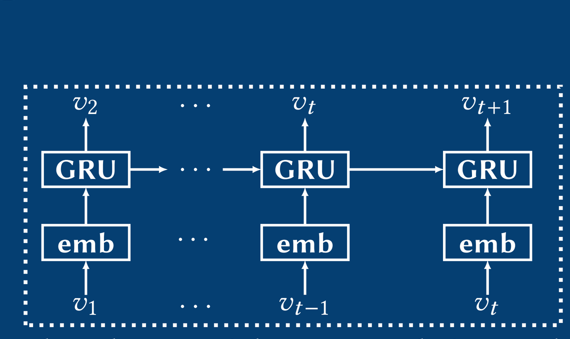
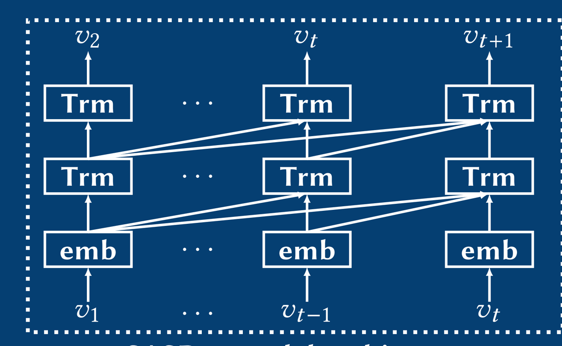
Instead of left-to-right **unidirectional model** & **predicting next**

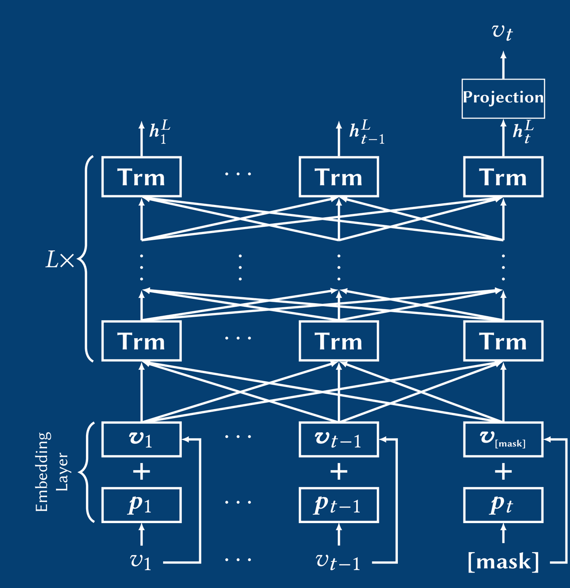


RNN based sequential recommendation methods.



SASRec model architecture.

we use **bidirectional self-attention model** & **cloze task**



for representation learning in **sequential recommendation**

BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer

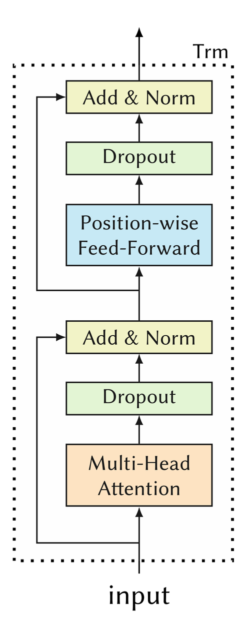
Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang

Motivation

$$p(v^{(u)}_{n_u+1} = v \mid \mathcal{S}_u), \quad \mathcal{S}_u = [v^{(u)}_1, \ldots, v^{(u)}_t, \ldots, v^{(u)}_{n_u}]$$

- Unidirectional models restrict the power of hidden representations for items in the historical sequences.
- Rigid order assumption in unidirectional sequential models is not always right in user behavior sequences.
- User behaviors are often noisy due to a variety of unobservable external factors. They usually are roughly chronological, but not rigidly ordered.

