BERT4Rec: Sequential Recommendation with BERT

Authors: Fei Sun, Jun Liu, Jian Wu, … from Alibaba Group
Presenter: Hong Fan Zhao

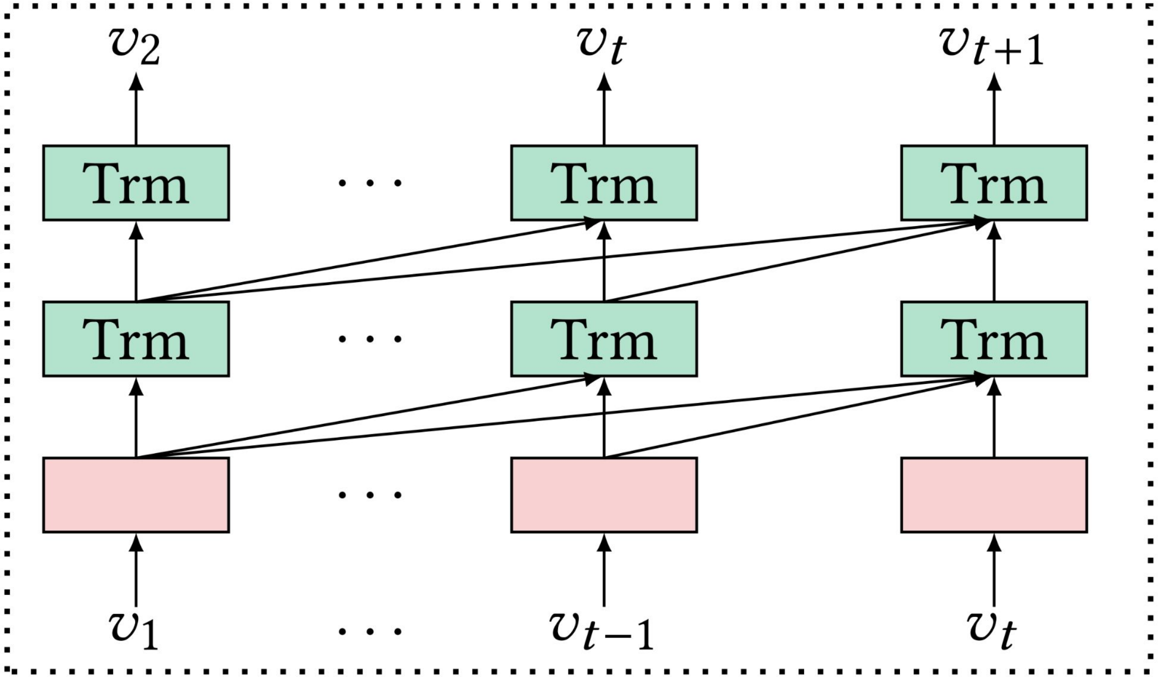
Sequential Recommendation

Sequential Recommendation

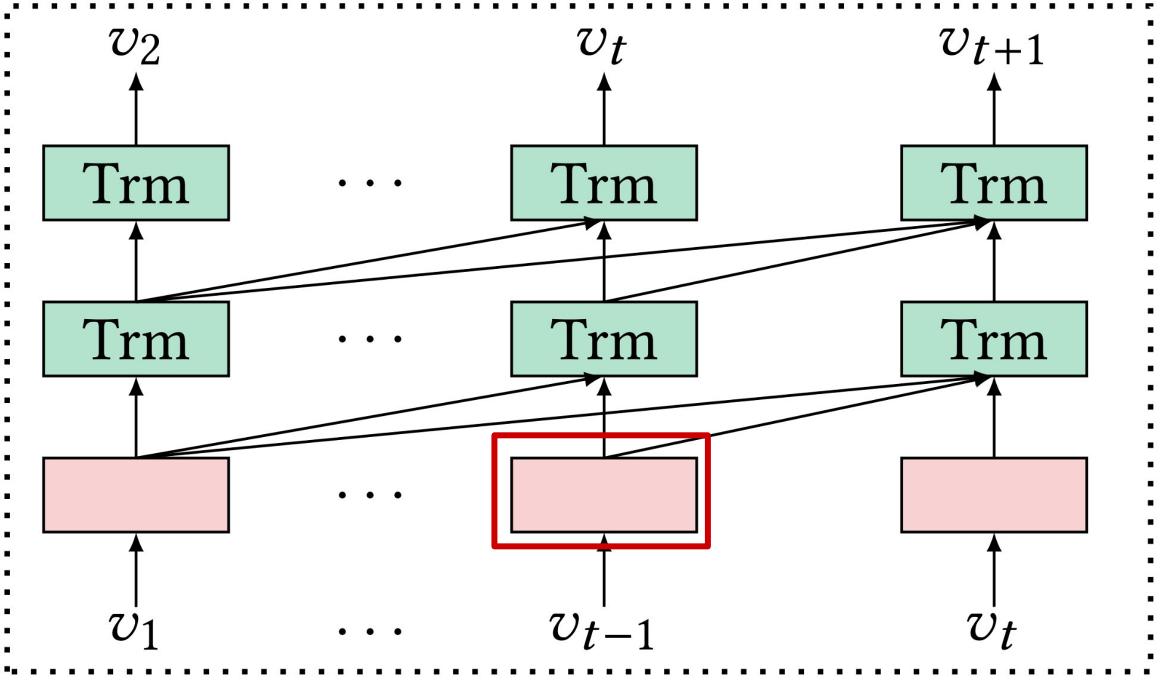
Sequential Recommendation

Previous Works

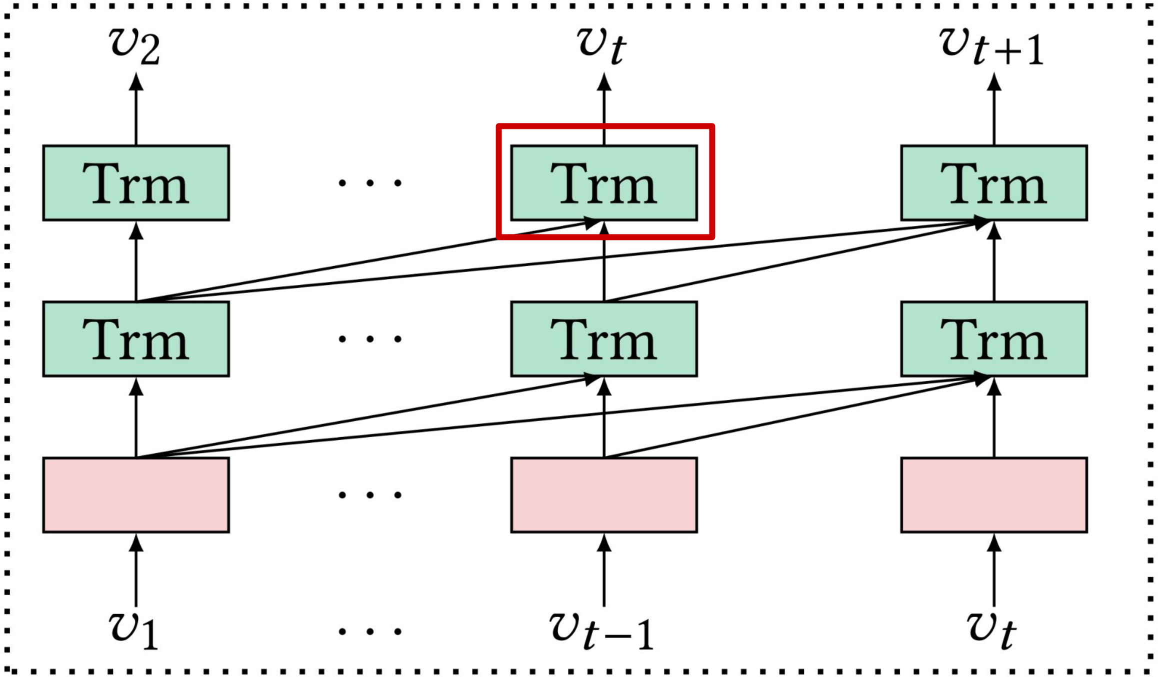
Previous Works: SASRec



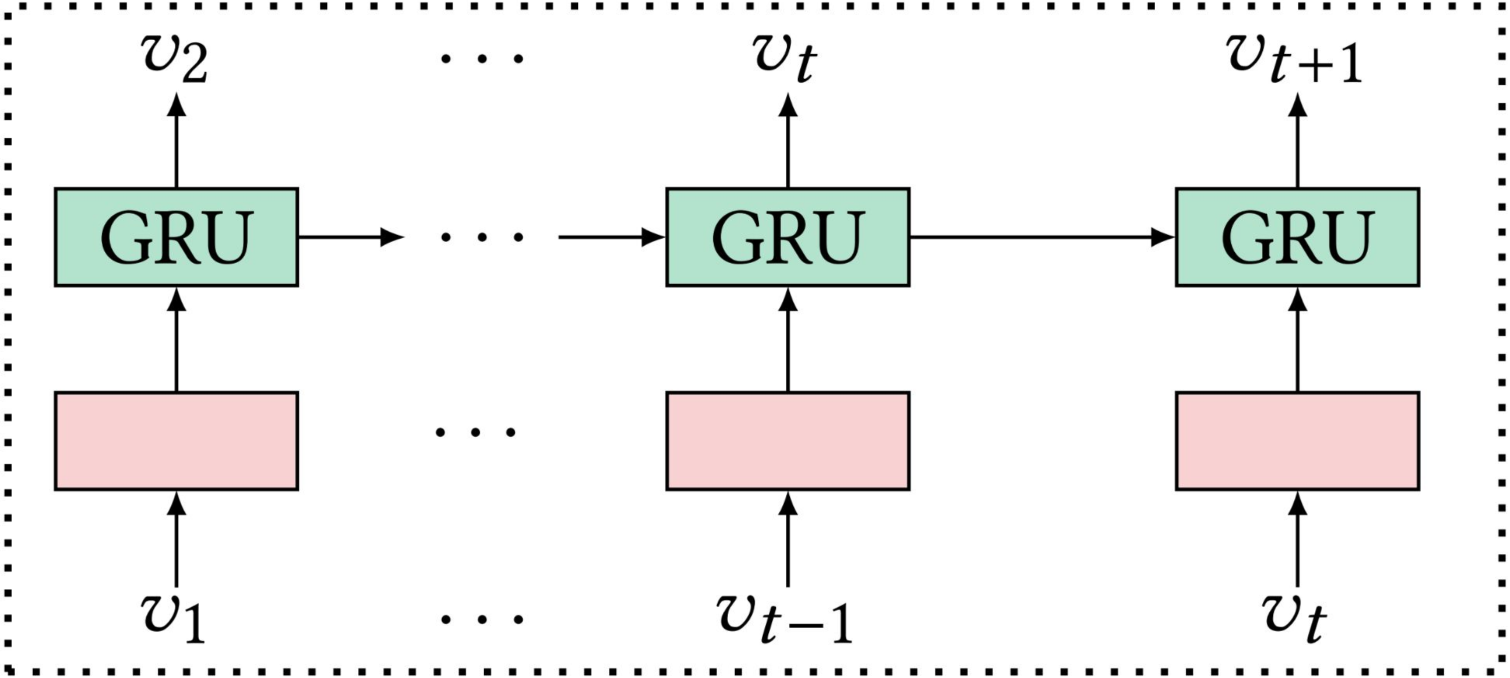
Previous Works: SASRec



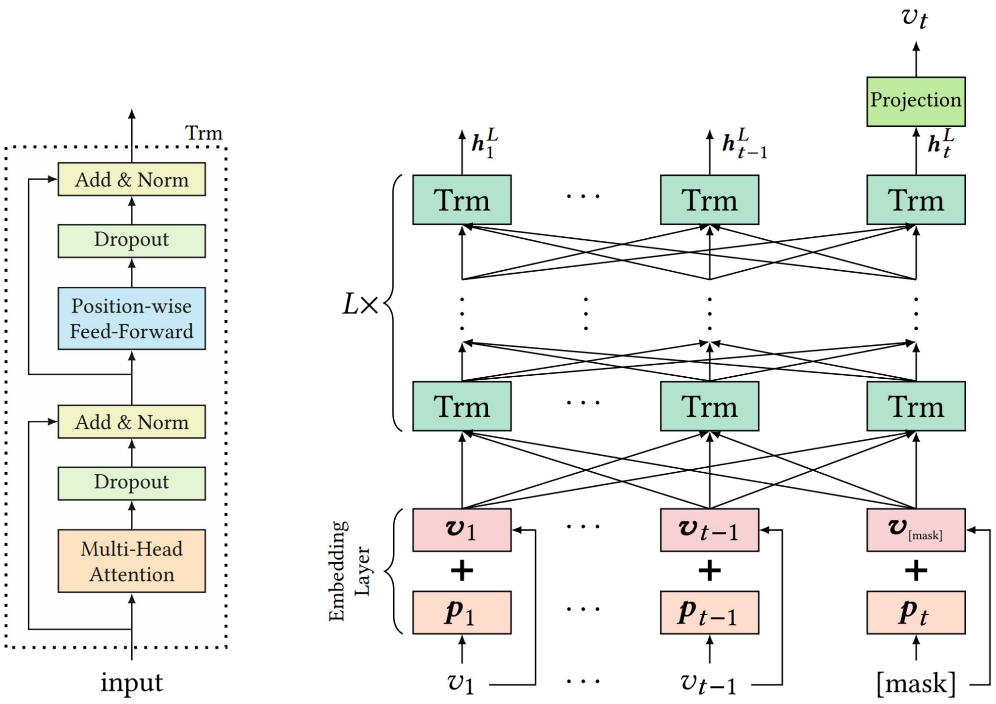
Previous Works: SASRec



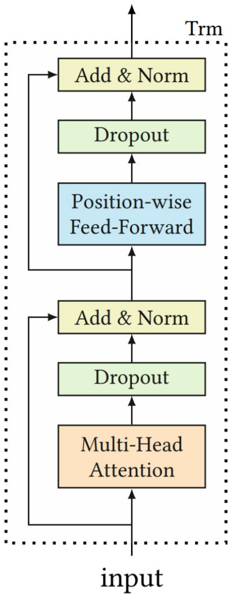
Previous Works: RNN Based Sequential Recommendation

