Item-Provider Co-learning for Sequential Recommendation

Lei Chen*
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
Shenzhen, China
lei.chen@siat.ac.cn

Jingtao Ding
WeChat Technical Architecture Department, Tencent Inc
Shenzhen, China
dingjt15@tsinghua.org

Min Yang†
Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
Shenzhen, China
min.yang@siat.ac.cn

Chengming Li†
School of Intelligent Systems Engineering, Sun Yat-sen University
Shenzhen, China
lichengming@mail.sysu.edu.cn

Chonggang Song
WeChat Technical Architecture Department, Tencent Inc
Shenzhen, China
jerrycgsong@tencent.com

Lingling Yi
WeChat Technical Architecture Department, Tencent Inc
Shenzhen, China
chrisyi@tencent.com

ABSTRACT

Sequential recommender systems (SRSs) have become a research hotspot recently due to its powerful ability in capturing users' dynamic preferences. The key idea behind SRSs is to model the sequential dependencies over the user-item interactions. However, we argue that users' preferences are not only determined by their view or purchase items but also affected by the item-providers with which users have interacted. For instance, in a short-video scenario, a user may click on a video because he/she is attracted to either the video content or simply the video-providers as the vloggers are his/her idols. Motivated by the above observations, in this paper, we propose IPSRec, a novel **I**tem-**P**rovider co-learning framework for **S**equential **Rec**ommendation. Specifically, we propose two representation learning methods (single-steam and cross-stream) to learn comprehensive item and user representations based on the user's historical item sequence and provider sequence. Then, contrastive learning is employed to further enhance the user embeddings in a self-supervised manner, which treats the representations of a specific user learned from the item side as well as the item-provider side as the positive pair and treats the representations of different users in the batch as the negative samples. Extensive experiments on three real-world SRS datasets demonstrate that IPSRec achieves substantially better results than the strong competitors. For reproducibility, our code and data are available at https://github.com/siat-nlp/IPSRec.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Personalization**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Sequential recommendation, Co-learning, Co-attention fusion, Contrastive learning

ACM Reference Format:
Lei Chen, Jingtao Ding, Min Yang, Chengming Li, Chonggang Song, and Lingling Yi. 2022. Item-Provider Co-learning for Sequential Recommendation. In *Proceedings of the 45th Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531756

1 INTRODUCTION

Sequential recommendation (SR) models each user as a sequence of items interacted in the past and aims at predicting the next item for the user. Recent advances on SR are overwhelmingly contributed by deep learning techniques [1, 3, 4, 10, 12, 18], which have taken the state-of-the-art of SR to a new level. In general, these models could be classified into four categories, namely recurrent neural network (RNN) based models [3], convolutional neural network (CNN) based models [10, 16], self-attention based models [4, 6, 9, 18], and graph neural network (GNN) based methods [12, 13, 15]. In this paper, we instantiate the self-attention models as our backbone network given its easiness for implementation and superior performance that has been well evaluated in the literature [6, 9, 18].

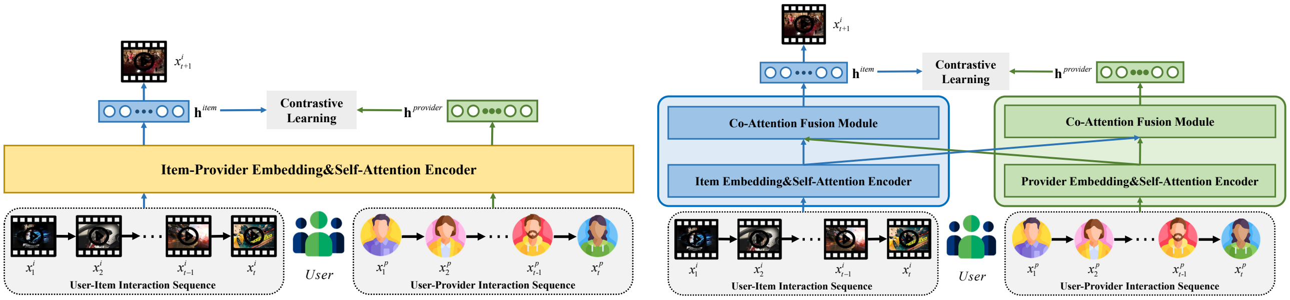
More recently, contrastive learning techniques, which aims to learn effective representations by pulling semantically close neighbors together and pushing apart non-neighbors, have shown impressive performance in SR tasks [7, 14]. The key idea behind the contrastive learning is to randomly perturb the input data twice as the positive pair and sample unobserved items for each input sample as negative samples. Then, the users' and items' representations can be learned in a self-supervised manner by maximizing the similarity between the representations of a positive pair while minimizing the similarity between that of negative pairs.

