AdaptSSR: Pre-training User Model with Augmentation-Adaptive Self-Supervised Ranking

Yang Yu[1,2], Qi Liu[1,2*], Kai Zhang[1,2], Yuren Zhang[1,2], Chao Song[3], Min Hou[4]
Yuqing Yuan[3], Zhihao Ye[3], Zaixi Zhang[1,2], Sanshi Lei Yu[1,2]

[1]Anhui Province Key Laboratory of Big Data Analysis and Application,
University of Science and Technology of China

[2]State Key Laboratory of Cognitive Intelligence

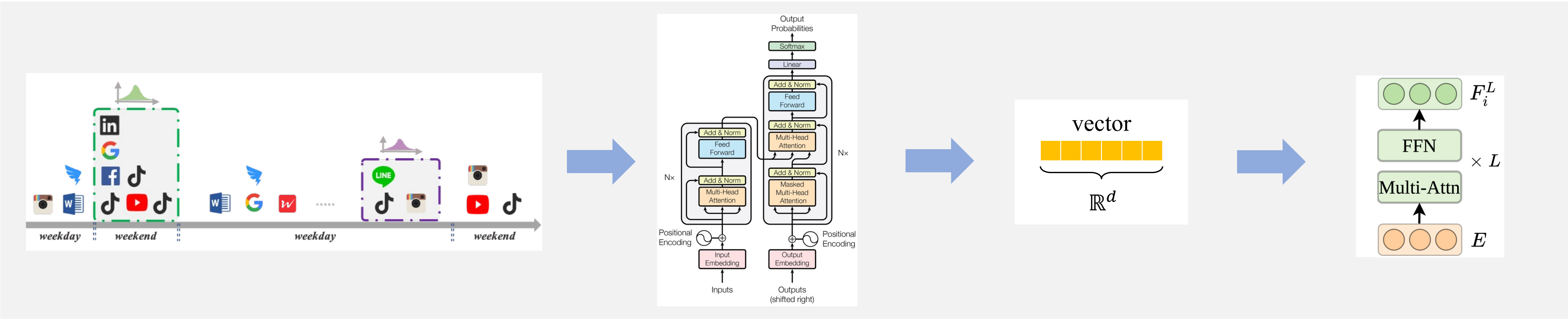
[3]OPPO Research Institute [4]Hefei University of Technology

Outline

- Introduction
- Methodology
- Experiments
- Conclusion

Introduction

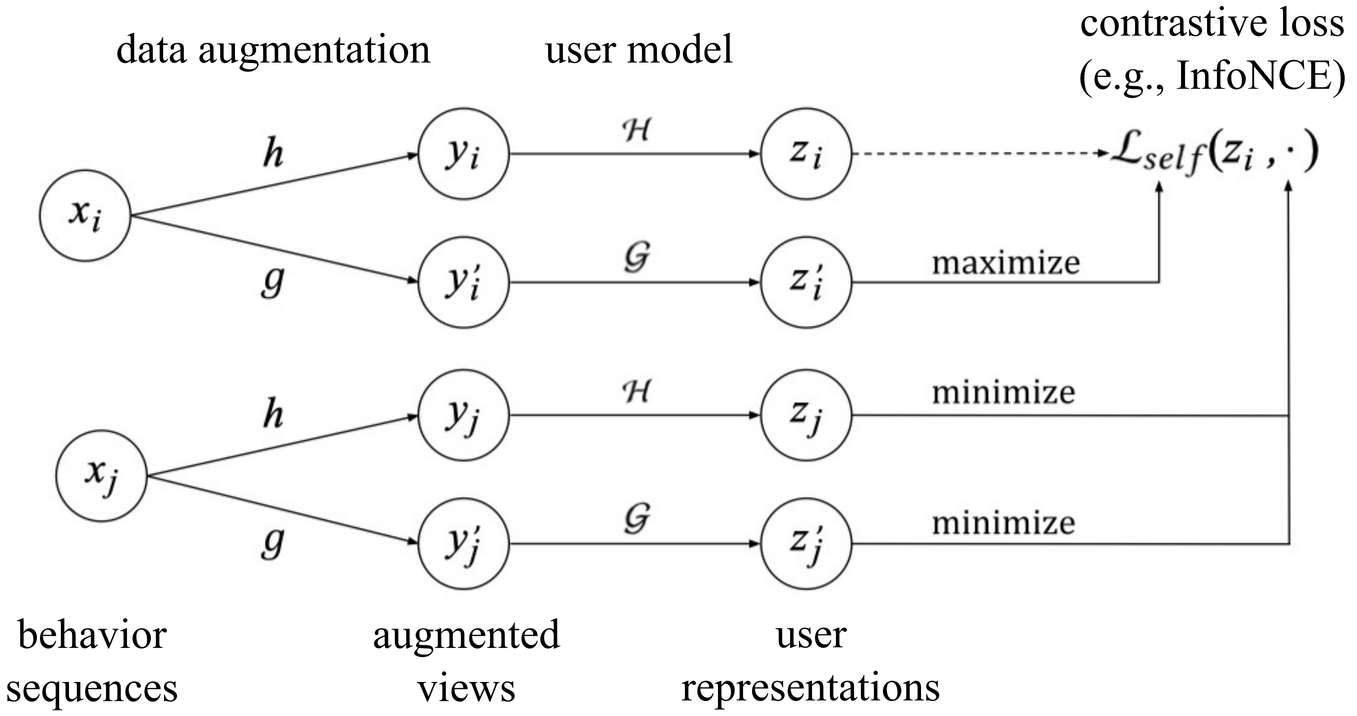
- **User modeling** aims to capture the user's characteristics or interests for a specific user-oriented task, such as user profiling and personalized recommendation.
- Existing supervised methods heavily rely on **task-specific labeled data** and suffer from the **data sparsity problem**.
- A mainstream technique to tackle this challenge is the **pre-training paradigm**.
  - The user model is first pre-trained on a mass of unlabeled user behavior data.
  - Then the model is transferred to benefit various downstream tasks via fine-tuning.