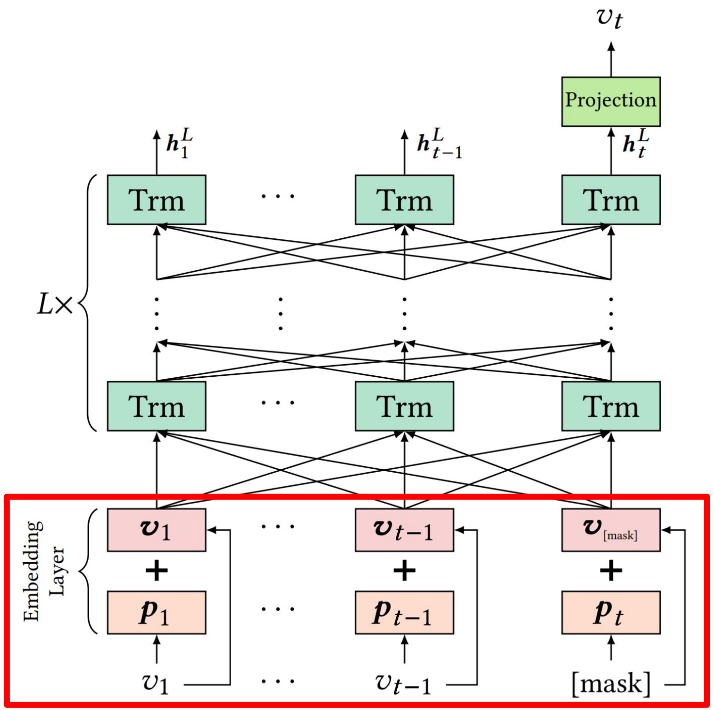


Bidirectional Encoder Representations from Transformers

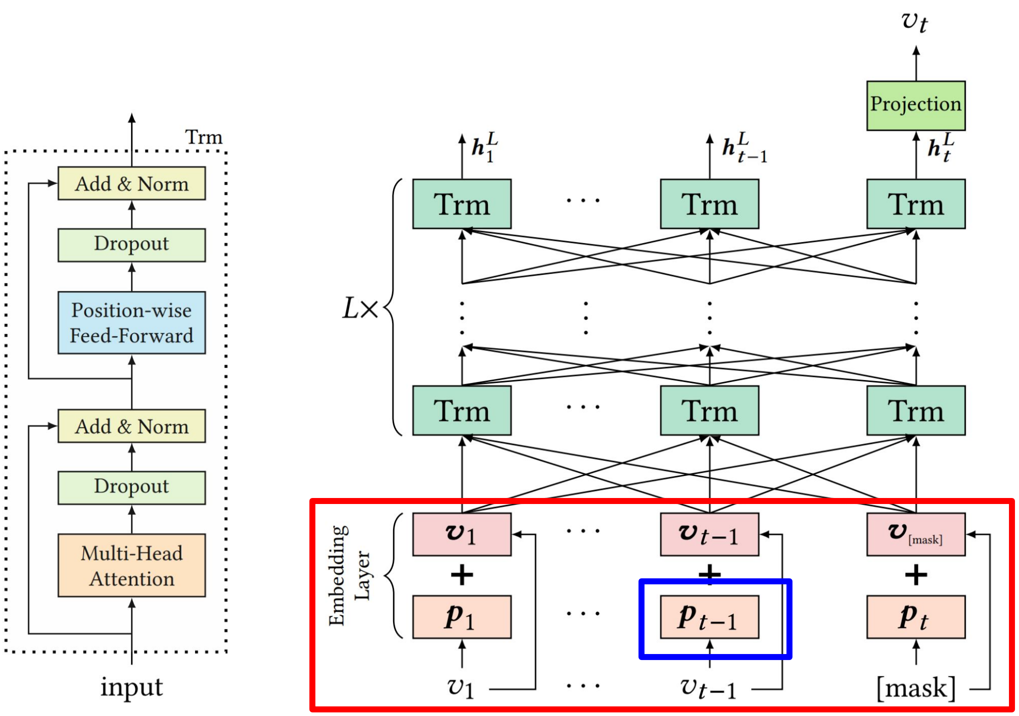


Bidirectional Encoder Representations from Transformers

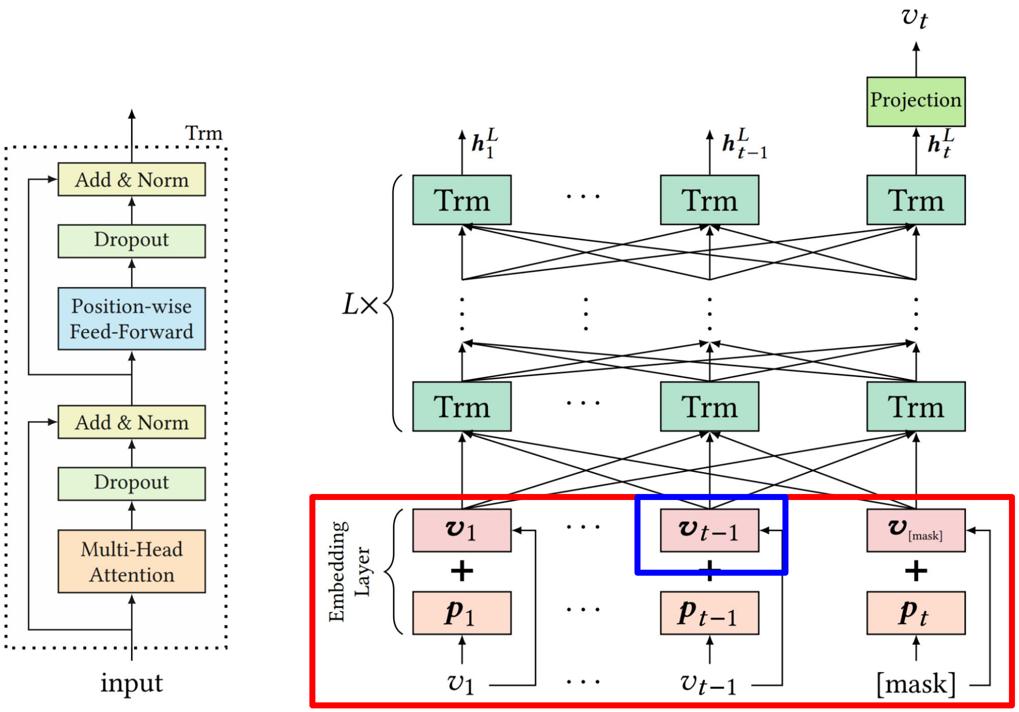




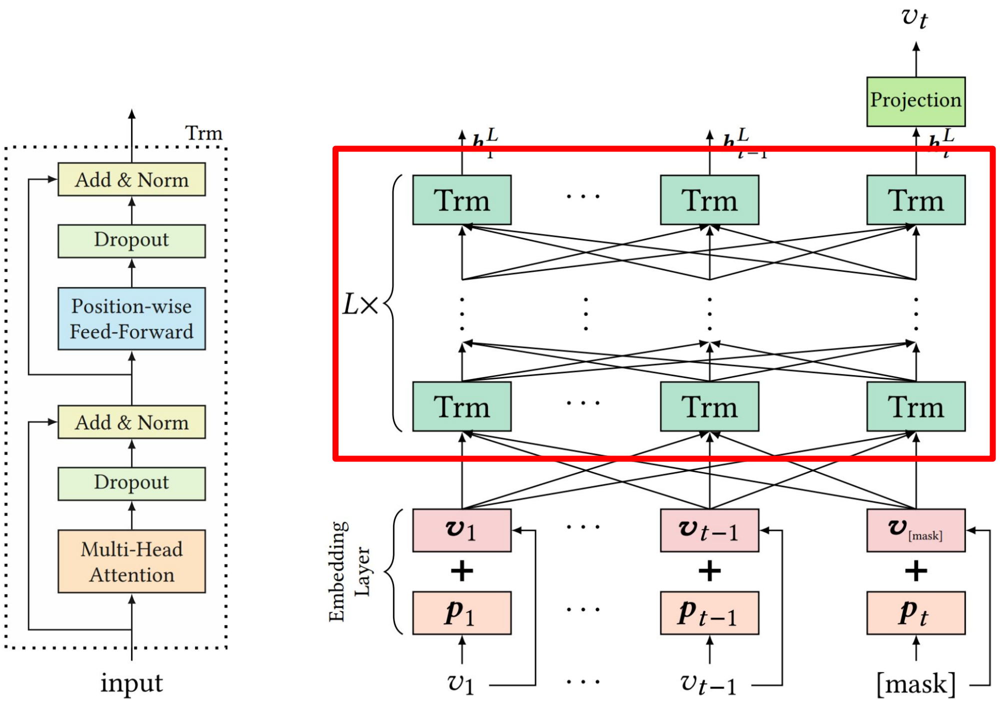
Bidirectional Encoder Representations from Transformers



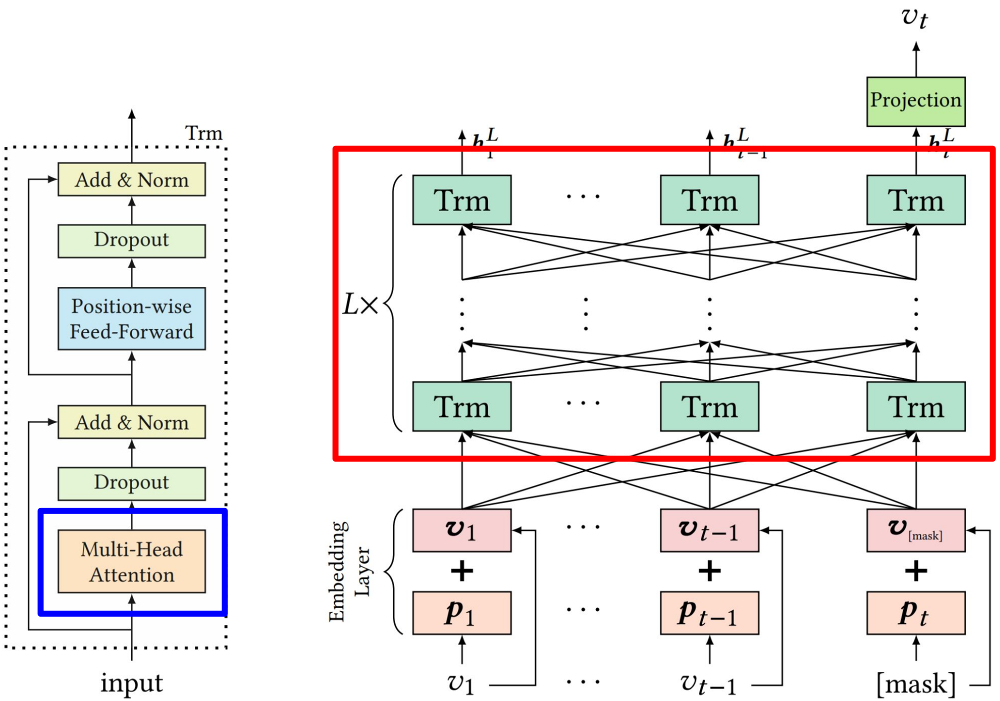
Bidirectional Encoder Representations from Transformers



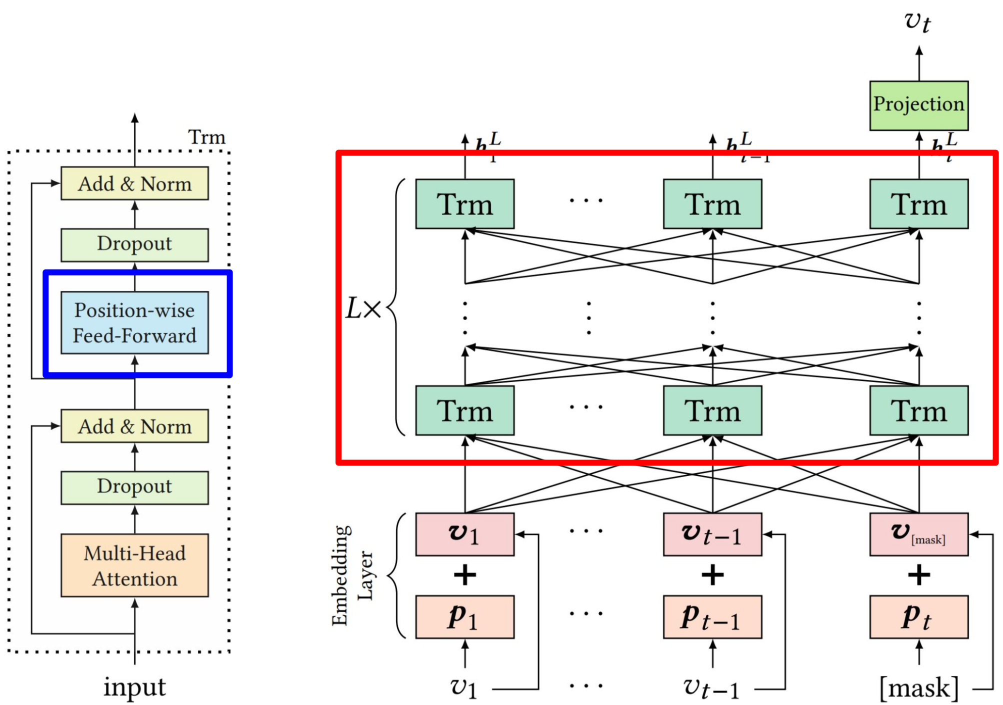
Bidirectional Encoder Representations from Transformers



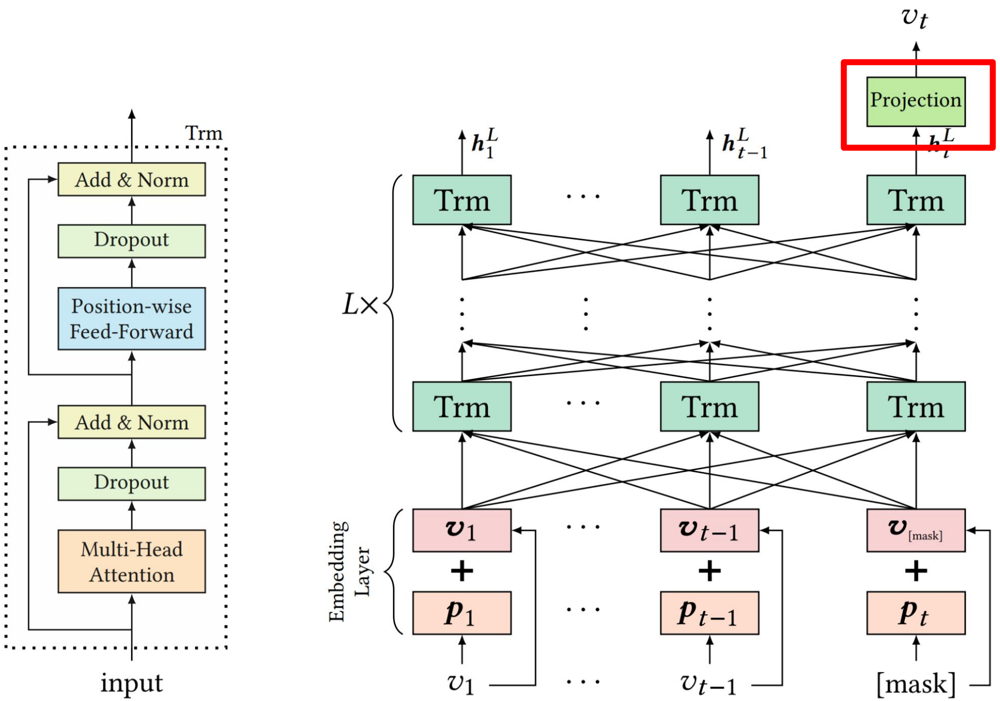
Bidirectional Encoder Representations from Transformers



