CORE: Simple and Effective Session-based Recommendation within Consistent Representation Space

Yupeng Hou†
houyupeng@ruc.edu.cn
Gaoling School of Artificial Intelligence, Renmin University of China

Binbin Hu
bin.hbb@antfin.com
Ant Group

Zhiqiang Zhang
lingyao.zzq@antfin.com
Ant Group

Wayne Xin Zhao✉
batmanfly@gmail.com
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing Key Laboratory of Big Data Management and Analysis Methods
Beijing Academy of Artificial Intelligence

SIGIR 2022

Code:https://github.com/RUCAIBox/CORE.

gesis Leibniz-Institut für Sozialwissenschaften

Reported by liang li

Motivation

Motivation：

- session embedding learned by a non-linear encoder is usually not in the same representation space as item embeddings, resulting in the inconsistent prediction issue while recommending items.

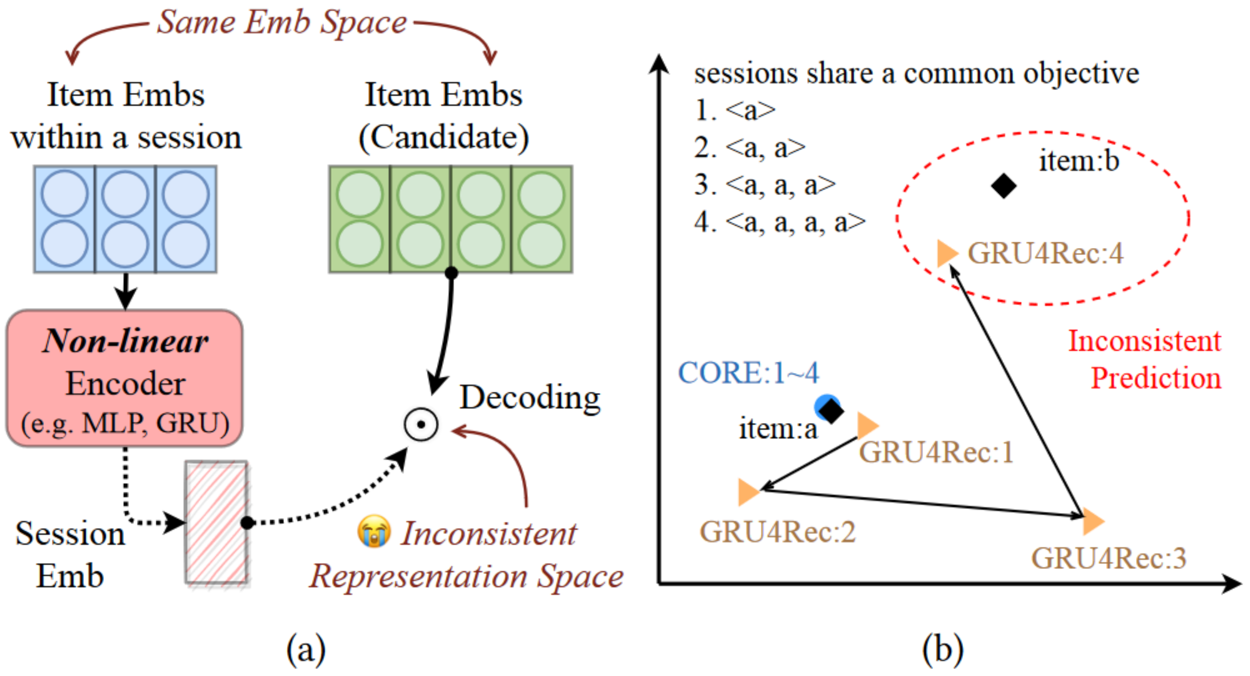


Figure 1: (a) Encoder-decoder framework of most existing session-based recommendation models and (b) Inconsistent prediction issue while measuring the distance between embeddings for recommending.

