0007	12.23%
	HR@20	0.0346	0.0371	0.0406	0.0443	0.0464	0.0564	0.0314	0.0630±0.0009	**0.0677**±0.0016	7.47%
	NDCG@5	0.0082	0.0091	0.008	0.0101	0.0112	0.0121	0.0068	0.0144±0.0001	**0.0162**±0.0003	12.50%
	NDCG@20	0.0143	0.0145	0.0156	0.0179	0.0193	0.0223	0.0127	0.0256±0.0003	**0.0279**±0.0006	8.98%

5 EXPERIMENTS

5.1 Experimental Setting

5.1.1 ***Datasets.*** We conduct experiments on four public datasets. *Sports*, *Beauty* and *Toys* are three subcategories of Amazon review data introduced in [28]. Yelp[2] is a dataset for business recommendation.

We follow [45, 51] to prepare the datasets. In detail, we only keep the '5-core' datasets, in which all users and items have at least 5 interactions. The statistics of the prepared datasets are summarized in Appendix C.

5.1.2 ***Evaluation Metrics.*** We follow [16, 42] to rank the prediction on the whole item set without negative sampling. Performance is evaluated on a variety of evaluation metrics, including *Hit Ratio@k* (HR@k), and *Normalized Discounted Cumulative Gain@k* (NDCG@k) where $k \in \{5, 20\}$.

5.1.3 ***Baseline Methods.*** Four groups of baseline methods are included for comparison.

- **Non-sequential models:** BPR-MF [33] characterizes the pairwise interactions via a matrix factorization model and optimizes through a pair-wise Bayesian Personalized Ranking loss.
- **Standard sequential models.** We include solutions that train the models with a next-item prediction objective. Caser [36] is a CNN-based approach, GRU4Rec [12] is an RNN-based method, and SASRec [13] is one of the state-of-the-art Transformer-based baselines for SR.
- **Sequential models with additional SSL:** BERT4Rec [35] replaces the next-item prediction with a *Cloze* task [38] to fuse information between an item (a view) in a user behavior sequence and its contextual information. S^3-Rec [51] uses SSL to capture correlation-ship among item, sub-sequence, and associated attributes from the given user behavior sequence. Its modules for mining on attributes are removed because we don't have attributes for items, namely S^3-Rec$_{ISP}$. CL4SRec [45] fuses contrastive SSL with a Transformer-based SR model.
- **Sequential models considering latent factors**: We include DSSRec[27], which utilizes seq2seq training and performs optimization in latent space. We do not directly compare ASLI [37], as it requires user action type information (e.g., click, add-to-favorite, etc). Instead, we provide a case study in Sec. 5.6 to evaluate the benefits of the learnt intent factor with additional item category information.

5.1.4 ***Implementation Details.*** Caser[3], BERT4Rec[4], and S3-Rec[5] are provided by the authors. BPRMF[6], GRU4Rec[7], and DSSRec [8] are implemented based on public resources. We implement SASRec and CL4SRec in PyTorch. The mask ratio in BERT4Rec is tuned from $\{0.2, 0.4, 0.6, 0.8\}$. The number of attention heads and number of self-attention layers for all self-attention based methods (SASRec, S^3-Rec, CL4SRec, DSSRec) are tuned from $\{1, 2, 4\}$, and $\{1, 2, 3\}$, respectively. The number of latent factors introduced in DSSRec is tuned from $\{1, 2, \ldots, 8\}$.

Our method is implemented in PyTorch. Faiss [9] is used for K-means clustering to speed up the training and query stages. For the encoder architecture, we set self-attention blocks and attention heads as 2, the dimension of the embedding as 64, and the maximum sequence length as 50. The model is optimized by an Adam

[2]https://www.yelp.com/dataset

[3]https://github.com/graytowne/caser_pytorch
[4]https://github.com/FeiSun/BERT4Rec
[5]https://github.com/RUCAIBox/CIKM2020-S3Rec
[6]https://github.com/xiangwang1223/neural_graph_collaborative_filtering
[7]https://github.com/slientGe/Sequential_Recommendation_Tensorflow
[8]https://github.com/abinashsinha330/DSSRec
[9]https://github.com/facebookresearch/faiss

optimizer [15] with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and batch size of 256. For hyper-parameters of ICLRec, we tune K, λ and β within $\{8, 64, 128, 256, 512, 1024, 2048\}$, $\{0.1, 0.2, \cdots, 0.8\}$, and $\{0.1, 0.2, \cdots, 0.8\}$ respectively. All experiments are run on a single Tesla V100 GPU.