User behavior sequence　　Pre-trained user model　　Pre-trained user representation　　Downstream task model

Introduction

- Inspired by the recent progress in CV and NLP, several recent works explored pre-training the user model with a **contrastive learning** task.
- They assume different views of the same behavior sequence constructed via data augmentation are **semantically consistent**, i.e., reflecting similar characteristics or interests of the same user, and thus maximizing their agreement in the feature space.



Introduction

- Due to the **diverse interests** and **heavy noise** in user behaviors, existing data augmentation methods tend to lose certain characteristics of the user or introduce noisy behaviors.
- To address this problem, we propose to replace the contrastive learning task with a new pretext task: Augmentation-**Adap**tive **S**elf-**S**upervised **R**anking (**AdaptSSR**).

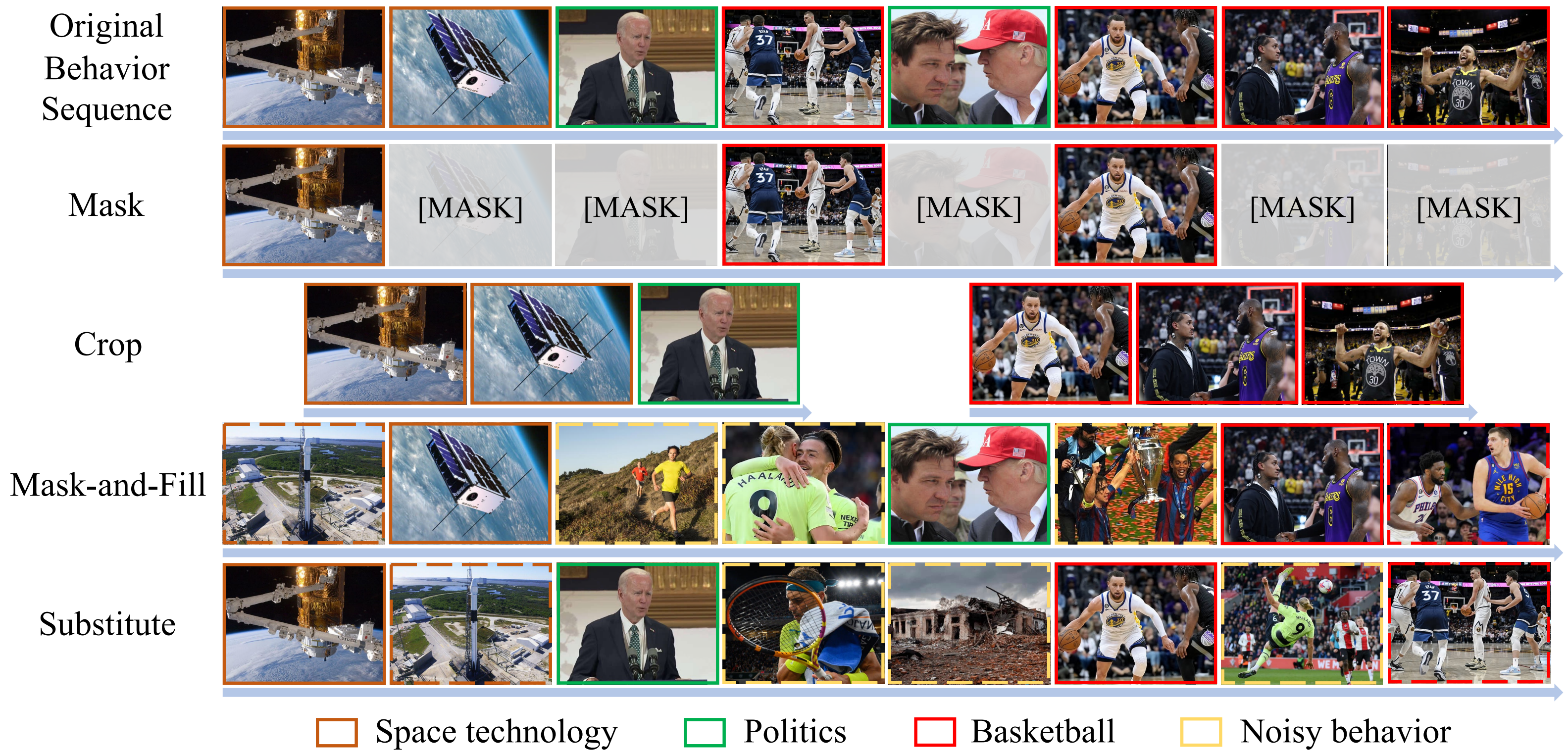


Figure 1: An illustration of the impact of different data augmentation methods on the user behavior sequence.

Methodology

- **Main Idea: Self-Supervised Ranking**
  - Train the user model $\mathcal{M}$ to capture the similarity order between the implicitly augmented view, the explicitly augmented view, and views from other users.