Problem Statement

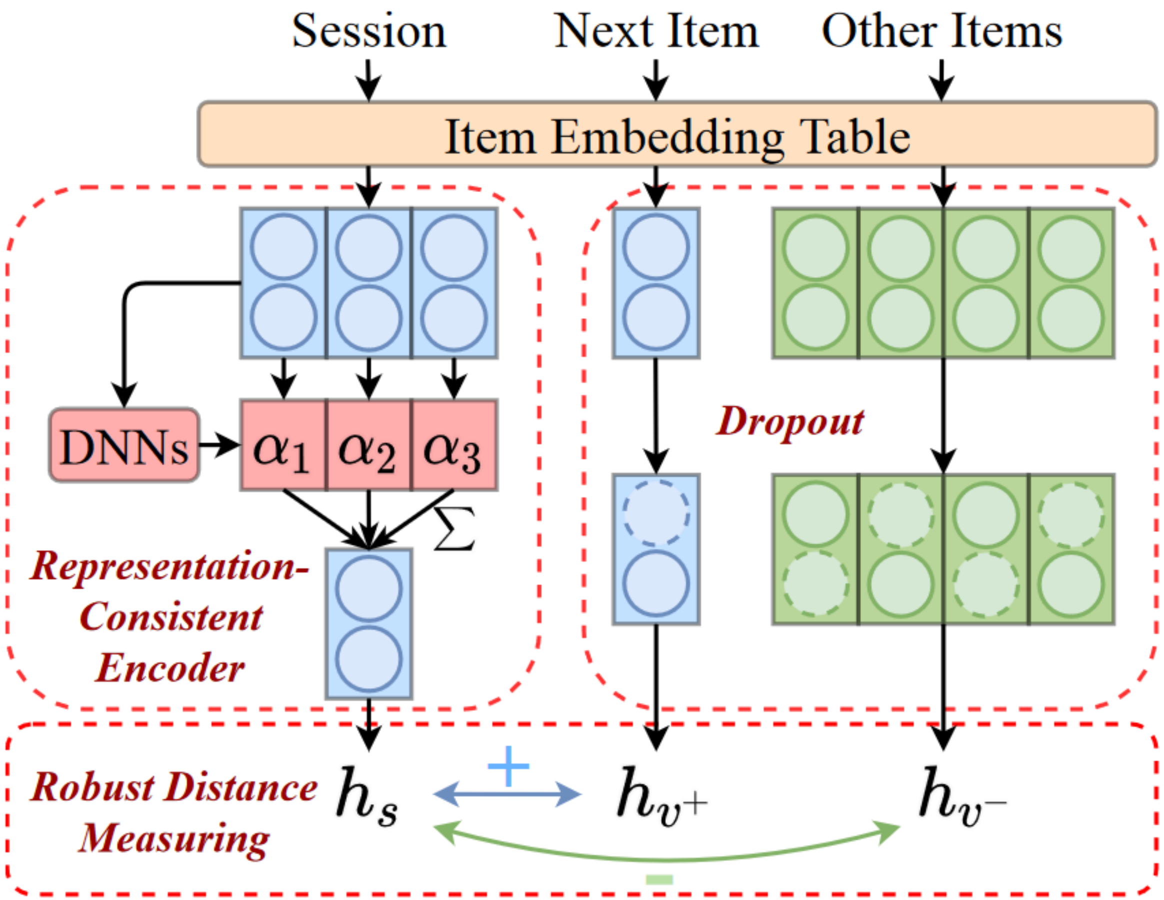


Figure 2: Overall framework of CORE.