Despite the effectiveness of previous studies, there are still several challenges for learning high-quality item and user representations. First, existing methods were focused on modeling the sequential dependencies given the user-item interactions and overlooked the influence from item-providers. However, based on our observations, the users' preferences are not only determined by their view or purchase items but also affected by the item-providers. For

[*] This work was conducted when Lei Chen was a research intern at WeChat Technical Architecture Department, Tencent Inc as a member of 2021 Tencent Rhino-Bird Research Elite Training Program.
[†] Corresponding authors.

SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07.
https://doi.org/10.1145/3477495.3531756


x^i_{t+1}
$\mathbf{h}^{item}$
Contrastive Learning
$\mathbf{h}^{provider}$
Item-Provider Embedding&Self-Attention Encoder
x^i_1
x^i_2
x^i_{t-1}
x^i_t
User-Item Interaction Sequence
User
x^p_1
x^p_2
x^p_{t-1}
x^p_t
User-Provider Interaction Sequence
x^i_{t+1}
$\mathbf{h}^{item}$
Contrastive Learning
$\mathbf{h}^{provider}$
Co-Attention Fusion Module
Item Embedding&Self-Attention Encoder
Co-Attention Fusion Module
Provider Embedding&Self-Attention Encoder
x^i_1
x^i_2
x^i_{t-1}
x^i_t
User-Item Interaction Sequence
User
x^p_1
x^p_2
x^p_{t-1}
x^p_t
User-Provider Interaction Sequence

(a) Single-stream

(b) Cross-stream

Figure 1: The architecture of IPSRec framework. IPSRec consists of a single-stream version IPSRec-S (on the left) and a cross-stream version IPSRec-C (on the right).

instance, in a short-video scenario, a user may click on a video because he/she is attracted to either the video content or simply the video-providers as the vloggers are his/her idols; in an E-commerce scenario, a user may purchase a dress because it suits her taste or she trusts the seller. It is necessary to take into account both user-item interactions and user-item-provider interactions. Second, most previous contrastive learning methods for SR tasks randomly select unobserved items as negative samples. However, when a user interacts with a huge amount of items, the irrelevant items may wash out the user's real interest, leading to inferior sequential recommendation.

To deal with the aforementioned challenges, we propose IPSRec, a novel **I**tem-**P**rovider co-learning framework for **S**equential **Rec**ommendation with contrastive learning. First, we propose two representation learning methods (single-steam and cross-stream) to learn comprehensive item and user representations based on the user's historical item sequence and provider sequence. Specifically, in single-stream IPSRec, we concatenate the item sequence and provider sequence together, and then leverage the self-attention operation on the concatenated sequence to obtain the integrated item, provider and user representations. In cross-stream IPSRec, we first learn the item and provider representations separately via self-attention, and then deploy a co-attention fusion module to capture interactions between item and provider representations. Second, we utilize the contrastive learning framework to further refine the user representations in a self-supervised manner from both the item side and the provider side, which treats the representations of a specific user learned from the item side as well as the item-provider side as the positive pair and treats the representations of different users in the batch as the negative pairs.