Bidirectional Encoder Representations from Transformers



Bidirectional Encoder Representations from Transformers



Cloze Task (Masked Language Model)

Cloze Task

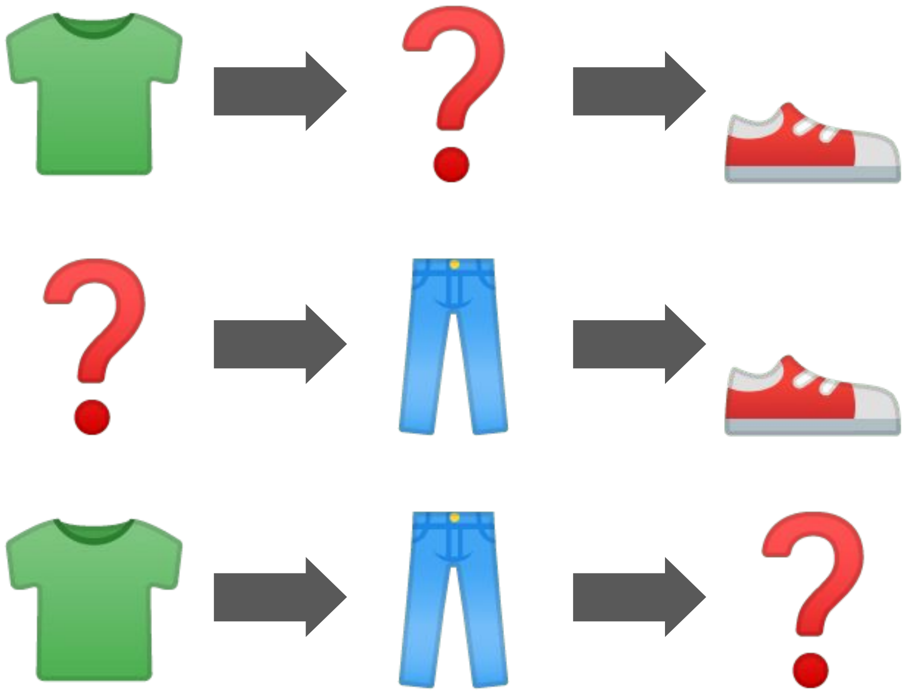
I have bought a Big-Mac menu, containing a [mask], some french [mask] and a soft [mask].

Cloze Task

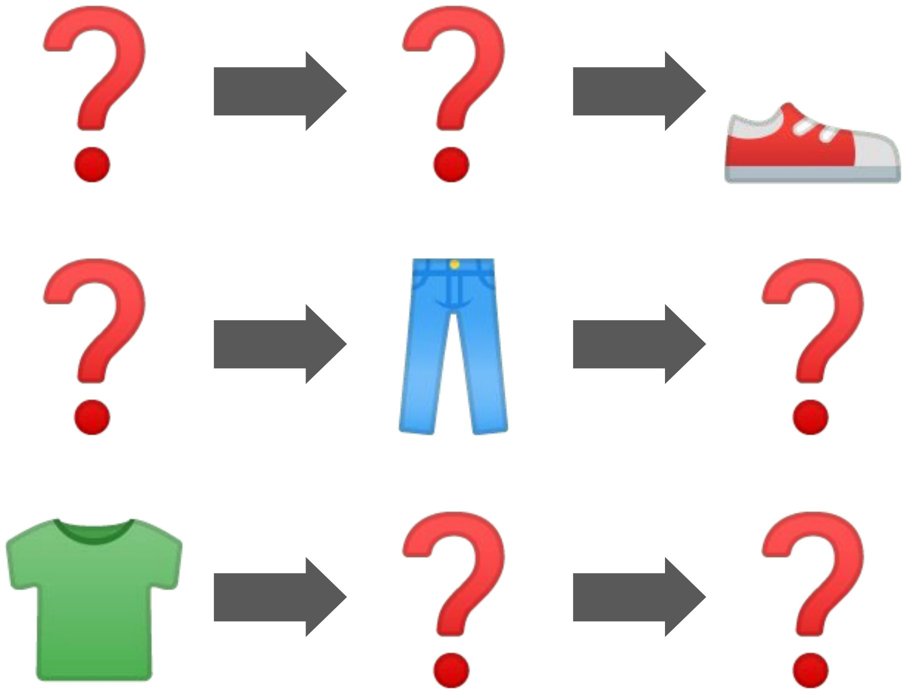
I have bought a Big-Mac menu, containing a Big-Mac, some french fries and a soft drink.

Cloze Task

Cloze Task: Training



Cloze Task: Training

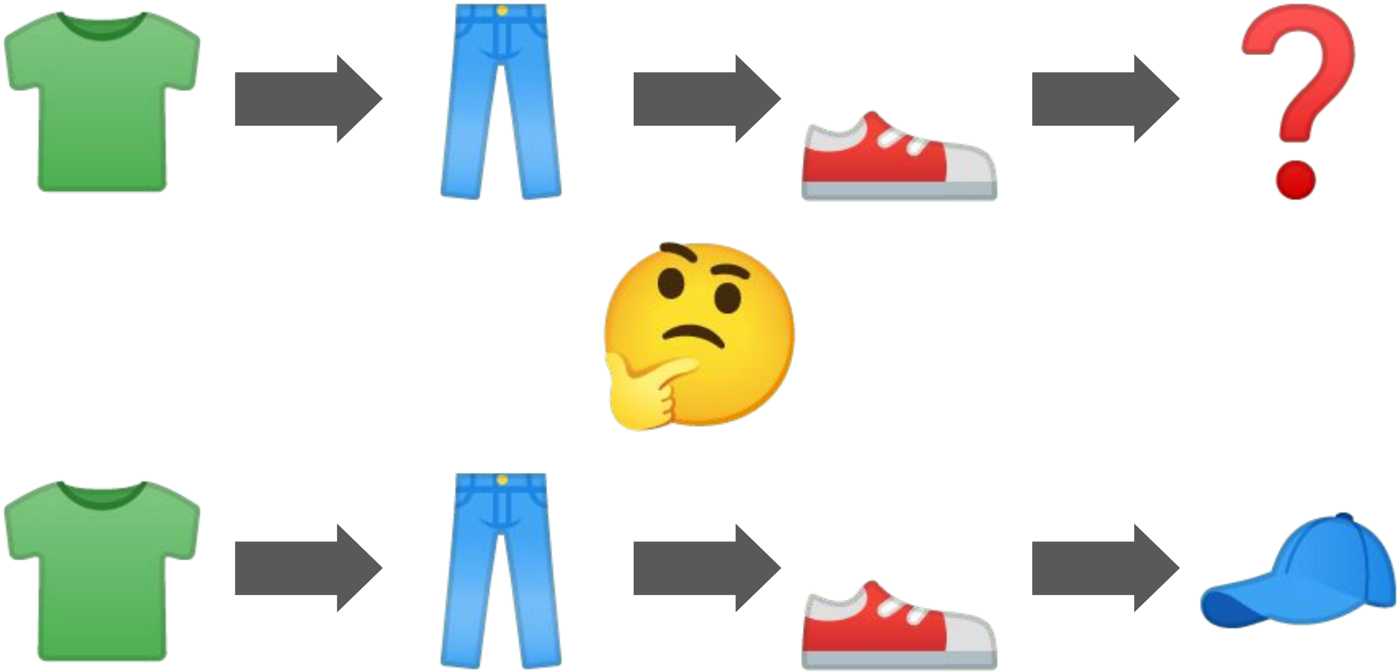


Cloze Task: Training

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Cloze Task: Inference

Cloze Task: Inference



Loss Function

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

Loss Function

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

Loss Function

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

Loss Function

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

Experiments

Baselines

Matrix Factorization Based:

- POP
- BPR-MF
- NCF
- FPMC

RNN or CNN Based:

- GRU4Rec
- GRU4Rec^{+}
- Caser

Transformer Based:

- SASRec (Previous State of the Art)

Baselines

Matrix Factorization Based:

- POP
- BPR-MF
- NCF
- FPMC

RNN or CNN Based:

- GRU4Rec
- GRU4Rec^{+}
- Caser

Transformer Based:

- SASRec (Previous State of the Art)

Datasets ($d = 32$, $L = 2$, $h = 2$)

Amazon Beauty: dataset crawled from **Amazon** containing users reviews in the **Beauty** category. ($\rho = 0.6$, $N = 50$)

Steam: dataset collected from **Steam**, which is an online video game distribution platform. ($\rho = 0.4$, $N = 50$)

MovieLens: a dataset for movie recommendation (**ML-1m** and **ML-20m** are used for the experiments). ($\rho = 0.2$, $N = 200$)

Table 1: Statistics of 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%

Metrics: HR@k

$$HR@k = \frac{1}{N}\sum_{i}^{N} in\_top\_k$$

Prediction:

- 👟 (50%)
- 🧢 (30%)
- 👕 (15%)
- 👖 (5%)

Ground Truth: 👕

With k = (1, 2, 3): **in_top_k** is **True**

With k = 4: **in_top_k** is **False**

Metrics: NDCG@k

$$NDCG@k = \frac{DCG@k}{IDCG}$$

$$DCG@k = \sum_{i=1}^{k} \frac{G_i}{\log_2 (i+1)}$$

$$IDCG = \frac{1}{\log_2 2} = 1$$

Prediction:

- 👟 (50%)
- 🧢 (30%)
- 👕 (15%)
- 👖 (5%)

Ground Truth: 👕

DCG@1 = 0.5

DCG@2 = 0.5 + 0.3 / log(i + 2)

...