  - Given a user behavior sequence $S = \{x_1, x_2, \dots, x_n\}$
    - Input S into $\mathcal{M}$ twice with different independently sampled dropout masks $\rightarrow \boldsymbol{u}, \boldsymbol{u}^+$ (implicit data augmentation)
    - Input the augmented behavior sequence $\hat{S}$ into $\mathcal{M} \rightarrow \hat{\boldsymbol{u}}$ (explicit data augmentation)
    - Input the behavior sequence of another user into $\mathcal{M} \rightarrow \boldsymbol{u}^-$
  - **Pre-training objective**: $\mathrm{sim}(\boldsymbol{u}, \boldsymbol{u}^+) \geq \mathrm{sim}(\boldsymbol{u}, \hat{\boldsymbol{u}}) \geq \mathrm{sim}(\boldsymbol{u}, \boldsymbol{u}^-)$

Methodology

- **Multiple Pairwise Ranking (MPR) with In-batch Hard Negative Sampling**
  - Given a batch of user behavior sequences $\{S_i\}_{i=1}^{B}$, apply two randomly selected explicit augmentation methods to each sequence $S_i \rightarrow \hat{S}_i$ and $\tilde{S}_i$
  - Input $\hat{S}_i$ and $\tilde{S}_i$ into $\mathcal{M}$ twice $\rightarrow \hat{\boldsymbol{u}}_i, \hat{\boldsymbol{u}}_i^+$ and $\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_i^+$