We summarize our main contributions as follows. (1) To the best of our knowledge, we are the first to emphasize the importance of item-providers for the SR tasks. We propose two attention-based methods (single-stream and cross-stream) to learn high-quality item and user representations given the user's historical item sequence and provider sequence. (2) We propose a contrastive learning method to further refine the user representations in a self-supervised manner, which learns the user representations from both the item sequence and the item-provider sequence. (3) Extensive experiments on three real-world SR datasets demonstrate that IPSRec outperforms the strong baselines by a substantial margin.

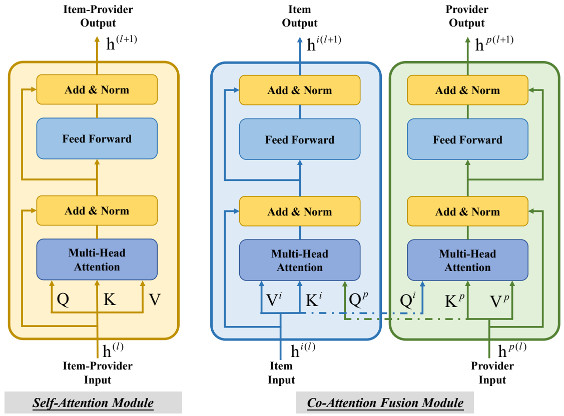


Figure 2: Self-attention and co-attention fusion modules.

2 OUR METHODOLOGY

Problem Definition. The goal of SR is to predict the next item with which a user will likely interact in a near future. Formally, given the users' historical item sequence $X^i = [x^i_1, \dots, x^i_t]$ (also denoted by $x^i_{1:t}$) and the corresponding item-provider sequence $X^p = [x^p_1, \dots, x^p_t]$ (also denoted by $x^p_{1:t}$), where x^i_t and x^p_t denote the t-th interacted item and item-provider respectively, our goal is to predict the next item x^i_{t+1} that the user would like to interact with at time step $t + 1$.

The Overall Architecture. In this paper, we propose IPSRec, a novel item-provider co-learning framework for enhancing sequential recommendation, which consists of two alternative versions (i.e., a single-stream version IPSRec-S and a cross-stream version IPSRec-C). Figure 1 illustrates the overview of the IPSRec framework. Specifically, in IPSRec-S, we first concatenate the item sequence and provider sequence together, and then perform the self-attention operation on the whole sequence to obtain the fused item and user representations. While in IPSRec-C, we first model the item sequence and provider sequence separately with self-attention, and then deploy a co-attention fusion module to model the interactions between item and provider representations. In addition, we propose a contrastive learning method to further refine the user representations in a self-supervised manner. Next, we will introduce IPSRec-S, IPSRec-C, and contrastive learning in detail.

2.1 IPSRec-S

In IPSRec-S, we first concatenate the item sequence and provider sequence together, and then perform self-attention operation on the whole sequence to obtain the fused representations. Similar to SASRec [4], the main architecture of IPSRec-S is a Transformer [11], which consists of an embedding layer and a multi-head self-attention module.

For each item x^i in users' historical item sequence X^i, we first convert it into an embedding vector $\mathbf{e}^i$. The item sequence X^i is thereby represented by an item embedding matrix $\mathbf{E}^i = [\mathbf{e}^i_1, \dots, \mathbf{e}^i_t]$. Similarly, we can obtain the provider embedding matrix $\mathbf{E}^p = [\mathbf{e}^p_1, \dots, \mathbf{e}^p_t]$ for the whole provider sequence X^p.