$\boldsymbol{h}_i = \mathrm{Emb}(v_i) \in \mathbb{R}^d$ denotes the item embedding for item v_i, where $\mathrm{Emb}(\cdot)$ is the item embedding look-up table and d is the dimension of vectors. Then we have $\boldsymbol{h}_s = \mathrm{Encoder}([\boldsymbol{h}_{s,1}, \ldots, \boldsymbol{h}_{s,n}]) \in \mathbb{R}^d$ to encode a session s with n items, where $\mathrm{Encoder}(\cdot)$ is usually a non-linear neural network. Finally, we can predict the probability distribution for the next item, *i.e.*, $\hat{\boldsymbol{y}} = \mathrm{Decoder}(\boldsymbol{h}_s) \in \mathbb{R}^m$, where m is the number of all items.

Method

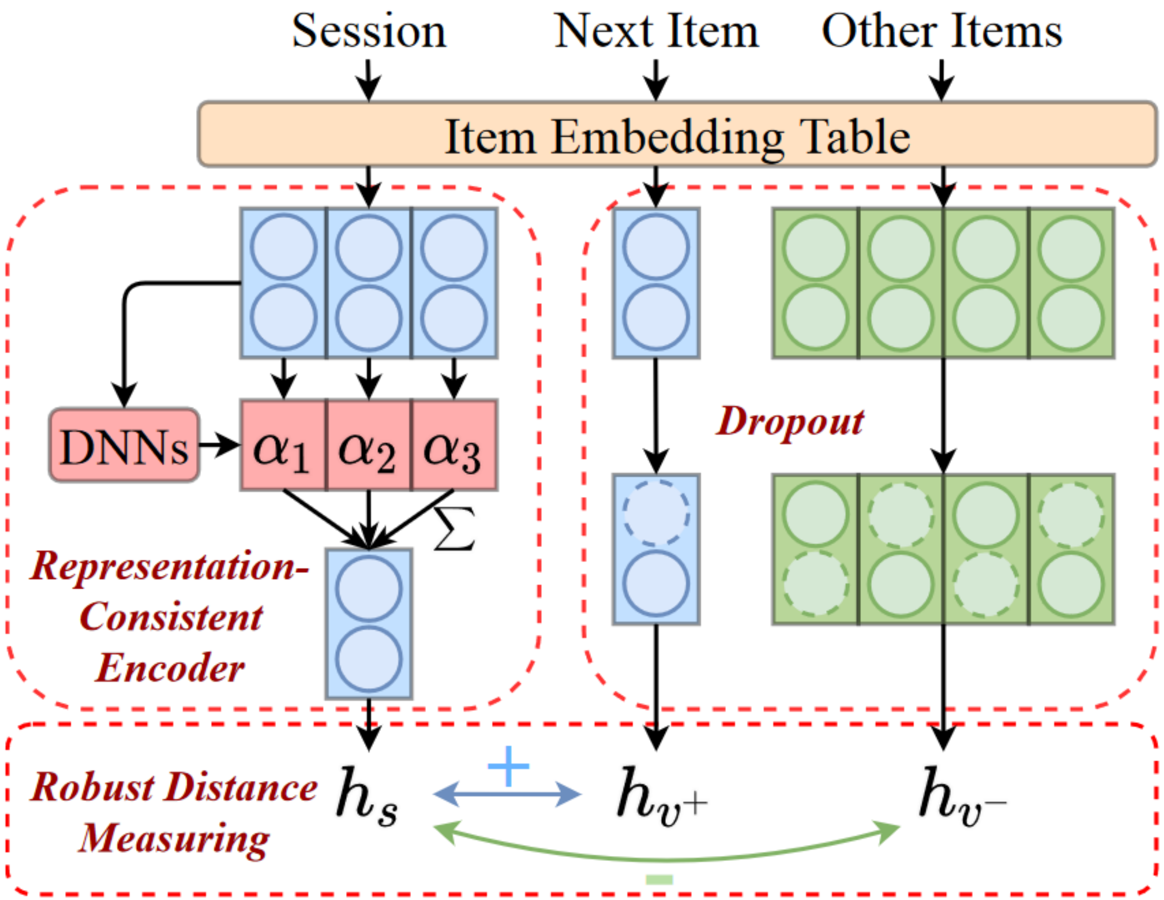


Figure 2: Overall framework of CORE.