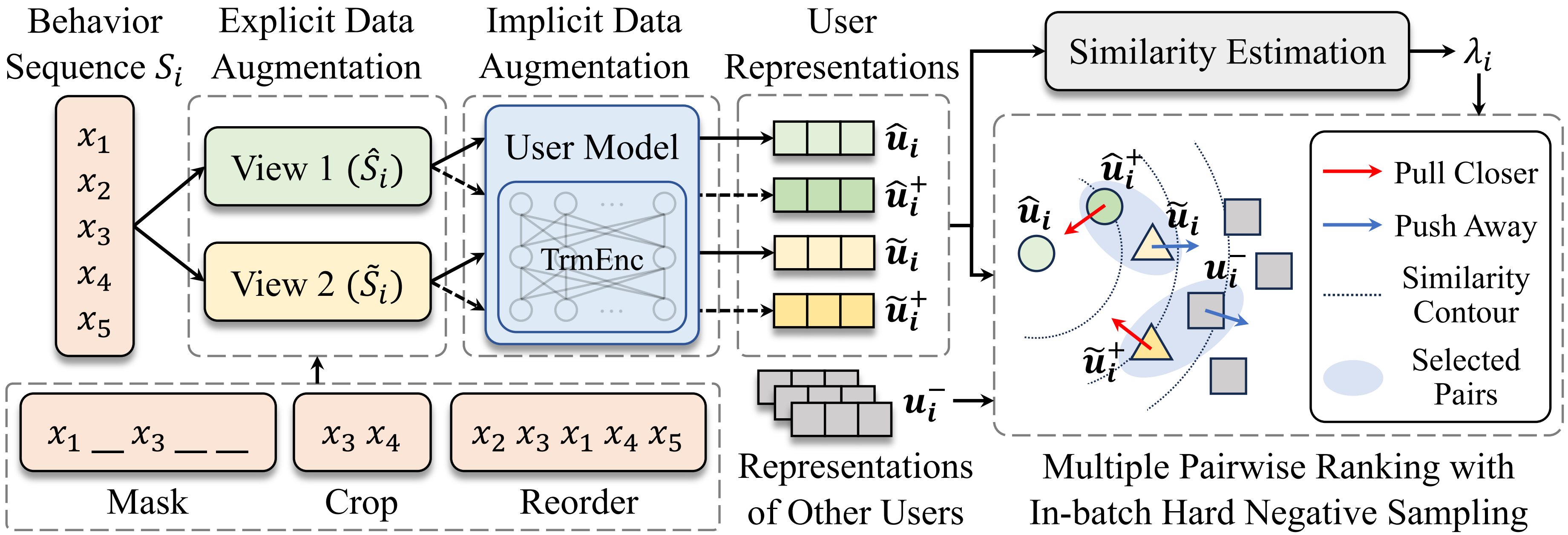


Figure 2: The framework of AdaptSSR. A sequence with five user behaviors is used for illustration.

Methodology

- **Multiple Pairwise Ranking (MPR) with In-batch Hard Negative Sampling**
  - **MPR loss**: extend the BPR loss to learn two pairwise ranking orders simultaneously.
  - For the augmented sequence $\hat{S}_i$, the user representation $\hat{\boldsymbol{u}}_i, \hat{\boldsymbol{u}}_i^+$ and each $\boldsymbol{v} \in \{\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_i^+\}$, $\boldsymbol{w} \in \mathbf{U}_i^- = \{\hat{\boldsymbol{u}}_j, \hat{\boldsymbol{u}}_j^+, \tilde{\boldsymbol{u}}_j, \tilde{\boldsymbol{u}}_j^+\}_{j=1, j\neq i}^{B}$ form a quadruple for model training.

$$\hat{\mathcal{L}}_i = -\frac{1}{2|\mathbf{U}_i^-|} \sum_{\boldsymbol{v}\in\{\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_i^+\}} \sum_{\boldsymbol{w}\in\mathbf{U}_i^-} \log\sigma\left[\lambda\left(\mathrm{sim}(\hat{\boldsymbol{u}}_i, \hat{\boldsymbol{u}}_i^+) - \mathrm{sim}(\hat{\boldsymbol{u}}_i, \boldsymbol{v})\right) + (1-\lambda)\left(\mathrm{sim}(\hat{\boldsymbol{u}}_i, \boldsymbol{v}) - \mathrm{sim}(\hat{\boldsymbol{u}}_i, \boldsymbol{w})\right)\right]$$

  - **In-batch hard negative sampling**: for each pairwise ranking order, select the pair with the smallest similarity difference to facilitate model training.