To preserve the chronological order and segment information of the item and provider sequences, we constructed an item positional embedding matrix $\mathbf{P}^i = [\mathbf{p}^i_1, \dots, \mathbf{p}^i_t]$, a provider positional embedding matrix $\mathbf{P}^p = [\mathbf{p}^p_1, \dots, \mathbf{p}^p_t]$, and a segment embedding matrix S consisting of an item segment embedding vector $\mathbf{s}^i$ and a provider segment embedding vector $\mathbf{s}^p$. Formally, we add up the item embedding, item positional embedding and item segment embedding as the initial input vector for each item in the item sequence at time step t:

$$\mathbf{h}_t^{S_{i(0)}} = \mathbf{e}^i_t + \mathbf{p}^i_t + \mathbf{s}^i \tag{1}$$

Meanwhile, we can get the initial input vector $\mathbf{h}_t^{S_{p(0)}}$ for each provider in the provider sequence at time step t similarly.

In IPSRec-S, the item sequence embedding and the provider sequence embedding are concatenated together to form the combined initial input of the Transformer:

$$\mathbf{H}^{S_{(0)}} = [\mathbf{h}_1^{S_{i(0)}}, \dots, \mathbf{h}_t^{S_{i(0)}}, \mathbf{h}_1^{S_{p(0)}}, \dots, \mathbf{h}_t^{S_{p(0)}}] \tag{2}$$

Given the combined input $\mathbf{H}^{S_{(0)}}$, we employ an l-layer Transformer to compute the representations of each item and provider:

$$\mathbf{H}^S = \mathrm{Transformer}(\mathbf{H}^{S_{(0)}}) \tag{3}$$

where each Transformer layer contains two sub-layers (i.e., a multi-head self-attention layer and a fully connected feed-forward layer), and the residual connection and layer normalization are applied to each of the two continuous sub-layers. In particular, the multi-head self-attention (SA) layer is the core module of the Transformer architecture, which is formulated as:

$$\mathrm{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{\mathrm{d/h}}}\right)\mathbf{V} \tag{4}$$

where queries ($\mathbf{Q}$), keys ($\mathbf{K}$) and values ($\mathbf{V}$) are all derived by a linear projection from the combined input $\mathbf{H}^{S_{(0)}}$. Noting that $\sqrt{d/h}$ is the scaling factor to avoid large values of the inner product [11].

Then, we regard the item representation $\mathbf{h}_t^{S_i}$ in $\mathbf{H}^\mathrm{S} = [\mathbf{h}_1^{S_i}, \dots, \mathbf{h}_t^{S_i}, \mathbf{h}_1^{S_p}, \dots, \mathbf{h}_t^{S_p}]$ as the final user representation to predict the next item, which can be optimized by the standard cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$:

$$p(\hat{x}^i_{t+1}) = \mathrm{softmax}(\mathbf{h}_t^{S_i}\mathbf{W}_1) \tag{5}$$

$$\mathcal{L}_{\mathrm{CE}} = -p(x^i_{t+1}) \log p(\hat{x}^i_{t+1}) \tag{6}$$

where $p(x^i_{t+1})$ and $p(\hat{x}^i_{t+1})$ represent the ground truth distribution and the prediction distribution for the next item respectively. $\mathbf{W}_1$ is a learnable projection parameter.

Table 1: Dataset statistcs (after preprocessing). Noting that "I-Actions" and "P-Actions" are short for number of user-item interactions and user-provider interactions, respectively.

Dataset	#Users	#Items	#I-Actions	#Providers	#P-Actions
Tmall	885,760	1,144,125	7,592,214	9,998	7,592,214
Twitch	100,001	739,992	3,051,733	162,626	3,051,733
WeChat	20,001	96,565	7,317,882	18,430	7,317,882

2.2 IPSRec-C

Different from IPSRec-S, in IPSRec-C, we first model the item sequence and provider sequence separately with self-attention, and then deploy a novel co-attention fusion module to learn the interactions between item and provider representations.