Model



$$\mathtt{Att}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d/h}}\right)\boldsymbol{V}$$
$$\mathrm{head}_i = \mathtt{Att}\left(\boldsymbol{H}^l\boldsymbol{W}^Q_i, \boldsymbol{H}^l\boldsymbol{W}^K_i, \boldsymbol{H}^l\boldsymbol{W}^V_i\right)$$
$$\mathtt{MH}(\boldsymbol{H}^l) = [\mathrm{head}_1; \mathrm{head}_2; \ldots; \mathrm{head}_h]\boldsymbol{W}^O$$
$$\mathtt{FFN}(\boldsymbol{x}) = \mathtt{GELU}\left(\boldsymbol{x}\boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)}\right)\boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)}$$
$$\mathtt{GELU}(x) = x\Phi(x), \Phi(x) \text{ is CDF of standard normal}$$
$$\mathtt{PFFN}(\boldsymbol{H}^l) = \left[\mathtt{FFN}(\boldsymbol{h}^l_1)^\top; \ldots; \mathtt{FFN}(\boldsymbol{h}^l_t)^\top\right]^\top$$
$$\boldsymbol{A}^{l-1} = \mathtt{LN}\left(\boldsymbol{H}^{l-1} + \mathtt{Dropout}\left(\mathtt{MH}(\boldsymbol{H}^{l-1})\right)\right)$$
$$\mathtt{Trm}(\boldsymbol{H}^{l-1}) = \mathtt{LN}\left(\boldsymbol{A}^{l-1} + \mathtt{Dropout}\left(\mathtt{PFFN}(\boldsymbol{A}^{l-1})\right)\right)$$
$$\boldsymbol{H}^l = \mathtt{Trm}\left(\boldsymbol{H}^{l-1}\right), \quad \forall i \in [1, \ldots, L]$$
$$\boldsymbol{h}^0_i = \boldsymbol{v}_i + \boldsymbol{p}_i$$

Learning *cloze* task/random item mask

Samples: **Input**: $[v_1, v_2, v_3, v_4, v_5] \xrightarrow{\text{randomly mask}} [v_1, \texttt{[mask]}_1, v_3, \texttt{[mask]}_2, v_5]$
Labels: $\texttt{[mask]}_1 = v_2, \quad \texttt{[mask]}_2 = v_4$

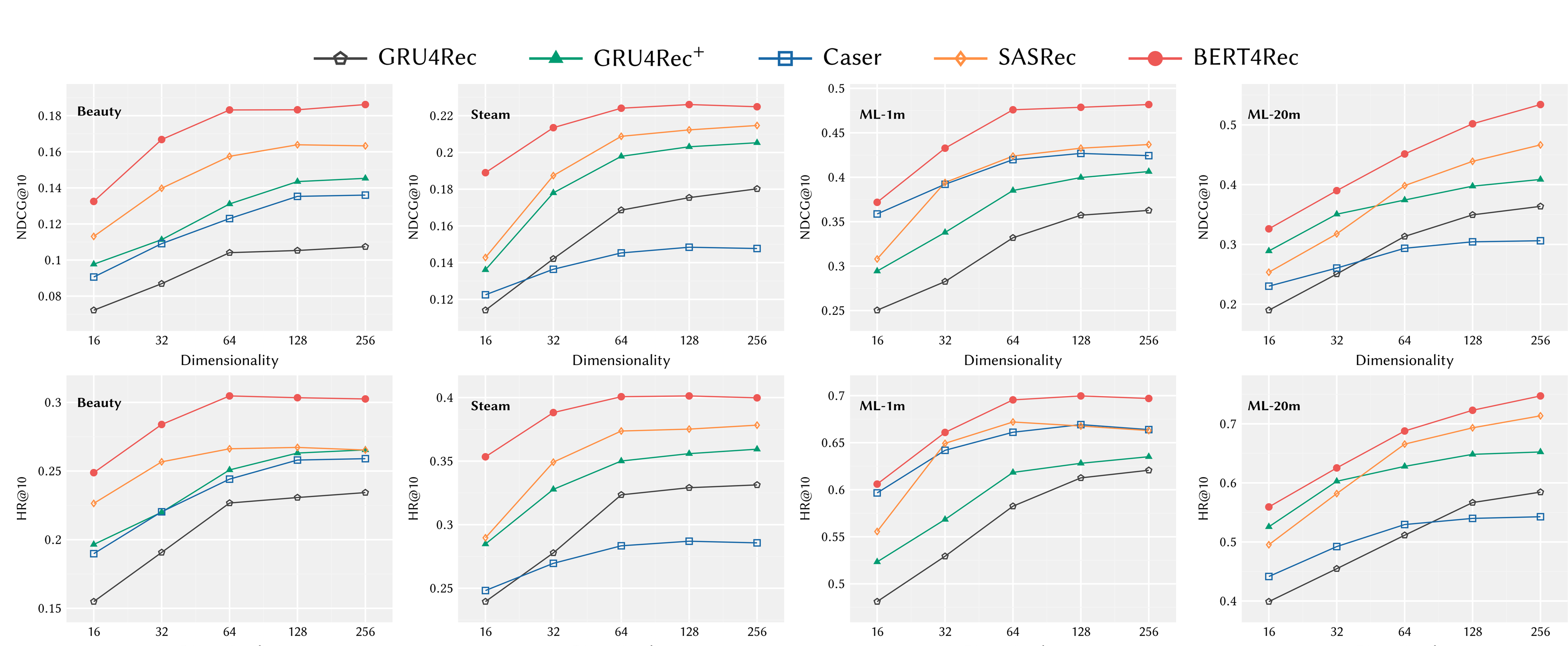
Training:

$$\mathcal{L} = \frac{1}{|\mathcal{S}^m_u|}\sum_{v_m \in \mathcal{S}^m_u} -\log P(v_m = v^*_m | \mathcal{S}'_u)$$

$$P(v) = \mathrm{softmax}\left(\mathtt{GELU}(\boldsymbol{h}^L_t\boldsymbol{W}^P + \boldsymbol{b}^P)\boldsymbol{E}^\top + \boldsymbol{b}^O\right)$$

Predicting: $\mathcal{S}_u$.append("`[mask]`") & predict

Impact of hidden dimensionality d



Experiments

Datasets

Datasets	#users	#items	#actions	Avg. length	Density
Beauty	40,226	54,542	0.35m	8.8	0.02%
Steam	281,428	13,044	3.5m	12.4	0.10%
ML-1m	6040	3416	1.0m	163.5	4.79%
ML-20m	138,493	26,744	20m	144.4	0.54%

Task Settings & Evaluation Metrics & Baselines

Protocal	Metrics	Baselines
leave-one-out evaluation: $S_u = [v^{(u)}_1, \ldots, v^{(u)}_{n_u}]$ train: $[v^{(u)}_1, \ldots, v^{(u)}_{n_u-2}]$, val: $v^u_{n_u-1}$, test: $v^u_{n_u}$	$\mathrm{HR@}k = \frac{1}{\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}} \mathbb{1}(\mathrm{R}_{u,g_u} \le k)$	GRU4Rec
	$\mathrm{NDCG@}k = \frac{1}{\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}} \frac{2^{\mathbb{1}(\mathrm{R}_{u,g_u} \le k)} - 1}{\log_2(\mathrm{R}_{u,g_u} + 1)}$	Caser
Picking up the ground truth item from 100 randomly sampled (by spopularity) negative items	$\mathrm{MRR} = \frac{1}{\lvert\mathcal{U}\rvert}\sum_{u\in\mathcal{U}} \frac{1}{\mathrm{R}_{u,g_u}}$	GRU4Rec$^+$
		SASRec