$$\hat{\mathcal{L}}_i = -\log\sigma\left[\lambda\left(\mathrm{sim}(\hat{\boldsymbol{u}}_i, \hat{\boldsymbol{u}}_i^+) - \max_{\boldsymbol{v}\in\{\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_i^+\}} \mathrm{sim}(\hat{\boldsymbol{u}}_i, \boldsymbol{v})\right) + (1-\lambda)\left(\min_{\boldsymbol{v}\in\{\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_i^+\}} \mathrm{sim}(\hat{\boldsymbol{u}}_i, \boldsymbol{v}) - \max_{\boldsymbol{w}\in\mathbf{U}_i^-} \mathrm{sim}(\hat{\boldsymbol{u}}_i, \boldsymbol{w})\right)\right]$$

  - The loss function $\tilde{\mathcal{L}}_i$ for another augmented sequence $\tilde{S}_i$ is symmetrically defined and the overall loss is computed as $\mathcal{L} = \sum_{i=1}^{B} (\hat{\mathcal{L}}_i + \tilde{\mathcal{L}}_i)/2B$.

Methodology

NEURAL INFORMATION PROCESSING SYSTEMS

Augmentation-Adaptive Fusion

- The effects of data augmentation vary significantly across different behavior sequences.
- The constant hyper-parameter λ applies a fixed and unified constraint to all samples.

$$\hat{\mathcal{L}}_i = -\log\sigma\left[\lambda\left(\mathrm{sim}(\hat{\boldsymbol{u}}_i, \hat{\boldsymbol{u}}_i^+) - \max_{\boldsymbol{v}\in\{\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_i^+\}} \mathrm{sim}(\hat{\boldsymbol{u}}_i, \boldsymbol{v})\right) + (1-\lambda)\left(\min_{\boldsymbol{v}\in\{\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_i^+\}} \mathrm{sim}(\hat{\boldsymbol{u}}_i, \boldsymbol{v}) - \max_{\boldsymbol{w}\in\mathbf{U}_i^-} \mathrm{sim}(\hat{\boldsymbol{u}}_i, \boldsymbol{w})\right)\right]$$

- Replace λ with a **dynamic coefficient** λ_i, which is estimated based on the average similarity between the user representations generated from $\hat{S}_i$ and $\tilde{S}_i$.

$$\boxed{\lambda_i = 1 - \frac{1}{4}\sum_{\hat{\boldsymbol{s}}\in\{\hat{\boldsymbol{u}}_i, \hat{\boldsymbol{u}}_i^+\}} \sum_{\tilde{\boldsymbol{s}}\in\{\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{u}}_i^+\}} \max(\mathrm{sim}(\hat{\boldsymbol{s}}, \tilde{\boldsymbol{s}}), 0)}$$

- If $\hat{S}_i$ and $\tilde{S}_i$ are semantically similar, λ_i will be small and force the user model to discriminate these similar explicitly augmented views from views of other users.
- Otherwise, λ_i will be large and train the user model to pull the implicitly augmented view and these dissimilar explicitly augmented views apart.

Experiments

Datasets and Downstream tasks

- Tencent Transfer Learning (TTL) dataset
  - $\mathcal{T}_1$: age prediction
  - $\mathcal{T}_2$: life status prediction
  - $\mathcal{T}_3$: click recommendation
  - $\mathcal{T}_4$: thumb-up recommendation
- App dataset
  - $\mathcal{T}_5$: gender prediction
  - $\mathcal{T}_6$: CVR prediction

Dataset	TTL				App	
# Behavior Sequences	1,470,149				1,575,837	
# Different Behaviors	645,972				4,047	
Avg. Sequence Length	54.84				44.13	
Downstream Task	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_6$
# Samples	1,470,147	1,020,277	1,397,197	255,646	1,178,603	564,940
# Labels/Items	8	6	17,879	7,539	2	2

Table 1: Detailed statistics of each dataset and downstream task.

Metrics

- Classification accuracy for multi-class classification tasks ($\mathcal{T}_1$, $\mathcal{T}_2$).
- NDCG@10 for cold-recommendation tasks ($\mathcal{T}_3$, $\mathcal{T}_4$).
- AUC for binary classification tasks ($\mathcal{T}_5$, $\mathcal{T}_6$).