Embedding Layer. Similar to IPSRec-S, we add up the item embedding, item positional embedding and item segment embedding to obtain the initial item sequence embedding $\mathbf{H}^{C_{i(0)}} = [\mathbf{h}_1^{C_{i(0)}}, \dots, \mathbf{h}_t^{C_{i(0)}}]$. We can also get the initial provider sequence embedding $\mathbf{H}^{C_{p(0)}} = [\mathbf{h}_1^{C_{p(0)}}, \dots, \mathbf{h}_t^{C_{p(0)}}]$ in the same way. Then, we employ two separate Transformers to encode the item sequence as well as provider sequence respectively and learn the interactions between them through a co-attention fusion module.

Co-attention Fusion Module. After getting the separate representations of the item sequence and the provider sequence from the corresponding Transformer encoders (i.e., $\mathrm{Transformer}^\mathrm{i}$ and $\mathrm{Transformer}^\mathrm{p}$), a co-attention fusion module is devised to learn the fused representations by emphasizing the interactions between representations of the item side and the provider side, as illustrated in Figure 2. Formally, the co-attention fusion function is defined as:

$$\mathrm{CA}^\mathrm{i}(\mathbf{Q}^\mathrm{p}, \mathbf{K}^\mathrm{i}, \mathbf{V}^\mathrm{i}) = \mathrm{softmax}\left(\frac{\mathbf{Q}^\mathrm{p}(\mathbf{K}^\mathrm{i})^\top}{\sqrt{\mathrm{d/h}}}\right)\mathbf{V}^\mathrm{i} \tag{7}$$

$$\mathrm{CA}^\mathrm{p}(\mathbf{Q}^\mathrm{i}, \mathbf{K}^\mathrm{p}, \mathbf{V}^\mathrm{p}) = \mathrm{softmax}\left(\frac{\mathbf{Q}^\mathrm{i}(\mathbf{K}^\mathrm{p})^\top}{\sqrt{\mathrm{d/h}}}\right)\mathbf{V}^\mathrm{p} \tag{8}$$

where queries ($\mathbf{Q}^i$), keys ($\mathbf{K}^i$), values ($\mathbf{V}^i$) are from the item side and queries ($\mathbf{Q}^p$), keys ($\mathbf{K}^p$), values ($\mathbf{V}^p$) are from the provider side, which are all derived by a linear projection with the corresponding representations (i.e., $\mathrm{Transformer}^\mathrm{i}(\mathbf{H}^{C_{i(0)}})$ and $\mathrm{Transformer}^\mathrm{p}(\mathbf{H}^{C_{p(0)}})$), formulated as:

$$\mathbf{Q}^i = \mathbf{W}^{\mathrm{Q}_i}\mathrm{Transformer}^\mathrm{i}(\mathbf{H}^{C_{i(0)}}),\ \mathbf{Q}^p = \mathbf{W}^{\mathrm{Q}_p}\mathrm{Transformer}^\mathrm{p}(\mathbf{H}^{C_{p(0)}}) \tag{9}$$

$$\mathbf{K}^i = \mathbf{W}^{\mathrm{K}_i}\mathrm{Transformer}^\mathrm{i}(\mathbf{H}^{C_{i(0)}}),\ \mathbf{K}^p = \mathbf{W}^{\mathrm{K}_p}\mathrm{Transformer}^\mathrm{p}(\mathbf{H}^{C_{p(0)}}) \tag{10}$$

$$\mathbf{V}^i = \mathbf{W}^{\mathrm{V}_i}\mathrm{Transformer}^\mathrm{i}(\mathbf{H}^{C_{i(0)}}),\ \mathbf{V}^p = \mathbf{W}^{\mathrm{V}_p}\mathrm{Transformer}^\mathrm{p}(\mathbf{H}^{C_{p(0)}}) \tag{11}$$

where $\mathbf{W}^{\mathrm{Q}_i}$, $\mathbf{W}^{\mathrm{Q}_p}$, $\mathbf{W}^{\mathrm{K}_i}$, $\mathbf{W}^{\mathrm{K}_p}$, $\mathbf{W}^{\mathrm{V}_i}$ and $\mathbf{W}^{\mathrm{V}_p}$ are learnable projection parameters. With this co-attention fusion function, we can achieve full interactions between item representations and provider representations.