$$\boldsymbol{\alpha} = \mathrm{DNN}([\boldsymbol{h}_{s,1}; \boldsymbol{h}_{s,2}; \ldots; \boldsymbol{h}_{s,n}]), \quad (1)$$

$$\boldsymbol{h}_s = \sum_{i=1}^{n} \alpha_i \boldsymbol{h}_{s,i}, \quad (2)$$

$$\alpha_i = \frac{1}{n}$$

$$\boldsymbol{F} = \mathrm{Transformers}([\boldsymbol{h}_{s,1}; \boldsymbol{h}_{s,2}; \ldots; \boldsymbol{h}_{s,n}]), \quad (3)$$

$$\boldsymbol{\alpha} = \mathrm{softmax}(\boldsymbol{w} \cdot \boldsymbol{F}^\top), \quad (4)$$

Method

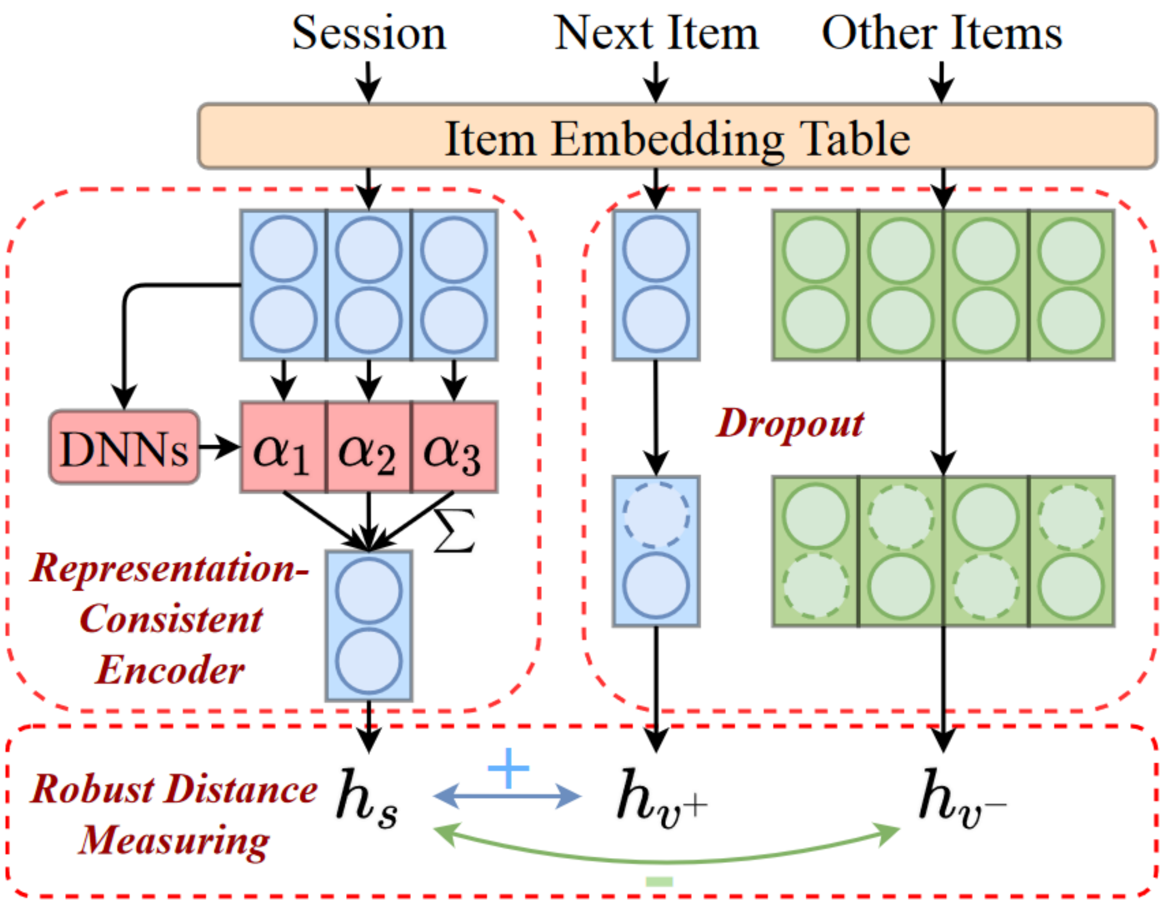


Figure 2: Overall framework of CORE.