5.2 Performance Comparison

Table 1 shows the results of different methods on all datasets. We have the following observations. First, BPR performs worse than sequential models in general, which indicates the importance of mining the sequential patterns under user behavior sequences. As for standard sequential models, SASRec utilizes a Transformer-based encoder and achieves better performance than Caser and GRU4Rec. This demonstrates the effectiveness of Transformer for capturing sequential patterns. DSSRec further improves SASRec's performance by using a seq2seq training strategy and reconstructs the representation of the future sequence in latent space for alleviating non-convergence problems.

Moreover, though BERT4Rec and S^3-Rec and adopt SSL to provide additional training signals to enhance representations , we observe that both of them exhibit worse performance than SASRec in some datasets (e.g., in the Toys dataset). The reason might be that both BERT4Rec and S^3-Rec aim to incorporate context information of given user behavior sequences via masked item prediction. Such a goal may not align well with the next item prediction target, and it requires that each user behavior sequence is long enough to provide comprehensive 'context' information. Thus their performances are degenerated when most sequences are short. Besides, S^3-Rec is proposed to fuse additional contextual information. Without such features, its two-stage training strategy prevents information sharing between the next-item prediction and SSL tasks, thus leading to poor results. CL4SRec consistently performs better than other baselines, demonstrating the effectiveness of enhancing sequence representations via contrastive SSL on an individual user level.

Finally, ICLRec consistently outperforms existing methods on all datasets. The average improvements compared with the best baseline ranges from **7.47%** to **33.33%** in HR and NDCG. The proposed ICL estimates a good distribution of intents and fuses them into SR model by a new contrastive SSL, which helps the encoder discover a good semantic structure across different user behavior sequences.

We also report the model efficiency on Sports. SASRec is the most efficient solution. It spends 3.59 s/epoch on model updates. CL4SRec and the proposed ICLRec spend 6.52 and 11.75 s/epoch, respectively. In detail, ICLRec spends 3.21 seconds to perform intent representation learning, and rest of 8.54 seconds on multi-task learning. The evaluation times of SASRec, CL4SRec, and ICLRec are about the same(~12.72s over testset) since the introduced ICL task is only used during the training stage.

5.3 Robustness Analysis

Robustness w.r.t. user interaction frequency. The user 'cold-start' problem [1, 48] is one of the typical data-sparsity issues that recommender systems often face, i.e., most users have limited historical behaviors. To check whether ICL improves the robustness under such a scenario, we split user behavior sequences into three groups based on their behavior sequences' length, and keep the total number of behavior sequences the same. Models are trained and evaluated on each group of users independently. Figure 3 shows the comparison results on four datasets. We observe that: (1) The proposed ICLRec can consistently performs better than SASRec among all user groups while CL4SRec fails to outperform SASRec in Beauty and Yelp when user behavior sequences are short. This demonstrates that CL4SRec requires individual user behavior sequences long enough to provide 'complete' information for auxiliary supervision while ICLRec reduces the need by leveraging user intent information, thus consistently benefiting user representation learning even when users have limited historical interactions. (2) Compared with CL4SRec, we observe that the improvement of ICLRec is mainly because it provides better recommendations to users with low interaction frequency. This verifies that user intent information is beneficial, especially when the recommender system faces data-sparsity issues where information in each individual user sequence is limited.