Overall Performances

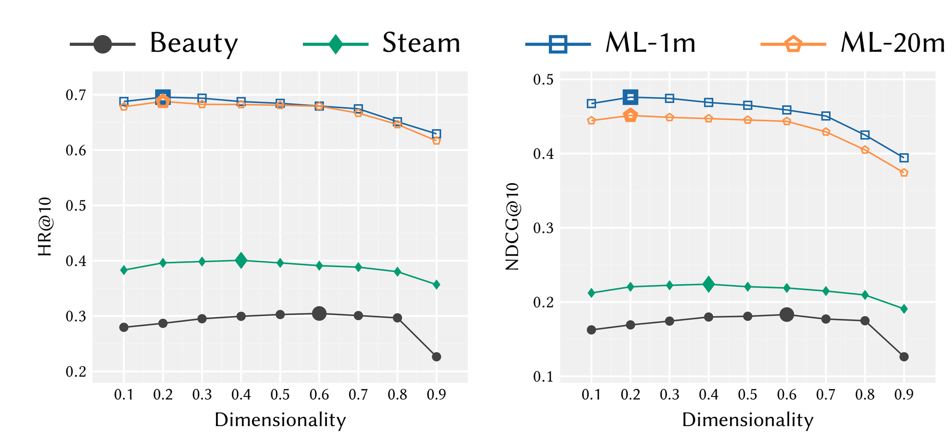
Datasets	Metric	GRU4Rec	GRU4Rec$^+$	Caser	SASRec	BERT4Rec	Improv.
Beauty	HR@5	0.1315	0.1781	0.1625	0.1934	**0.2207**	14.12%
	HR@10	0.2343	0.2654	0.2590	0.2653	**0.3025**	14.02%
	NDCG@5	0.0812	0.1172	0.1050	0.1436	**0.1599**	11.35%
	NDCG@10	0.1074	0.1453	0.1360	0.1633	**0.1862**	14.02%
	MRR	0.1023	0.1299	0.1205	0.1536	**0.1701**	10.74%
Steam	HR@5	0.2171	0.2391	0.1766	0.2559	**0.2710**	5.90%
	HR@10	0.3313	0.3594	0.2870	0.3783	**0.4013**	6.08%
	NDCG@5	0.1370	0.1613	0.1131	0.1727	**0.1842**	6.66%
	NDCG@10	0.1802	0.2053	0.1484	0.2147	**0.2261**	5.31%
	MRR	0.1420	0.1757	0.1305	0.1874	**0.1949**	4.00%
ML-1m	HR@5	0.4673	0.5103	0.5353	0.5434	**0.5876**	8.13%
	HR@10	0.6207	0.6351	0.6692	0.6629	**0.6970**	4.15%
	NDCG@5	0.3196	0.3705	0.3832	0.3980	**0.4454**	11.91%
	NDCG@10	0.3627	0.4064	0.4268	0.4368	**0.4818**	10.32%
	MRR	0.3041	0.3462	0.3648	0.3790	**0.4254**	12.24%
ML-20m	HR@5	0.4657	0.5118	0.3804	0.5727	**0.6323**	10.41%
	HR@10	0.5844	0.6524	0.5427	0.7136	**0.7473**	4.72%
	NDCG@5	0.3090	0.3630	0.2538	0.4208	**0.4967**	18.04%
	NDCG@10	0.3637	0.4087	0.3062	0.4665	**0.5340**	14.47%
	MRR	0.2967	0.3476	0.2529	0.4026	**0.4785**	18.85%

Effectiveness of bidirectional self-attention & Cloze objective

Model	Beauty			ML-1m		
	HR@10	NDCG@10	MRR	HR@10	NDCG@10	MRR
SASRec	0.2653	0.1633	0.1536	0.6629	0.4368	0.3790
BERT4Rec (1 mask)	0.2940	0.1769	0.1618	0.6869	0.4696	0.4127
BERT4Rec	0.3025	0.1862	0.1701	0.6970	0.4818	0.4254