$$
\begin{aligned}
\ell_{\text{ori}} &= -\log \frac{\exp(\boldsymbol{h}_s \cdot \boldsymbol{h}_{v^+})}{\sum_{i=1}^{m} \exp(\boldsymbol{h}_s \cdot \boldsymbol{h}_{v_i})} \\
&= \log \left[1 + (|\mathcal{V}| - 1) \sum_{v^- \in \mathcal{V} \setminus \{v^+\}} \exp(\boldsymbol{h}_s \boldsymbol{h}_{v^-} - \boldsymbol{h}_s \boldsymbol{h}_{v^+}) \right], \\
&\simeq (|\mathcal{V}| - 1) \sum_{v^- \in \mathcal{V} \setminus \{v^+\}} \exp(\boldsymbol{h}_s \boldsymbol{h}_{v^-} - \boldsymbol{h}_s \boldsymbol{h}_{v^+}) \\
&\simeq (|\mathcal{V}| - 1) \sum_{v^- \in \mathcal{V} \setminus \{v^+\}} (\boldsymbol{h}_s \boldsymbol{h}_{v^-} - \boldsymbol{h}_s \boldsymbol{h}_{v^+} + 1), \\
&\propto \sum_{v^- \in \mathcal{V} \setminus \{v^+\}} \left(\|\boldsymbol{h}_s - \boldsymbol{h}_{v^+}\|^2 - \|\boldsymbol{h}_s - \boldsymbol{h}_{v^-}\|^2 + 2 \right).
\end{aligned}
$$

$$
\ell = -\log \frac{\exp\left(\cos(\boldsymbol{h}_s, \boldsymbol{h}'_{v^+})/\tau\right)}{\sum_{i=1}^{m} \exp\left(\cos(\boldsymbol{h}_s, \boldsymbol{h}'_{v_i})/\tau\right)}, \tag{5}
$$

Experiments

Table 1: Overall performance comparison on five datasets. "$*$" indicates the statistical significance for $p < 0.01$ compared to the best baseline method with paired t-test. Sessions are split into train/validation/test set in a ratio of 8:1:1 for fair evaluation. We indicate performances of FPMC on Yoochoose as "–" due to the OOM issue.

Dataset	Metric	FPMC	GRU4Rec	NARM	SR-GNN	NISER+	LESSR	SGNN-HN	SASRec	GC-SAN	CL4Rec	CORE-ave	CORE-trm	Improv.
Diginetica	R@20	31.83	45.43	47.68	48.76	51.23	48.80	50.89	49.86	50.95	50.03	50.21	**52.89***	+3.24%
	M@20	8.79	14.77	15.58	16.93	18.32	16.96	17.25	17.19	17.84	17.26	18.07	**18.58***	+1.42%
Nowplaying	R@20	10.18	13.80	14.17	15.28	16.55	17.60	16.75	20.69	18.30	20.59	20.31	**21.81***	+5.41%
	M@20	4.51	5.83	6.11	6.10	7.14	7.13	6.13	**8.14**	8.13	8.21	6.62	7.35	–
RetailRocket	R@20	46.04	55.32	58.65	58.71	60.36	56.22	58.82	59.81	60.18	59.69	59.18	**61.85***	+2.47%
	M@20	21.95	33.18	34.69	36.42	37.43	37.11	35.72	36.03	36.85	35.95	37.52*	**38.76***	+3.55%
Tmall	R@20	20.30	23.25	31.67	33.65	35.97	32.45	39.14	35.82	35.32	35.59	**44.67***	44.48*	+14.13%
	M@20	13.07	15.78	21.83	25.27	27.06	23.96	23.46	25.10	23.48	25.07	**31.85***	31.72*	+17.70%
Yoochoose	R@20	–	60.78	61.67	61.84	62.99	62.89	62.49	63.55	63.24	63.61	58.83	**64.61***	+1.57%
	M@20	–	27.27	27.82	28.15	28.98	28.59	28.24	28.63	**29.00**	28.73	25.05	28.24	–