Metrics: NDCG@k

$$\boxed{NDCG@k = \frac{DCG@k}{IDCG}}$$

$$DCG@k = \sum_{i=1}^{k} \frac{G_i}{\log_2 (i+1)}$$

$$IDCG = \frac{1}{\log_2 2} = 1$$

Prediction:

- 👟 (50%)
- 🧢 (30%)
- 👕 (15%)
- 👖 (5%)

Ground Truth: 👕

DCG@1 = 0.5

DCG@2 = 0.5 + 0.3 / log(i + 2)

...

Metrics: NDCG@k

$$NDCG@k = \frac{DCG@k}{IDCG}$$

$$\boxed{DCG@k = \sum_{i=1}^{k} \frac{G_i}{\log_2(i+1)}}$$

$$IDCG = \frac{1}{\log_2 2} = 1$$

Prediction:

- 👟 (50%)
- 🧢 (30%)
- 👕 (15%)
- 👖 (5%)

Ground Truth: 👕

DCG@1 = 0.5

DCG@2 = 0.5 + 0.3 / log(i + 2)

...

Metrics: NDCG@k

$$NDCG@k = \frac{DCG@k}{IDCG}$$

$$DCG@k = \sum_{i=1}^{k} \frac{G_i}{\log_2 (i+1)}$$

$$\boxed{IDCG = \frac{1}{\log_2 2} = 1}$$

Prediction:

- 👟 (50%)
- 🧢 (30%)
- 👕 (15%)
- 👖 (5%)

Ground Truth: 👕

DCG@1 = 0.5

DCG@2 = 0.5 + 0.3 / log(i + 2)

...

Metrics: MRR

$$MRR = \frac{1}{N}\sum_{i}^{N}\frac{1}{rank_i}$$

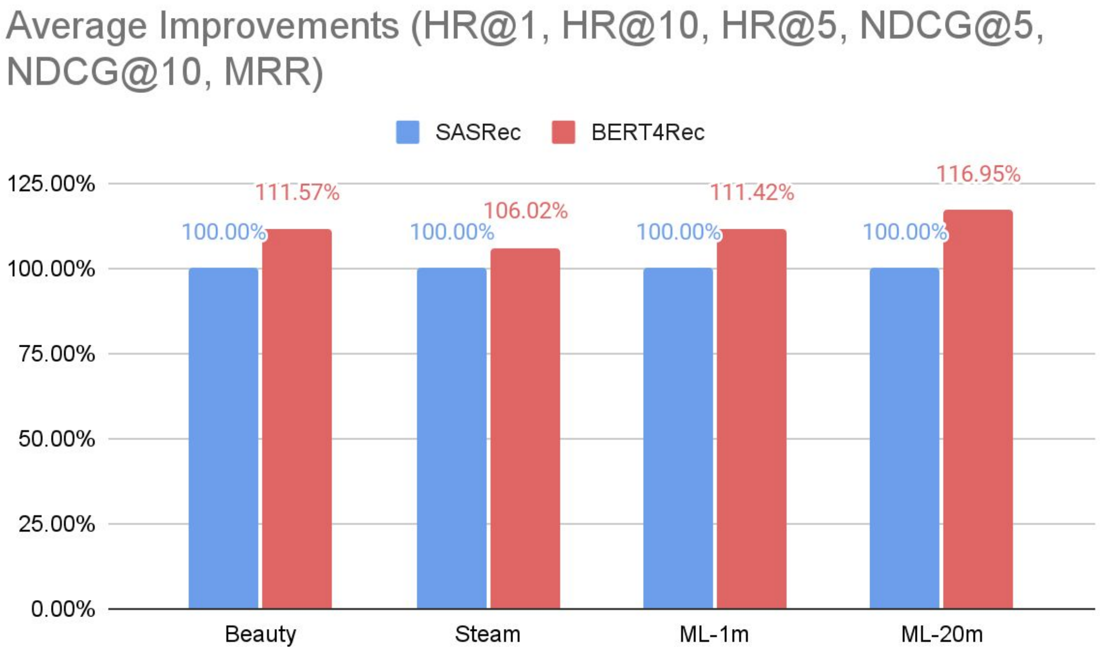
Prediction:

- 👟 (50%)
- 🧢 (30%)
- 👕 (15%)
- 👖 (5%)