Experiments

Overall Performance on Downstream Tasks

Pre-train Method	$\mathcal{T}_1$		$\mathcal{T}_2$		$\mathcal{T}_3$		$\mathcal{T}_4$		$\mathcal{T}_5$		$\mathcal{T}_6$	
	Acc	Impr	Acc	Impr	NDCG@10	Impr	NDCG@10	Impr	AUC	Impr	AUC	Impr
None	62.87±0.05	-	52.24±0.16	-	1.99±0.03	-	2.87±0.07	-	78.63±0.06	-	75.14±0.14	-
PeterRec	63.62±0.11	1.19	53.14±0.07	1.72	2.37±0.02	19.10	3.06±0.08	6.62	79.61±0.13	1.25	76.04±0.10	1.20
PTUM	63.21±0.14	0.54	53.05±0.04	1.55	2.29±0.03	15.08	2.96±0.03	3.14	79.48±0.11	1.08	75.82±0.13	0.90
CLUE	63.38±0.10	0.81	53.23±0.05	1.90	2.38±0.02	19.60	3.05±0.21	6.27	79.90±0.06	1.62	76.03±0.16	1.18
CCL	63.76±0.11	1.42	53.37±0.09	2.16	2.43±0.02	22.11	3.32±0.13	15.68	80.22±0.07	2.02	77.35±0.10	2.94
IDICL	63.88±0.04	1.61	53.45±0.05	2.32	2.46±0.02	23.62	3.42±0.04	19.16	80.34±0.05	2.17	77.92±0.08	3.70
CL4SRec	63.71±0.14	1.34	53.43±0.05	2.28	2.41±0.03	21.11	3.29±0.06	14.63	80.14±0.08	1.92	77.02±0.05	2.50
CoSeRec	63.89±0.03	1.62	53.53±0.09	2.47	2.44±0.02	22.61	3.33±0.05	16.03	80.48±0.06	2.35	77.71±0.09	3.42
DuoRec	63.50±0.09	1.00	53.26±0.06	1.95	2.39±0.01	20.10	3.11±0.16	8.36	80.03±0.09	1.78	76.85±0.09	2.28
AdaptSSR	**65.53±0.04**	**4.23**	**54.41±0.02**	**4.15**	**2.61±0.03**	**31.16**	**3.73±0.03**	**29.97**	**82.30±0.03**	**4.67**	**79.92±0.05**	**6.36**

Table 2: Performance (%) of various pre-training methods on downstream tasks. Impr (%) indicates the relative improvement compared with the end-to-end training. The best results are **bolded**.