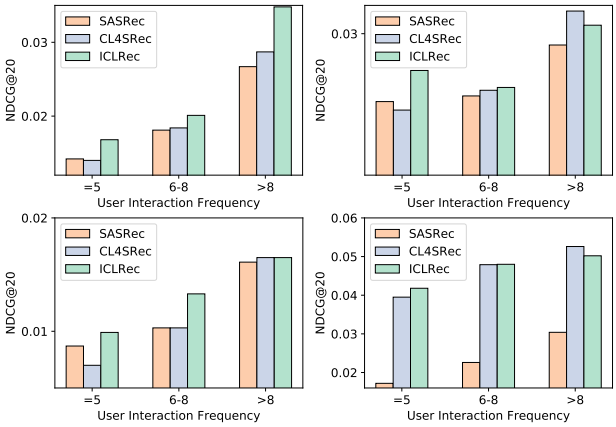

Figure 3: Performance comparison on different user groups among SASRec, CL4SRec and ICLRec (Upper left: Beauty, Upper right: Yelp, Lower left: Sports, Lower right: Toys).)

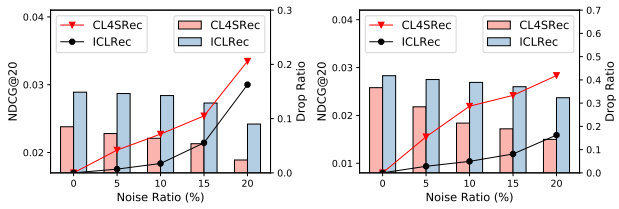

Figure 4: Performance comparison w.r.t. noise ratio on Sports and Yelp. The bar chart shows the performance in NDCG@5 and the line chart shows the corresponding drop rate.

Robustness to Noisy Data. We also conduct experiments on the Sports and Yelp datasets to verify the robustness of ICLRec against noisy interactions in the test phase. Specifically, we randomly add a certain proportion (i.e., 5%, 10%, 15%, 20%) of negative items to text sequences. From Figure 4 we can see that adding noisy

data deteriorates the performance of CL4SRec and ICLRec. However, the performance drop ratio of ICLRec is consistently lower than CL4SRec, and its performance with 15% noise proportion can still outperforms CL4SRec without noisy dataset on Sports. The reason might be the leveraged intent information is collaborative information that distilled from all the users. ICL helps the SR model capture semantic structures from user behavior sequences, which increases the robustness of ICLRec to noisy perturbations on individual sequences.

Table 2: Ablation study of ICLRec (NDCG@20).

Model	Dataset			
	Sports	Beauty	Toys	Yelp
(A) ICLRec	**0.0287**	**0.0480**	**0.0554**	**0.0283**
(B) w/o FNM	0.0283	0.0465	0.0524	0.0266
(C) only ICL	0.0263	0.0429	0.0488	0.0267
(D) w/o ICL	0.0238	0.0428	0.0505	0.0258
(E), is (C) w/o seq. aug	0.0242	0.0414	0.0488	0.0213
(F) SASRec	0.0216	0.0387	0.0441	0.0179
(G) ICL + S^3-Rec$_{ISP}$	**0.0157**	**0.0264**	**0.0266**	**0.0205**
(H) S^3-Rec$_{ISP}$	0.0146	0.0198	0.0162	0.0127

5.4 Ablation Study

Our proposed ICLRec contains a novel ICL objective, a false-negative noise mitigation (FNM) strategy, a SeqCL objective, and sequence augmentations. To verify the effectiveness of each component, we conduct an ablation study on four datasets and report results in Table 2. (A) is our final model, and (B) to (F) are ICLRec removed certain components. From (A)-(B) we can see that the FNM leverages the learned intent information to avoid users with similar intents pushing away in their representation space which helps the model to learn better user representations. Compared with (A)-(D), we find that without the proposed ICL, the performance drops significantly, which demonstrates the effectiveness of ICL. Compared with (A)-(C), we find that individual user level mutual information also helps to enhance user representations. As we analyze in Sec. 5.3, it contributes more to long user sequences. Compared with (E)-(F), we find that ICL can perform contrastive SSL without sequence augmentations and outperforms SASRec. While CL4SRec requires the sequence augmentation module to perform contrastive SSL. Comparison between (C) and (E) indicates sequence augmentation enlarges training set, which benefits improving performance.

Since ICL is a model-agostic learning paradigm, we also add ICL to the S^3-Rec$_{ISP}$ [51] model in the fine-tuning stage to further verify its effectiveness. Results are shown in Table. 2 (G)-(H). We find that the S^3-Rec$_{ISP}$ model also benefits from the ICL objective. The average improvement over the four datasets is 41.11% in NDCG@20, which further validate the effectiveness and practicality of ICLRec.