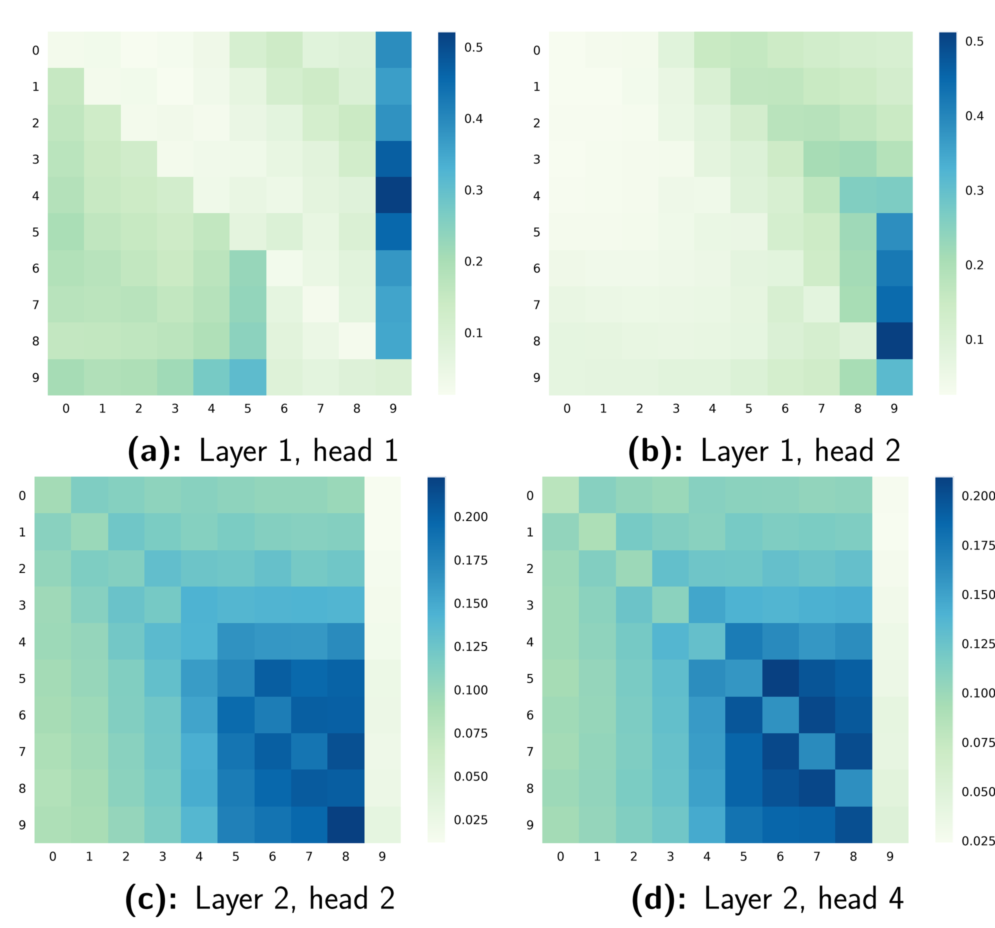
Extra Results

Impact of Mask Proportion ρ



Performance with different mask proportion ρ on $d = 64$. Bold symbols denote the best scores in each line.

Attention Visualization



Heat-maps of average attention weights on Beauty, the last position "9" denotes "`[mask]`" (best viewed in color).

- Attention varies across different heads.
- Attention varies across different layers.
- Items tend to attend on the items at both sides.

Ablation Study

Architecture	Dataset			
	Beauty	Steam	ML-1m	ML-20m
$L = 2$, $h = 2$	0.1832	0.2241	0.4759	0.4513
w/o PE	0.1741	0.2060	0.2155↓	0.2867↓
w/o PFFN	0.1803	0.2137	0.4544	0.4296
w/o LN	0.1642↓	0.2058	0.4334	0.4186
w/o RC	0.1619↓	0.2193	0.4643	0.4483
w/o Dropout	0.1658	0.2185	0.4553	0.4471
1 layer ($L = 1$)	0.1782	0.2122	0.4412	0.4238
3 layers ($L = 3$)	**0.1859**	**0.2262**	**0.4864**	**0.4661**
4 layers ($L = 4$)	**0.1834**	**0.2279**	**0.4898**	**0.4732**
1 head ($h = 1$)	**0.1853**	0.2187	0.4568	0.4402
4 heads ($h = 4$)	0.1830	**0.2245**	**0.4770**	**0.4520**
8 heads ($h = 8$)	0.1823	**0.2248**	0.4743	**0.4550**

Scan it!