Experiments

Performance with Different Data Augmentation Methods

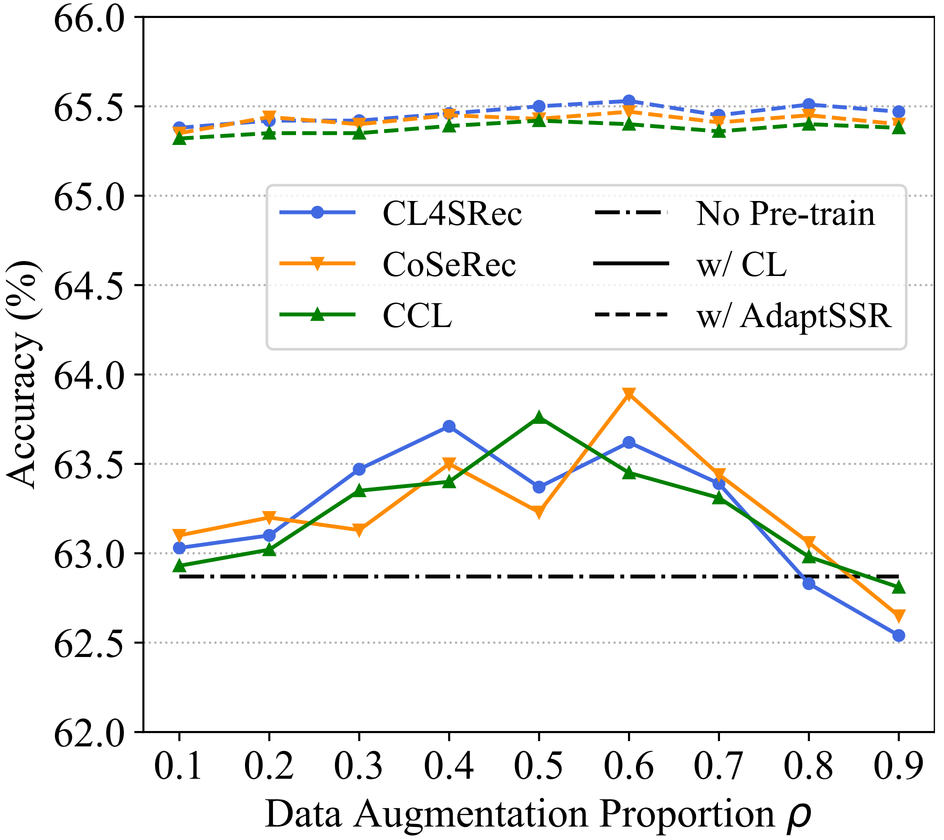


Figure 3: Effectiveness of AdaptSSR when combined with existing pre-training methods.

Ablation Study

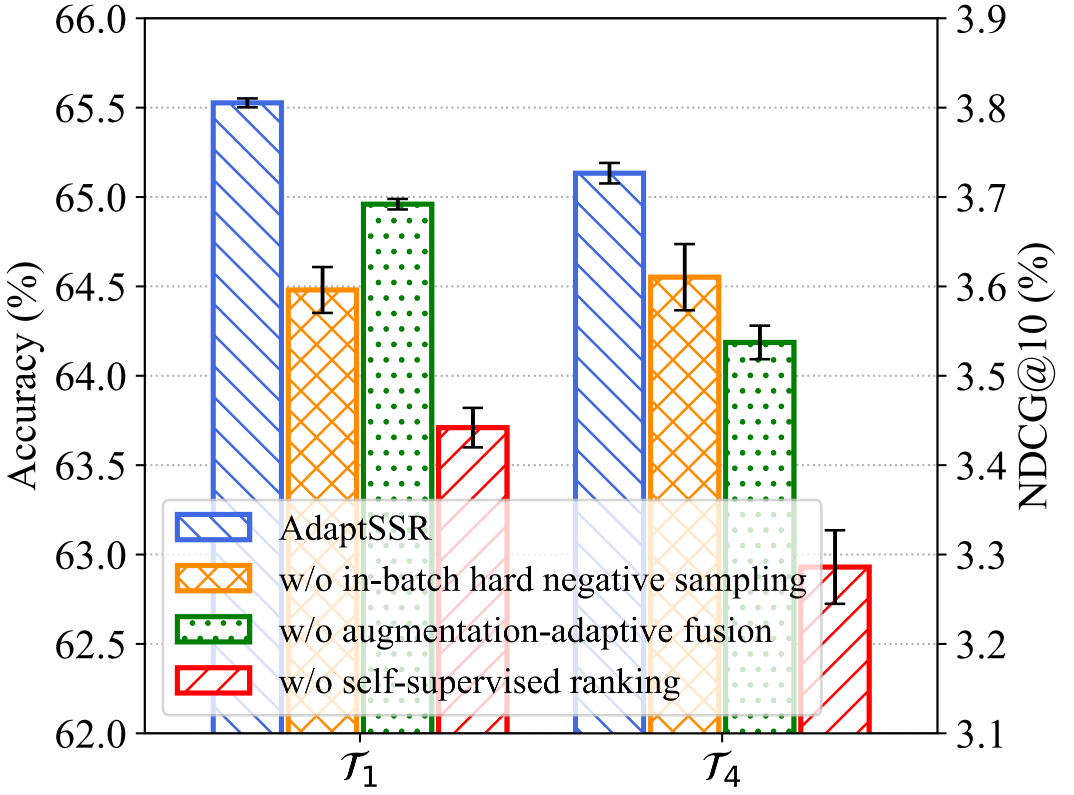


Figure 4: Effectiveness of each component in our AdaptSSR.