Table 2: Statistics of the datasets.

Dataset	# Interactions	# Items	# Sessions	Avg. Length
Diginetica	786,582	42,862	204,532	4.12
Nowplaying	1,085,410	59,593	145,612	9.21
RetailRocket	871,637	51,428	321,032	6.40
Tmall	427,797	37,367	66,909	10.62
Yoochoose	1,434,349	19,690	470,477	4.64

Experiments

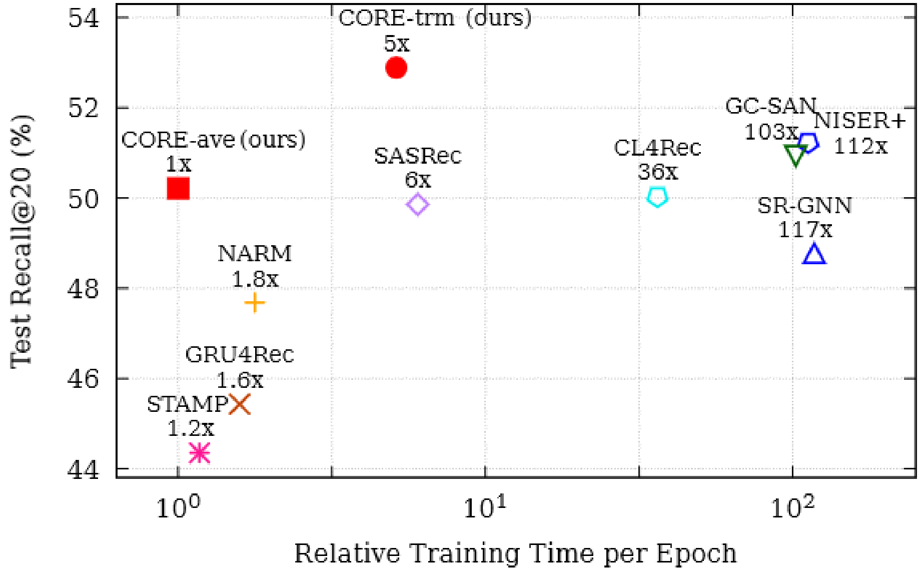


Figure 3: Performances over training time relative to that of CORE-ave on Diginetica.

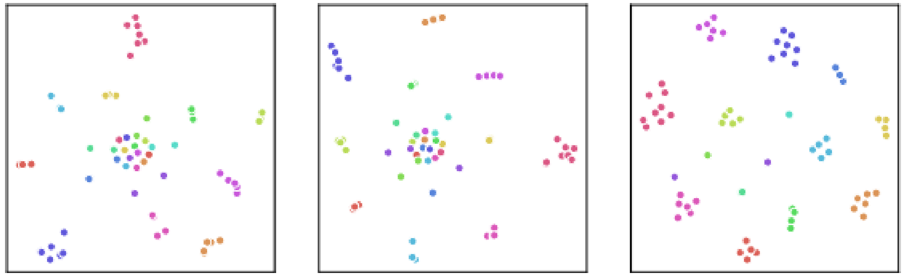
Table 3: Ablation study of CORE's variants on Diginetica and RetailRocket.