5.5 Hyper-parameter Sensitivity

We next move to investigate the impact of hyper-parameters: the intent class number K, the strength of ICL objective λ, the batch size, and the strength of SeqCL objective β.

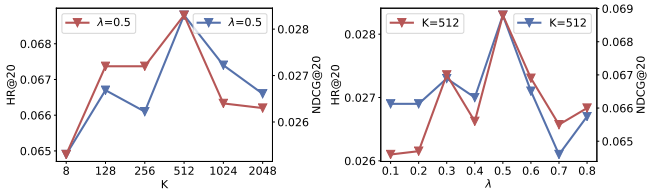

Figure 5: Impact of intent class numbers K and the intent contrastive learning strength λ on Yelp.

The larger value K means users can have more diverse intentions. The larger value λ means the ICL task contributes more to the final model. We conduct experiments on Yelp and show the results in Figure 5. We find that: (1) ICLRec reaches its best performance when increasing K to 512, and then it starts to deteriorate as K become larger. When K is very small, the number of users under each intent prototype can potentially be large. As a result, false-positive samples (i.e., users that actually have different intents are considered as having the same intent erroneously) are introduced to the contrastive SSL, thus affecting learning. On the other hand, when K is too large, the number of users under each intent prototype is small, the introduced false-negative samples will also impair contrastive SSL. In Yelp, 512 user intents summarize users' distinct behaviors best. (2) A 'sweet-spot' of $\lambda = 0.5$ can also be found. It indicates that the ICL task can benefit the recommendation prediction as an auxiliary task. The impact of the batch size and the strength of SeqCL task β is provided in Appendix D.

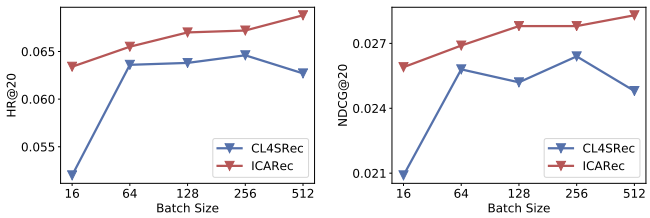

Figure 6: Performance comparison w.r.t. Batch Size.

5.6 Case Study

The Sports dataset [28] contains 2,277 fine-grained item categories, and the Yelp dataset provides 1,001 business categories. We utilize these attributes to study the effectiveness of the proposed ICLRec both quantitatively and qualitatively. Note that we did not use this information during the training phrase. The detailed analysis results are in Appendix E.

6 CONCLUSION

In this work, we propose a new learning paradigm ICL that can model latent intent factors from user interactions and fuse them into a sequential recommendation model via a new contrastive SSL objective. ICL is formulated within an EM framework, which guarantees convergence. Detailed analyses show the superiority of ICL and experiments conducted on four datasets further demonstrate the effectiveness of the proposed method.

REFERENCES

[1] Renqin Cai, Jibang Wu, Aidan San, Chong Wang, and Hongning Wang. 2021. Category-aware Collaborative Sequential Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 388–397.

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882* (2020).

[3] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2942–2951.

[4] 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.

[5] Yongjun Chen, Jia Li, Chenghao Liu, Chenxi Li, Markus Anderle, Julian McAuley, and Caiming Xiong. 2021. Modeling Dynamic Attributes for Next Basket Recommendation. *arXiv preprint arXiv:2109.11654* (2021).

[6] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S. Yu. 2021. Continuous-Time Sequential Recommendation with Temporal Graph Collaborative Transformer. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. ACM.

[7] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[8] Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403* (2020).

[9] 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.

[10] 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.

[11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

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

[13] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. IEEE, 197–206.

[14] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362* (2020).

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *SIGKDD*. 1748–1757.

[17] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2615–2623.

[18] Haoyang Li, Xin Wang, Ziwei Zhang, Jianxin Ma, Peng Cui, and Wenwu Zhu. 2021. Intention-aware Sequential Recommendation with Structured Intent Transition. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[19] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *WSDM*. 322–330.