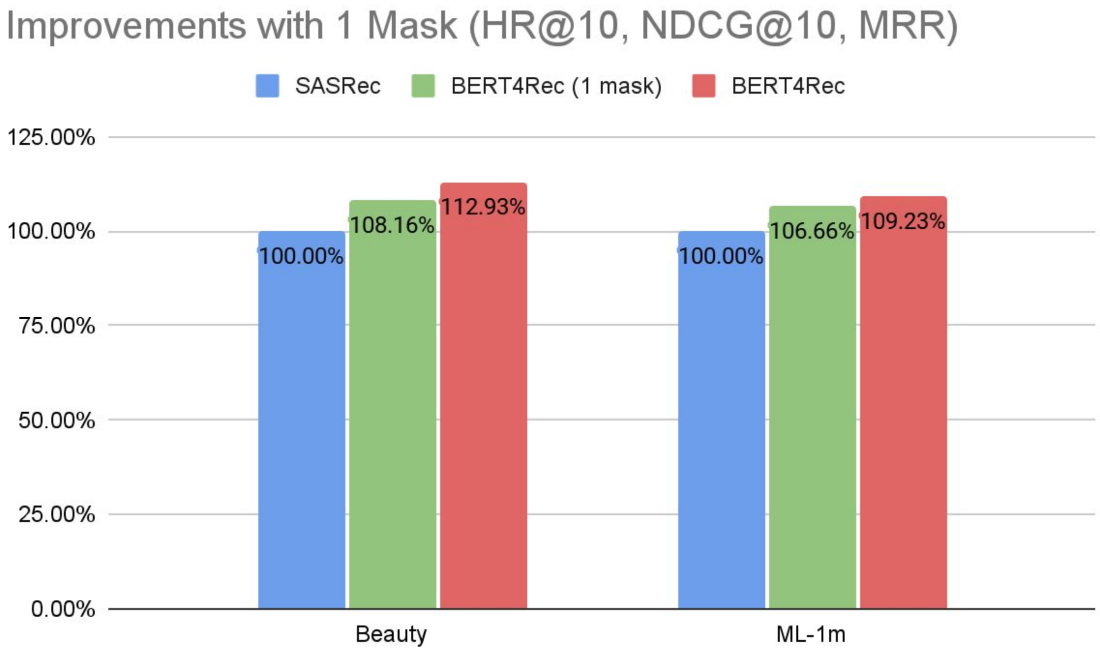
Ground Truth: 👕

rank_i = 3

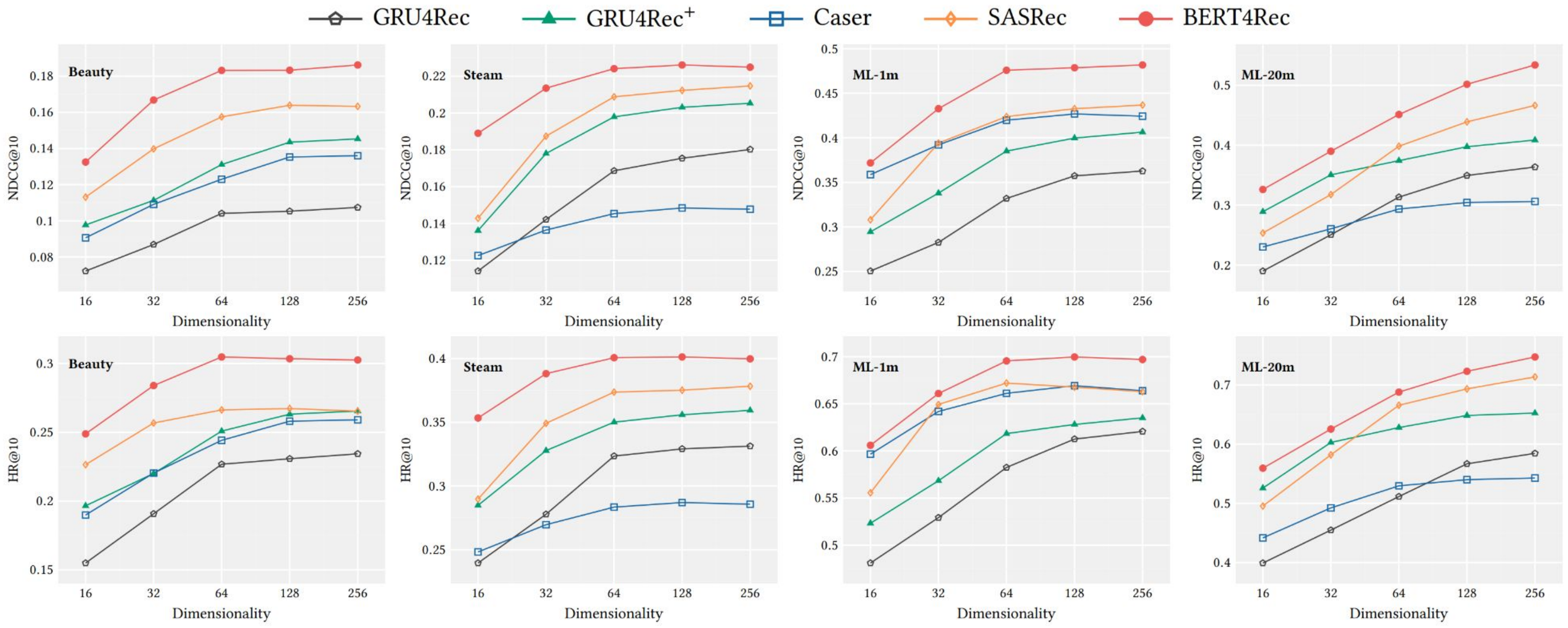
Improvements



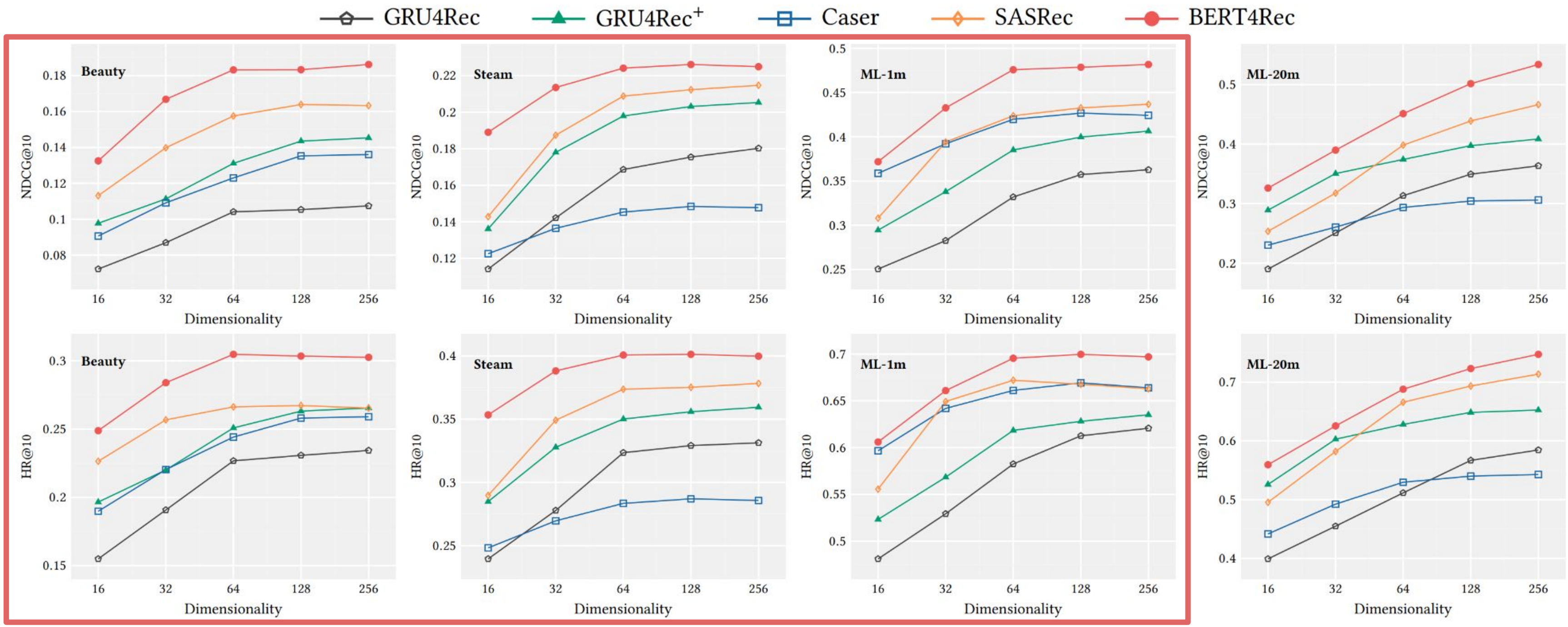
Cloze Task or Bidirectional Encoding?