After learning the co-attention fusion between item and provider representations, we can derive the final fused item representations $\mathbf{H}^{C_i} = [\mathbf{h}_1^{C_i}, \ldots, \mathbf{h}_t^{C_i}]$ and provider representations $\mathbf{H}^{C_p} = [\mathbf{h}_1^{C_p}, \ldots, \mathbf{h}_t^{C_p}]$, respectively. And we regard the last item representation $\mathbf{h}_t^{C_i}$ in $\mathbf{H}^{C_i}$ as the final user representation to make subsequent predictions and utilized the standard cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$ to optimize the whole model, similar to IPSRec-S.

2.3 Contrastive Learning Framework

In this paper, we model item sequence and provider sequence simultaneously for better capturing the user preferences, and we argue that it should be consistent on the item side and the provider side. In this paper, we utilize the contrastive learning framework in [2] to further refine the user representations in a self-supervised manner from both the item side and the provider side, which treats the representations of a specific user learned from the item side and the item-provider side as the positive pair, and treats the representations of different users in the batch as the negative pairs. In particular, we adopt the Noise Contrastive Estimation (NCE) loss $\mathcal{L}_{\mathrm{CL}}$ [2] to optimize the item representations and provider representations:

$$\mathcal{L}_{\mathrm{CL}} = -\log \frac{\exp\left(\rho\left(\mathbf{h}_t^i, \mathbf{h}_t^p\right)/\tau\right)}{\exp\left(\rho\left(\mathbf{h}_t^i, \mathbf{h}_t^p\right)/\tau\right) + \sum\limits_{\mathbf{h}_t^{p-}\in S^-} \exp\left(\rho\left(\mathbf{h}_t^i, \mathbf{h}_t^{p-}\right)/\tau\right)} \tag{12}$$

where $\mathbf{h}_t^i$ (represents $\mathbf{h}_t^{S_i}$ or $\mathbf{h}_t^{C_i}$) and $\mathbf{h}_t^p$ (represents $\mathbf{h}_t^{S_p}$ or $\mathbf{h}_t^{C_p}$) represent the last item representation and provider representation respectively. That is, $\mathbf{h}_t^i$ and $\mathbf{h}_t^p$ also represent the user representations from the item side and the provider side respectively. ρ represents dot product to measure the similarity between the representations, following [7, 14]. τ is a temperature parameter (set to 1 by default) to control the discreteness of the output. $\mathbf{h}_t^{p-}$ denotes the negative provider representations in the set S^- of in-batch negative samples.

Finally, we jointly optimize the weighted-sum of the standard cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$ and the contrastive learning loss $\mathcal{L}_{\mathrm{CL}}$ in a multi-task way. The combined overall loss $\mathcal{L}$ is defined by:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda \mathcal{L}_{\mathrm{CL}} \tag{13}$$

where λ is a hyperparameter to control the weight of the contrastive learning loss, which is set to 0.1 in our experiments.

3 EXPERIMENTAL SETUP

3.1 Experimental Datasets

We conduct extensive experiments on three real-world datasets: Tmall[1], Twitch [8] and WeChat[2]. Each dataset contains users’ historical behaviors in chronological order, including users’ involved items and the item-providers that users have interacted with. The statistics of the three datasets are reported in Table 1.

3.2 Baselines and Evaluation Metrics

To verify the effectiveness of IPSRec, we compare it with several strong representative baselines, including GRU4Rec [3], Caser [10], SASRec [4] and SR-GNN [13]. We adopt three popular top-N ranking evaluation metrics to measure the recommendation performance, including HR@N (Hit Ratio), MRR@N (Mean Reciprocal Rank) and NDCG@N (Normalized Discounted Cumulative Gain) [17]. Here, N is set to 5 and 10 for comparison.