[20] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. 2020. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966* (2020).

[21] Yang Li, Tong Chen, Peng-Fei Zhang, and Hongzhi Yin. 2021. Lightweight Self-Attentive Sequential Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 967–977.

[22] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[23] 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).

[24] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S Yu. 2021. Augmenting Sequential Recommendation with Pseudo-Prior Items via Reversely Pre-training Transformer. *arXiv preprint arXiv:2105.00522* (2021).

[25] Zhiwei Liu, Xiaohan Li, Ziwei Fan, Stephen Guo, Kannan Achan, and S Yu Philip. 2020. Basket recommendation with multi-intent translation graph neural network. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 728–737.

[26] Zhiwei Liu, Lin Meng, Jiawei Zhang, and Philip S Yu. 2020. Deoscillated Graph Collaborative Filtering. *arXiv preprint arXiv:2011.02100* (2020).

[27] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 483–491.

[28] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*. 43–52.

[29] Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in neural information processing systems*. 2265–2273.

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[31] Zhiqiang Pan, Fei Cai, Yanxiang Ling, and Maarten de Rijke. 2020. An intent-guided collaborative machine for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1833–1836.

[32] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

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

[34] 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.

[35] 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 *CIKM*. 1441–1450.

[36] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. 565–573.

[37] Md Mehrab Tanjim, Congzhe Su, Ethan Benjamin, Diane Hu, Liangjie Hong, and Julian McAuley. 2020. Attentive sequential models of latent intent for next item recommendation. In *Proceedings of The Web Conference 2020*. 2528–2534.

[38] Wilson L Taylor. 1953. "Cloze procedure": A new tool for measuring readability. *Journalism quarterly* 30, 4 (1953), 415–433.

[39] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[41] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Longbing Cao. 2019. Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In *International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence.

[42] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[43] 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.

[44] 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.

[45] 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).

[46] An Yan, Shuo Cheng, Wang-Cheng Kang, Mengting Wan, and Julian McAuley. 2019. CosRec: 2D convolutional neural networks for sequential recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2173–2176.

[47] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2020. Self-supervised Learning for Large-scale Item Recommendations. *arXiv preprint arXiv:2007.12865* (2020).

[48] Jianwen Yin, Chenghao Liu, Weiqing Wang, Jianling Sun, and Steven CH Hoi. 2020. Learning transferrable parameters for long-tailed sequential user behavior modeling. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 359–367.

[49] Hengrui Zhang and Julian McAuley. 2020. Stacked Mixed-Order Graph Convolutional Networks for Collaborative Filtering. In *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 73–81.

[50] Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. *arXiv preprint arXiv:2009.12061* (2020).

[51] 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.

[52] Yao Zhou, Jianpeng Xu, Jun Wu, Zeinab Taghavi, Evren Korpeoglu, Kannan Achan, and Jingrui He. 2021. PURE: Positive-Unlabeled Recommendation with Generative Adversarial Network. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2409–2419.

A PSEUDO-CODE OF ICL FOR SR

Algorithm 1: ICL for SR

Input: training dataset $\{s_u\}_{u=1}^{|\mathcal{U}|}$, sequence encoder f_θ, batch size N , hyper-parameters K, λ, β.

Output: θ.

while *epoch* $\leq$ *MaxTrainEpoch* **do**

  `// E-step: Intent Representation Learning`

  $c = Clustering(\{f_\theta(S^u)\}_{u=1}^{|\mathcal{U}|}, K)$

  Update distribution function $Q(c_i) = P_\theta(c_i|S^u)$

  `// M-step: Multi-Task Learning`

  for *a minibatch* $\{s_u\}_{u=1}^{N}$ **do**

    for $u \in \{1, 2, \cdots, N\}$ **do**

      `// Construct 2 views.`

      $\tilde{S}_1^u = g_1^u(S^u), \tilde{S}_2^u = g_2^u(S^u)$, *where* $g_1^u, g_2^u \sim \mathcal{G}$

      `// Encoding via` $f_\theta(\cdot)$

      $\mathbf{h}^u = f_\theta(S^u)$

      $\tilde{\mathbf{h}}_1^u = f_\theta(\tilde{S}_1^u)$, $\tilde{\mathbf{h}}_2^u = f_\theta(\tilde{S}_2^u)$