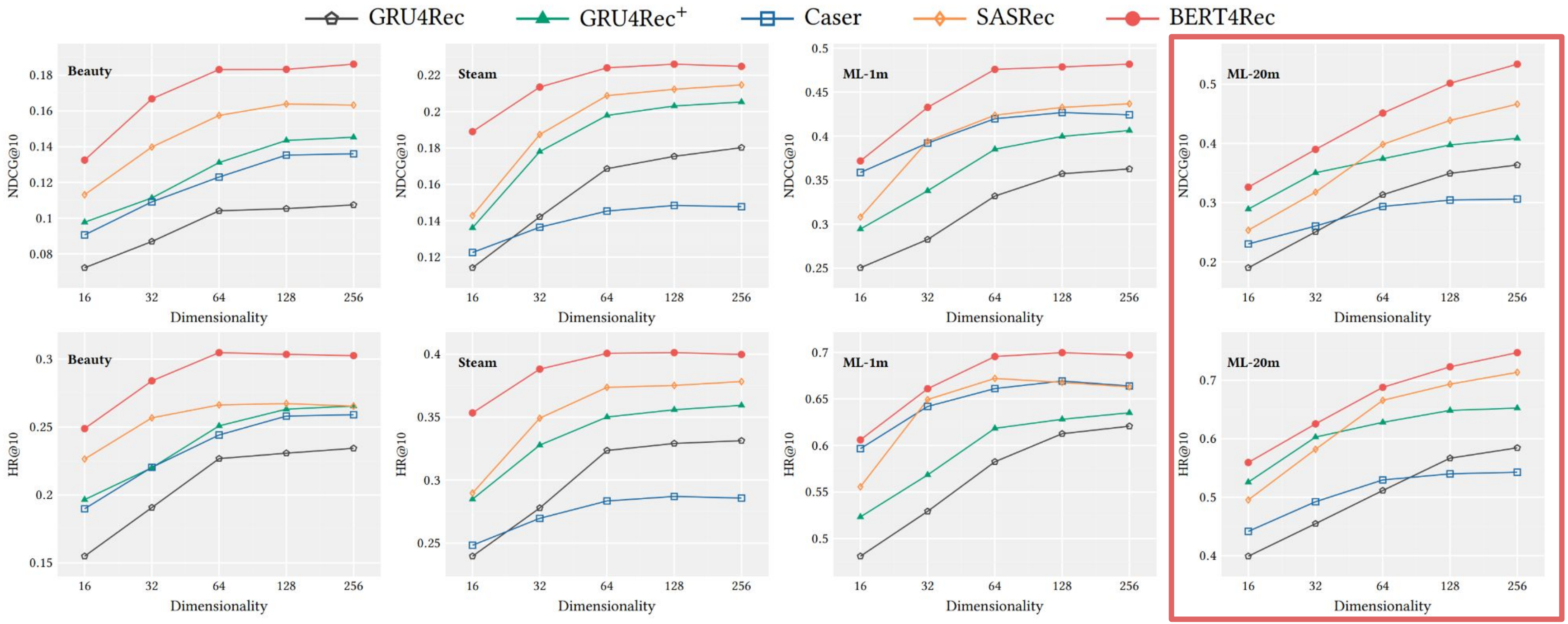
Influence of Hidden Dimensionality *d*



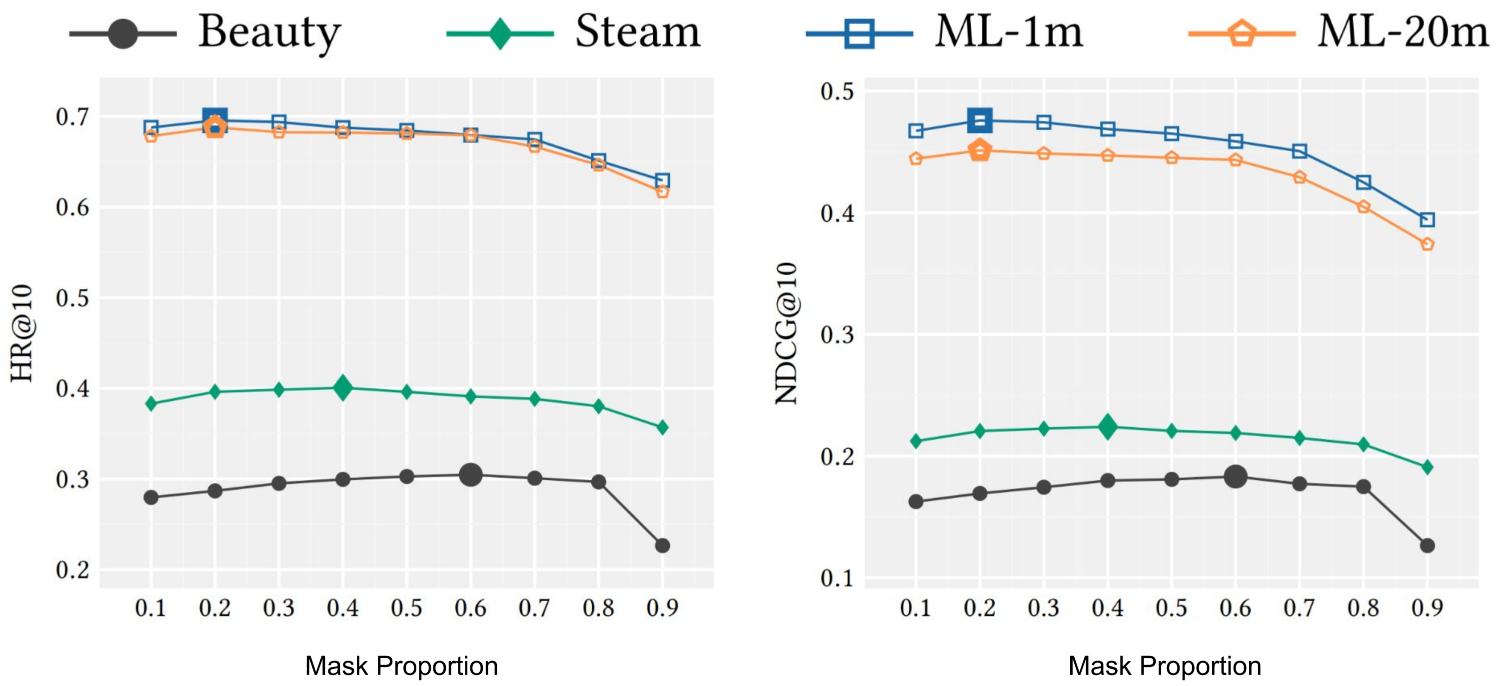
Influence of Hidden Dimensionality *d*



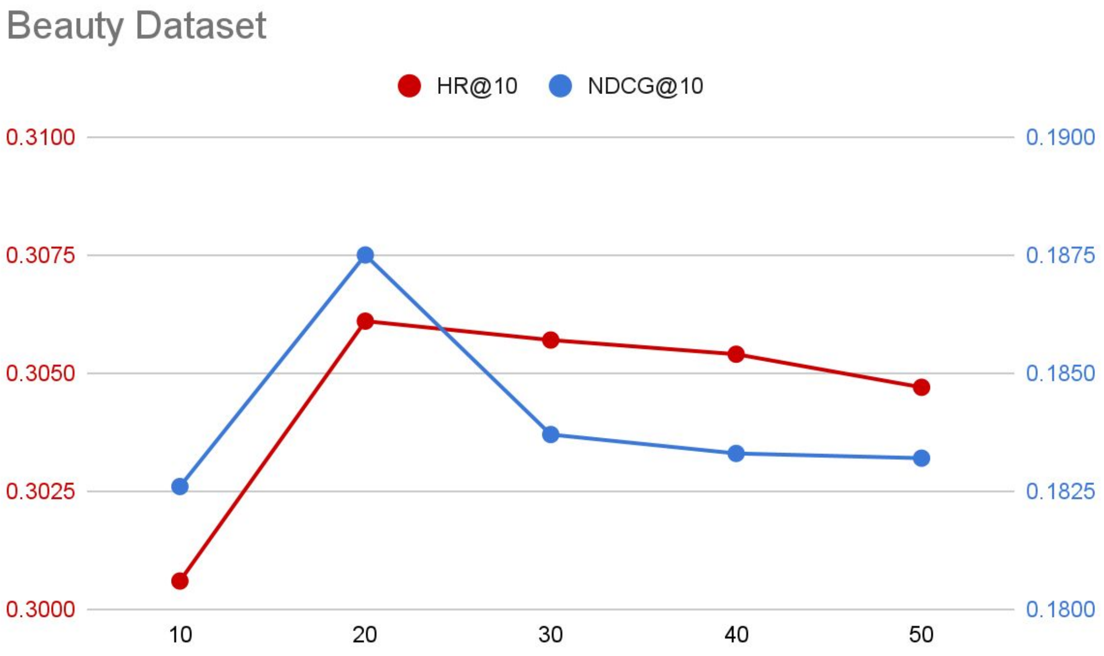
Influence of Hidden Dimensionality *d*



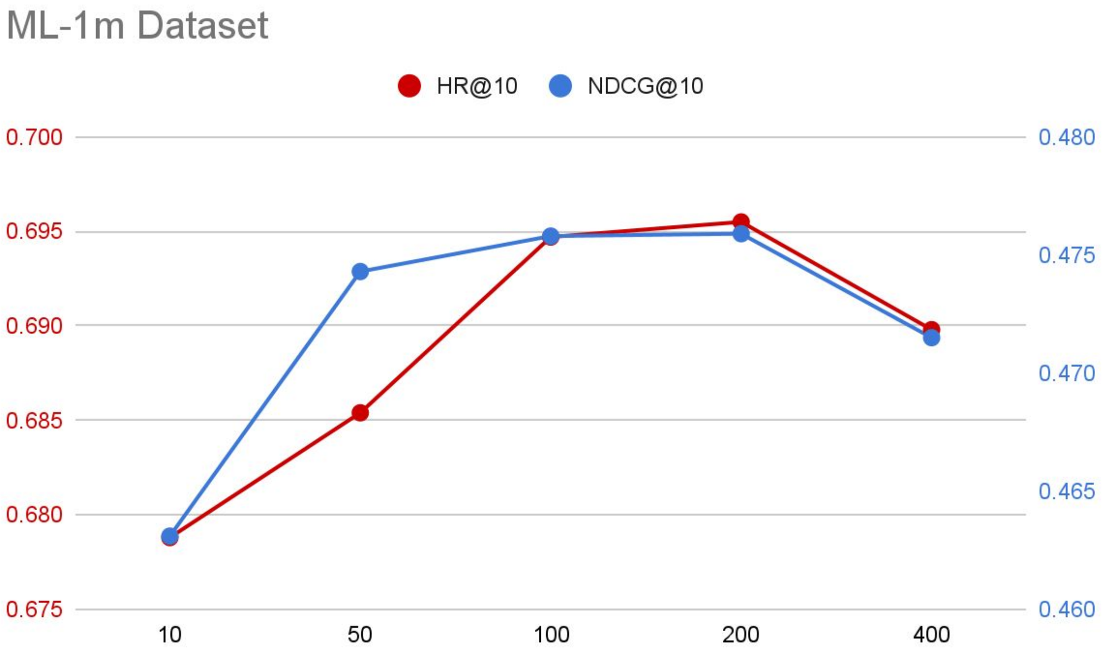
Impact of Mask Proportion ρ



Impact of Maximum Sequence Length ***N***



Impact of Maximum Sequence Length ***N***



Ablation Study (NDCG@10)

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**

Ablation Study (NDCG@10)

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**

Ablation Study (NDCG@10)

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**

Ablation Study (NDCG@10)

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**

Ablation Study (NDCG@10)

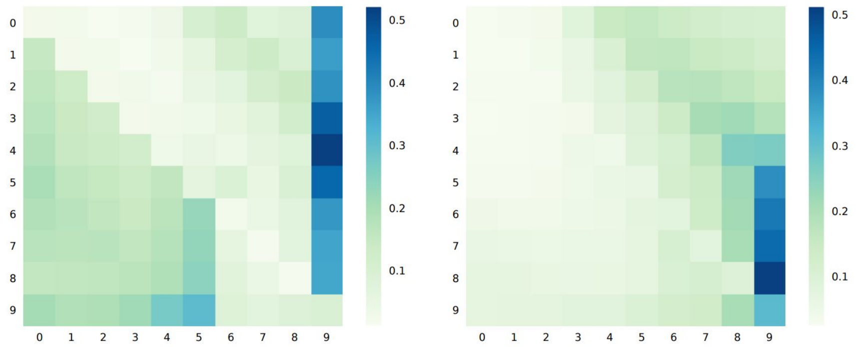
Architecture	Dataset			
	Beauty	Steam	ML-1m	ML-20m
$L = 2, h = 2$	0.1832	0.2241	0.4759	0.4513