Experiments

NEURAL INFORMATION PROCESSING SYSTEMS

User Representation Similarity Distribution Analysis

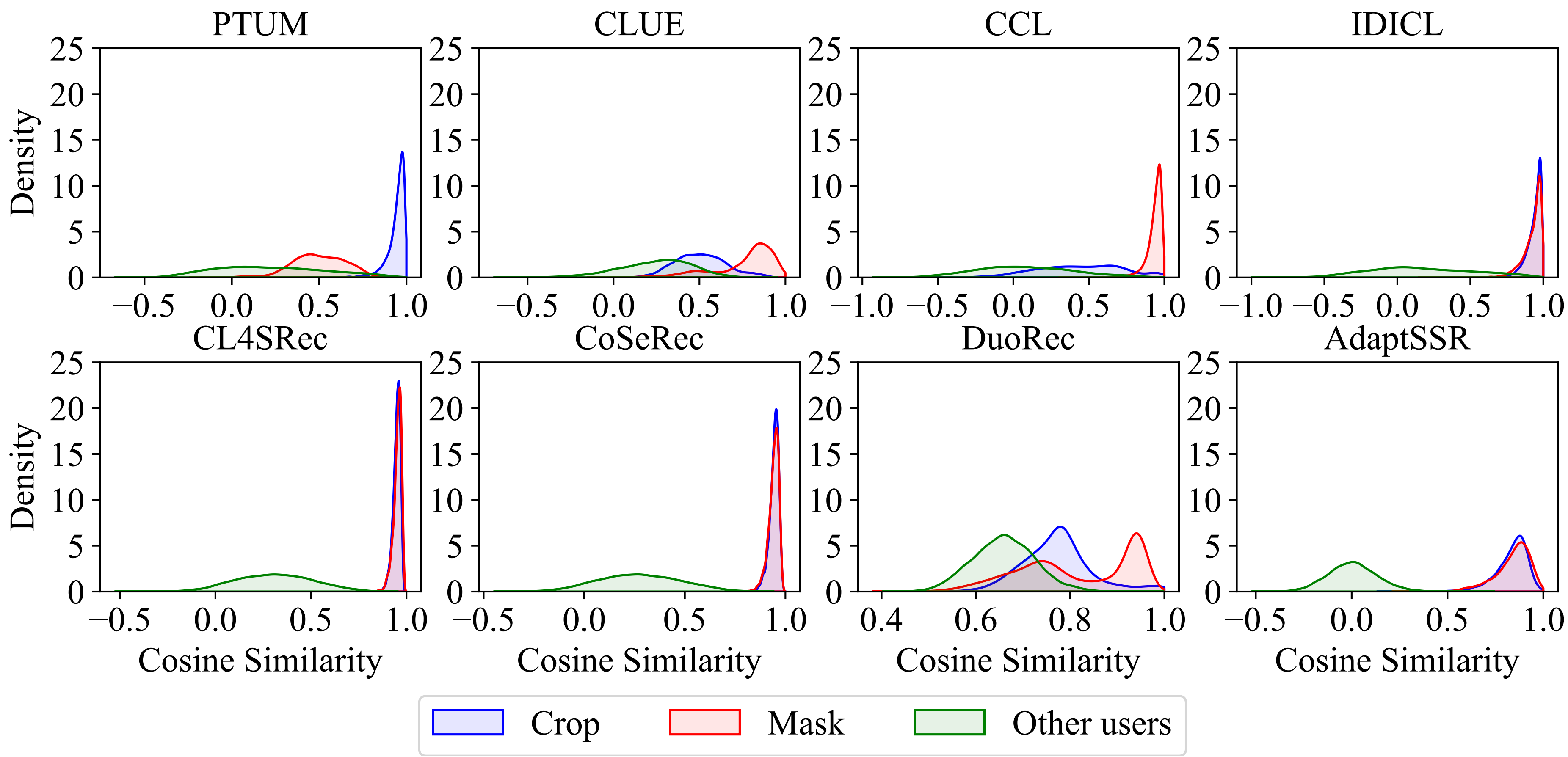


Figure 5: Distributions of the cosine similarity between user representations generated from the original behavior sequence, different augmented behavior sequences, and the behavior sequences of other users with various pre-training methods. The area under each curve equals to 1.

Conclusion

- We identified the **semantic inconsistency problem** faced by existing contrastive learning-based user model pre-training methods.
- **Augmentation-Adaptive Self-Supervised Ranking (AdaptSSR)**
  - Train the user model to capture the similarity between the implicitly augmented view, the explicitly augmented view, and views from other users with a **multiple pairwise ranking** loss.
  - Facilitate model training with **in-batch hard negative sampling**.
  - Adjust the similarity order constraint applied to each sample based on the estimated similarity between the augmented views with an **augmentation-adaptive fusion** mechanism.
- Extensive experiments validated the effectiveness of our method.

Paper

Code

Thanks For Your Attention