    `// Optimization`

    $\mathcal{L} = \mathcal{L}_{\text{NextItem}} + \lambda \cdot \mathcal{L}_{\text{ICL}} + \beta \cdot \mathcal{L}_{\text{SeqCL}}$

    Update network $f_\theta(\cdot)$ to minimize $\mathcal{L}$

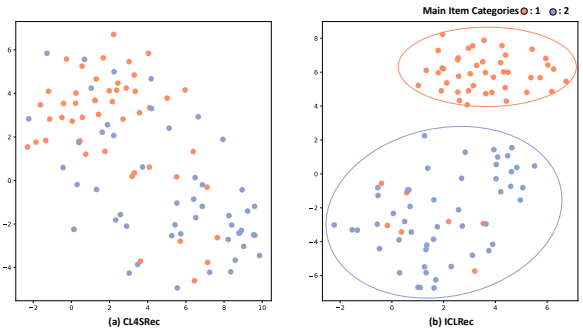

Figure 7: Visualization of the learned users' representations by CL4SRec and ICLRec on Sports.

B PROOF OF CONVERGENCE

To proof the convergence of ICL under the generalized EM framework, we just need to proof $P_{\theta^{(m+1)}}(S) \geq P_{\theta^{(m)}}(S)$, where m indicates the number of training iterations. Based on Eq. (7), we have

$$\ln P_{\theta^{(m)}}(S) = \ln \frac{P_{\theta^{(m)}}(S, c_i)}{P_{\theta^{(m)}}(c_i|S)} = \ln P_{\theta^{(m)}}(S, c_i) - \ln P_{\theta^{(m)}}(c_i|S). \tag{19}$$

Take the expectation in term of c condition over S on both sides, then we have:

$$\begin{aligned}\mathbb{E}_{(c|S,\theta^{(m)})}\left[\ln P_\theta(S)\right] = \mathbb{E}_{(c|S,\theta^{(m)})}\left[\ln P_{\theta^{(m)}}(S, c)\right] \\ - \mathbb{E}_{(c|S,\theta^{(m)})}\left[\ln P_{\theta_{(m)}}(c|S)\right].\end{aligned} \tag{20}$$

Based on Eq. (12), and 20, the term on left side equal to:

$$\mathbb{E}_{(c|S,\theta^{(m)})}\left[\ln P_{\theta^{(m)}}(S)\right] = \sum_{i=1}^{K} Q(c_i) \cdot \ln P_{\theta^{(m)}}(S) = \ln P_{\theta^{(m)}}(S). \tag{21}$$

Thus, proof $P_{\theta^{(m+1)}}(S) \geq P_{\theta^{(m)}}(S)$ is equivalent to proof

$$\ln P_{\theta^{(m+1)}}(S) \geq \ln P_{\theta^{(m)}}(S), \tag{22}$$

which is equivalent to:

$$\begin{aligned}\mathbb{E}_{(c|S,\theta^{(m+1)})}\left[\ln P_{\theta^{(m+1)}}(S, c)\right] - \mathbb{E}_{(c|S,\theta^{(m+1)})}\left[\ln P_{\theta_{(m+1)}}(c|S)\right] \\ \geq \mathbb{E}_{(c|S,\theta^{(m)})}\left[\ln P_{\theta^{(m)}}(S, c)\right] - \mathbb{E}_{(c|S,\theta^{(m)})}\left[\ln P_{\theta_{(m)}}(c|S)\right].\end{aligned} \tag{23}$$

Because we try to optimize θ at M-step, thus we have

$$\mathbb{E}_{(c|S,\theta^{(m+1)})}\left[\ln P_{\theta^{(m+1)}}(S, c)\right] \geq \mathbb{E}_{(c|S,\theta^{(m)})}\left[\ln P_{\theta^{(m)}}(S, c)\right]. \tag{24}$$

And based on Jsnson's inequality, we have

$$\mathbb{E}_{(c|S,\theta^{(m+1)})}\left[\ln P_{\theta_{(m+1)}}(c|S)\right] \leq \mathbb{E}_{(c|S,\theta^{(m)})}\left[\ln P_{\theta_{(m)}}(c|S)\right]. \tag{25}$$

Combining Eq. (23), (24),and (25), we show that $P_{\theta^{(m+1)}}(S) \geq P_{\theta^{(m)}}(S)$. Thus, the algorithm will converge.