w/o PE	0.1741	0.2060	0.2155↓	0.2867↓
w/o PFFN	0.1803	0.2137	0.4544	0.4296
w/o LN	0.1642↓	0.2058	0.4334	0.4186
w/o RC	0.1619↓	0.2193	0.4643	0.4483
w/o Dropout	0.1658	0.2185	0.4553	0.4471
1 layer ($L = 1$)	0.1782	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**

Ablation Study (NDCG@10)

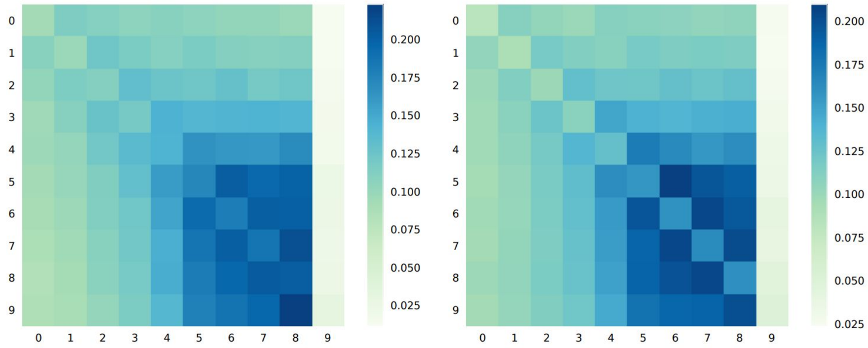
Architecture	Dataset			
	Beauty	Steam	ML-1m	ML-20m
$L = 2, h = 2$	0.1832	0.2241	0.4759	0.4513
w/o PE	0.1741	0.2060	0.2155↓	0.2867↓
w/o PFFN	0.1803	0.2137	0.4544	0.4296
w/o LN	0.1642↓	0.2058	0.4334	0.4186
w/o RC	0.1619↓	0.2193	0.4643	0.4483
w/o Dropout	0.1658	0.2185	0.4553	0.4471
1 layer ($L = 1$)	0.1782	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**

Why Bidirectional Encoding?

Why Bidirectional Encoding?

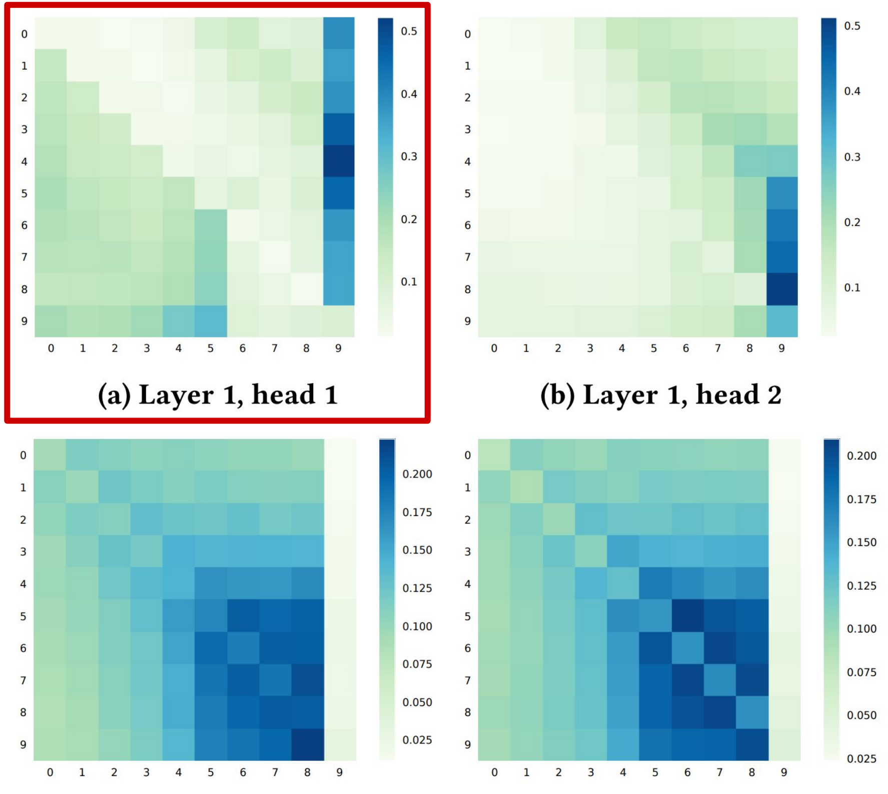


(a) Layer 1, head 1 (b) Layer 1, head 2



(c) Layer 2, head 2 (d) Layer 2, head 4

Why Bidirectional Encoding?