Method	Diginetica		RetailRocket	
	R@20	M@20	R@20	M@20
CORE	52.89	18.58	61.85	38.76
w/o RCE	49.82	17.41	59.59	36.27
w/o RDM	52.31	18.38	60.93	37.72
SASRec	49.86	17.19	59.81	36.03

Experiments

Table 4: Performance comparison of different methods and their improved variants on two datasets.

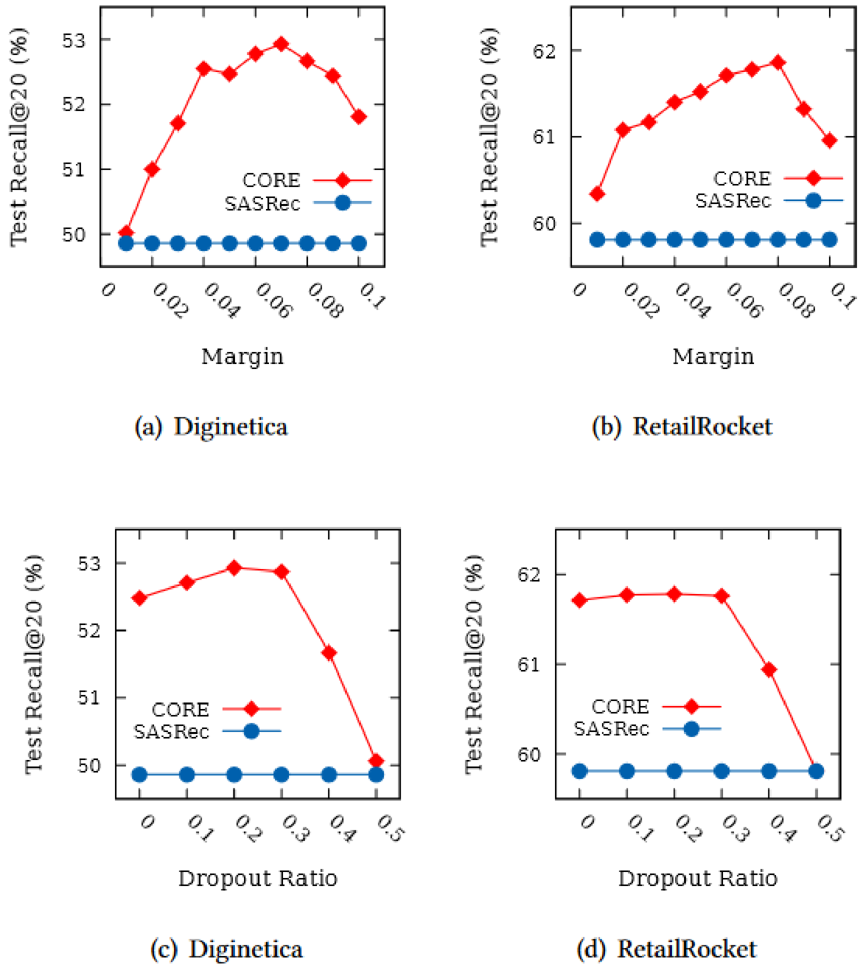
Method	Diginetica R@20	Diginetica M@20	RetailRocket R@20	RetailRocket M@20
NARM	47.68	15.58	58.65	34.69
+ RCE	51.86	18.27	60.77	37.01
+ RDM	51.62	17.79	61.33	37.11
+ All	52.51	18.58	62.19	38.84
SR-GNN	48.76	16.93	58.71	36.42
+ RCE	49.51	17.53	57.05	35.70
+ RDM	51.36	18.57	61.41	38.27
+ All	52.38	18.95	61.43	38.38



(a) GRU4Rec (b) SASRec (c) CORE (ours)

Figure 4: Visualization of learned session embeddings.

Experiments



(a) Diginetica (b) RetailRocket

(c) Diginetica (d) RetailRocket

Figure 5: Parameter tuning of CORE on Diginetica and RetailRocket datasets.

Thanks