C DATASET INFORMATION

Table 3: Dataset information.

Dataset	Sports	Beauty	Toys	Yelp
$\lvert\mathcal{U}\rvert$	35,598	22,363	19,412	30,431
$\lvert\mathcal{V}\rvert$	18,357	12,101	11,924	20,033
# Actions	0.3m	0.2m	0.17m	0.3m
Avg. length	8.3	8.9	8.6	8.3
Sparsity	99.95%	99.95%	99.93%	99.95%

D IMPACT OF BATCH SIZE AND THE STRENGTH OF SEQCL TASK β

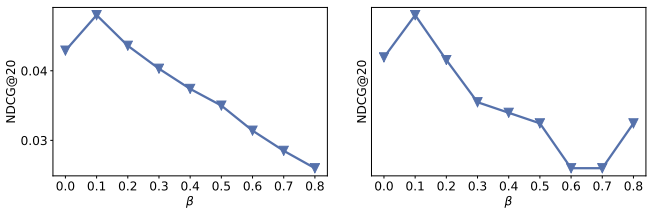

Figure 8: Impact of SeqCL task strength β on Beauty (left) and Yelp (right).

Performance w.r.t. batch size on Yelp between CL4SRec and the proposed ICLRec are shown in Figure. 6. We observe that with the batch size increases, CL4SRec's performance does not continually improve. The reason might because of larger batch sizes introduce false-negative samples, which harms learning. While ICLRec is relatively stable with different batch sizes, and out performs CL4SRec in all circumstances. Because the intent learnt can be seen as a pseudo label of sequences, which helps identify the true positive samples via the proposed contrastive SSL with FNM.

The impact of β is shown in Figure 8. We can see that, β does help ICLRec improve the performance when it is small (e.g., $\beta \leq 0.1$). However, when β continually increase, the model performance drop significantly. This phenomenon also indicates the limitation of SeqCL, since focusing on maximize mutual information between individual sequence pairs may break the global relationships among users.

Table 4: Quantitative Analysis Results. (NDCG@20)

Datasets	SASRec	CL4SRec	ICLRec-A	ICLRec
Sports	0.0216	0.0238	0.0272	0.0275 ($K = 2048$) **0.0287** ($K = 1024$)
Yelp	0.0179	0.0258	0.0264	0.0271 ($K = 1024$) **0.0283** ($K = 512$)

E CASE STUDY

Quantitative analysis. We study how ICLRec will perform by considering the item categories of users interacted items as their intents. Specifically, given a user behavior sequence S^u, we consider the mean of its corresponding trainable item category embeddings as the intent prototype $\mathbf{c}$, aiming to replace the intent representation learning described in Sec. 4.1.2. We run the corresponding model named ICLRec-A and show the comparison results in Table 4. We observe that on Sports (1) ICLRec-A performs better than CL4SRec, which shows the potential benefits of leveraging item category information. (2) ICLRec achieves similar performance as ICLRec-A's when $K = 2048$. Joint analysis with the above qualitative results indicates that ICL can capture meaningful user intents via SSL. (3) ICLRec can outperform ICLRec-A when $K = 1024$. We hypothesize that users' intents can be better described by the latent variables when $K = 1024$ thus improving performance. (e.g., parents of the existing item categories.) Similar observations in Yelp.

Qualitative analysis We also compare the proposed ICLRec with CL4SRec by visualizing the learned users' representations via t-SNE [39]. Specifically, we sampled 100 users for whom used to interact with one category of items or the other category. These 100 users also interacted with other categories of items in the past. We visualize the learned users' representations via t-SNE [39] in Figure 7. From Figure 7 we can see, users' representations learned by ICLRec intent to pull users that interacted with the same category of items closer to each other while pushing others further away in the representation space than CL4SRec. It reflects that representations learned by ICL can capture more semantic structures, therefore, improves the performance.