3.3 Implementation Details

For each user, we divide the last interacted item and provider as the test data, and hold out the penultimate interacted item and provider as the validation data, following [4, 14]. The remaining interacted items and providers are used for training. We set the maximum item sequence length and provider sequence length to 20, 50 and 100 for Tmall, Twitch and WeChat respectively. The embedding size of item or provider is set to 64. In addition, the numbers of layers and heads in the Transformer are set to 2. For all the baselines, we utilize the public implementations in [17] for reproducing the experimental results. We employ Adam [5] to optimize the parameters of IPSRec with learning rate 0.001, where the batch size is set to 256.

4 EXPERIMENTAL RESULTS

4.1 Quantitative Results

Table 2 reports the overall performance of IPSRec (including IPSRec-S and IPSRec-C) and baseline models on the three datasets. We can observe that IPSRec consistently and substantially surpasses the compared models by a noticeable margin on all the three datasets. For example, on Tmall, Twitch and WeChat, IPSRec (better result in IPSRec-S and IPSRec-C) obtains 7.6%, 5.8% and 9.6% improvements in terms of HR@5 over the best baseline, which indicates the superiority of our item-provider co-learning when compared to just modelling the item side.

4.2 Ablation Study

In order to evaluate the impacts of different components to the superiority of IPSRec, we conduct ablation study in terms of discarding item-provider co-learning and contrastive learning , as reported in Table 3. Noting that our proposed IPSRec will degenerate to the similar architecture of SASRec when removing the item-provider co-learning module, resulting in the sharp decreases of performance. In addition, when removing the contrastive learning loss $\mathcal{L}_{\mathrm{CL}}$ (i.e., w/o CL), the performance on all of the three datasets decreases significantly, demonstrating the importance and necessity to keep it consistent on the item side and the provider side regarding as the user preferences.

5 CONCLUSION

In this paper, we proposed IPSRec, a novel item-provider co-learning framework for sequential recommendation. We were the first to emphasize the importance of item-providers in capturing user preferences in SR tasks. In addition, we utilized contrastive learning framework to learn comprehensive user representations from both the item side and the provider side. Extensive experiments on three

[1] https://tianchi.aliyun.com/dataset/dataDetail?dataId=53
[2] https://algo.weixin.qq.com/

Table 2: Overall results. Note that the improvements of IPSRec over all baseline models are statistically significant in terms of paired t-test with p-value < 0.01.

Dataset	Metric	GRU4Rec	Caser	SASRec	SR-GNN	IPSRec-S	Improv.	IPSRec-C	Improv.
Tmall	HR@5↑	0.0638	0.0422	0.0706	0.0620	0.0743	5.2%	**0.0760**	7.6%
	HR@10↑	0.0767	0.0544	0.0880	0.0760	**0.0964**	9.5%	0.0956	8.6%
	MRR@5↑	0.0484	0.0305	0.0489	0.0450	0.0518	5.9%	**0.0530**	8.4%
	MRR@10↑	0.0501	0.0321	0.0512	0.0468	0.0552	7.8%	**0.0556**	8.6%
	NDCG@5↑	0.0523	0.0334	0.0543	0.0492	0.0568	4.6%	**0.0587**	8.1%
	NDCG@10↑	0.0564	0.0373	0.0599	0.0537	**0.0652**	8.8%	0.0650	8.5%
Twitch	HR@5↑	0.0907	0.0711	0.0972	0.0823	0.1014	4.3%	**0.1028**	5.8%
	HR@10↑	0.1348	0.1074	0.1487	0.1220	0.1549	4.2%	**0.1560**	4.9%
	MRR@5↑	0.0485	0.0374	0.0482	0.0440	0.0493	2.3%	**0.0505**	4.8%
	MRR@10↑	0.0544	0.0422	0.0550	0.0493	0.0564	2.5%	**0.0575**	4.5%
	NDCG@5↑	0.0589	0.0457	0.0603	0.0535	0.0622	3.2%	**0.0634**	5.1%
	NDCG@10↑	0.0732	0.0574	0.0769	0.0663	0.0795	3.4%	**0.0805**	4.7%
WeChat	HR@5↑	0.0452	0.0402	0.0469	0.0426	**0.0514**	9.6%	0.0510	8.7%
	HR@10↑	0.0733	0.0635	0.0740	0.0696	**0.0824**	11.4%	0.0801	8.2%
	MRR@5↑	0.0225	0.0214	0.0235	0.0221	0.0255	8.5%	**0.0263**	11.9%
	MRR@10↑	0.0254	0.0244	0.0271	0.0256	0.0296	9.2%	**0.0301**	11.1%
	NDCG@5↑	0.0281	0.0260	0.0293	0.0271	0.0319	8.9%	**0.0324**	10.6%
	NDCG@10↑	0.0365	0.0335	0.0380	0.0358	**0.0418**	10.0%	0.0417	9.7%

Table 3: Ablation study results.

Method	Tmall			Twitch			WeChat		
	HR@5↑	MRR@5↑	NDCG@5↑	HR@5↑	MRR@5↑	NDCG@5↑	HR@5↑	MRR@5↑	NDCG@5↑
SASRec	0.0706	0.0489	0.0543	0.0972	0.0482	0.0603	0.0469	0.0235	0.0293
IPSRec-S	0.0743	0.0518	0.0568	0.1014	0.0493	0.0622	0.0514	0.0255	0.0319
w/o CL	0.0712	0.0498	0.0546	0.0977	0.0481	0.0603	0.0467	0.0230	0.0296
IPSRec-C	0.0760	0.0530	0.0587	0.1028	0.0505	0.0634	0.0510	0.0263	0.0324
w/o CL	0.0717	0.0495	0.0551	0.0980	0.0487	0.0611	0.0473	0.0247	0.0302

real-world datasets demonstrated that IPSRec achieves substantially better performance than the strong competitors.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China (No. 61906185), Natural Science Foundation of Guangdong Province of China (No. 2021A1515011905), Youth Innovation Promotion Association of CAS China (No. 2020357), Shenzhen Science and Technology Innovation Program (No. KQTD20190 929172835662), Shenzhen Basic Research Foundation (No. JCYJ202 10324115614039 and No. JCYJ20200109113441941). Lei Chen was supported by 2021 Tencent Rhino-Bird Research Elite Training Program.

REFERENCES

[1] Lei Chen, Fajie Yuan, Jiaxi Yang, Xiang Ao, Chengming Li, and Min Yang. 2021. A User-Adaptive Layer Selection Framework for Very Deep Sequential Recommender Models. In *AAAI*, Vol. 35. 3984–3991.

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

[3] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. *ICLR* (2016).

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

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

[6] Chang Liu, Xiaoguang Li, Guohao Cai, Zhenhua Dong, Hong Zhu, and Lifeng Shang. 2021. Non-invasive Self-attention for Side Information Fusion in Sequential Recommendation. *arXiv preprint arXiv:2103.03578* (2021).

[7] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2021. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. *arXiv preprint arXiv:2110.05730* (2021).

[8] Jérémie Rappaz, Julian McAuley, and Karl Aberer. 2021. Recommendation on Live-Streaming Platforms: Dynamic Availability and Repeat Consumption. In *RecSys*. 390–399.

[9] 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.

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

[11] 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 *NeurIPS*. 5998–6008.

[12] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.

[13] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *AAAI*. 346–353.

[14] 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).

[15] 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*. 3940–3946.

[16] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *WSDM*. 582–590.

[17] 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 *CIKM*. 4653–4664.

[18] 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 *CIKM*. 1893–1902.