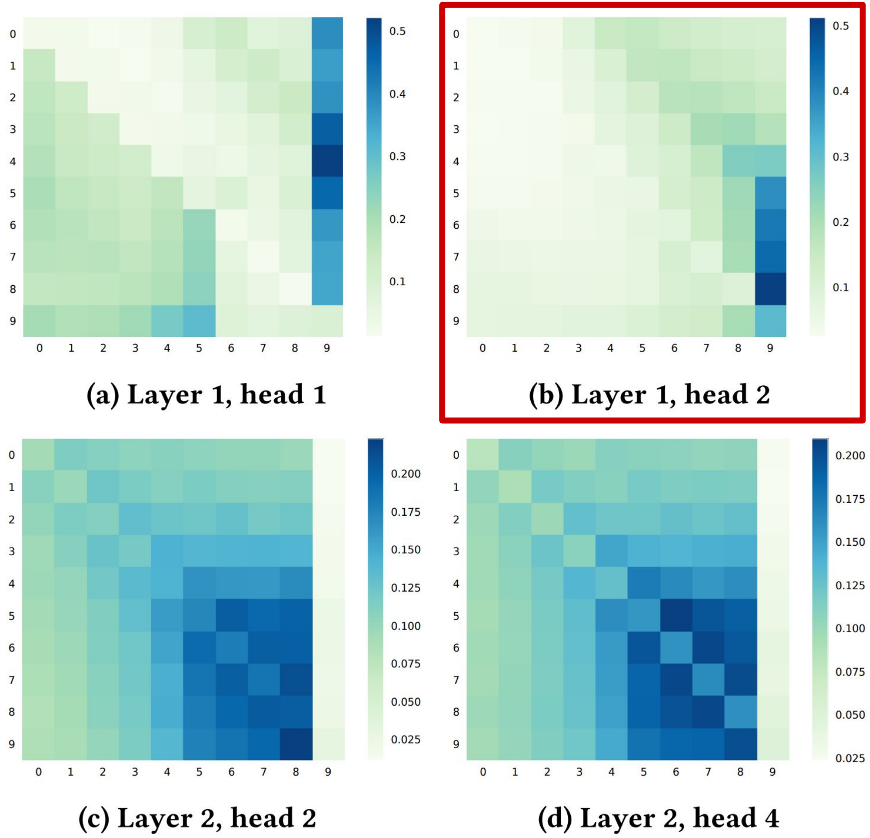
(a) Layer 1, head 1

(b) Layer 1, head 2

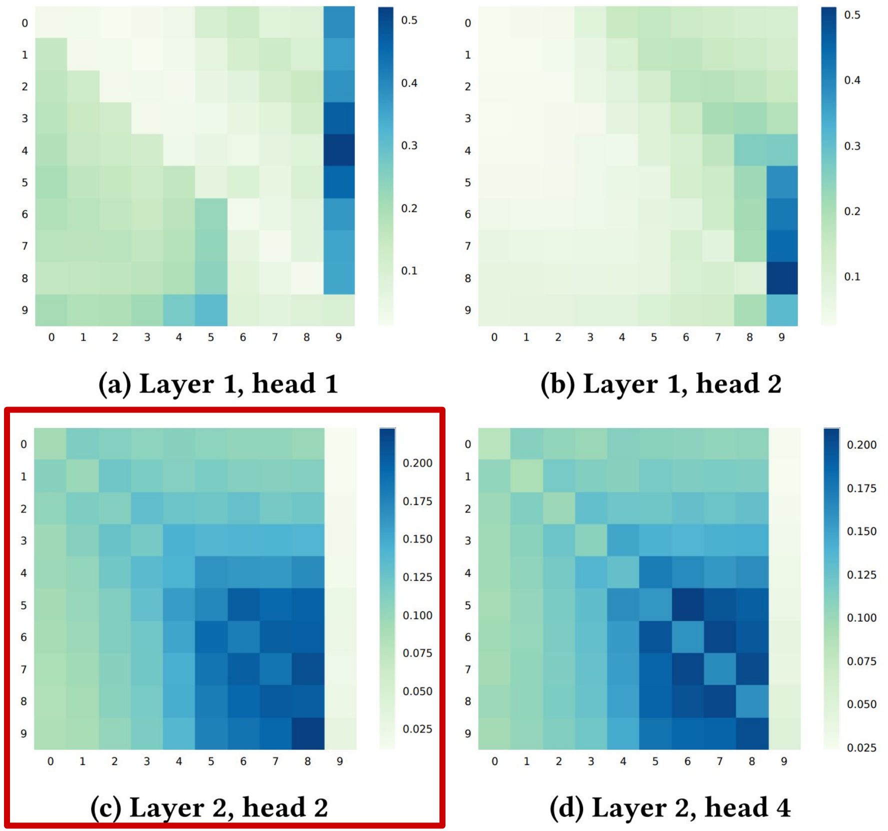
(c) Layer 2, head 2

(d) Layer 2, head 4

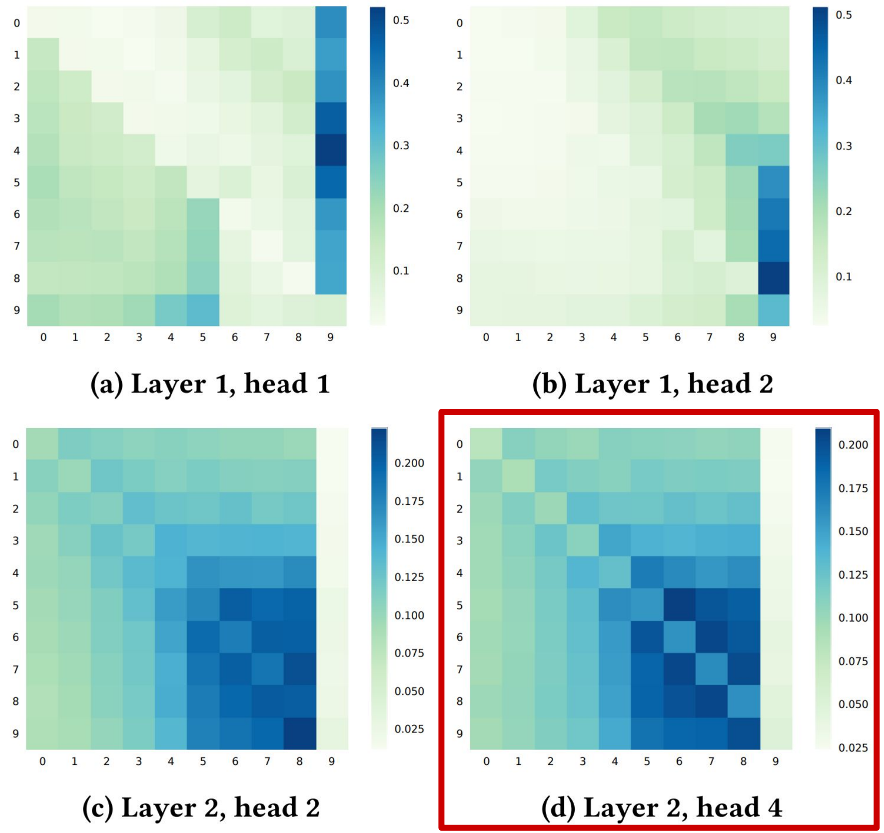
Why Bidirectional Encoding?



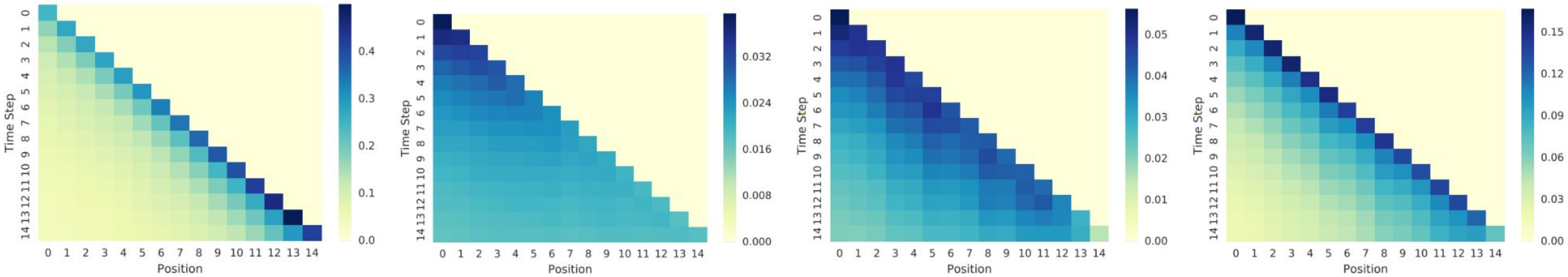
Why Bidirectional Encoding?



Why Bidirectional Encoding?

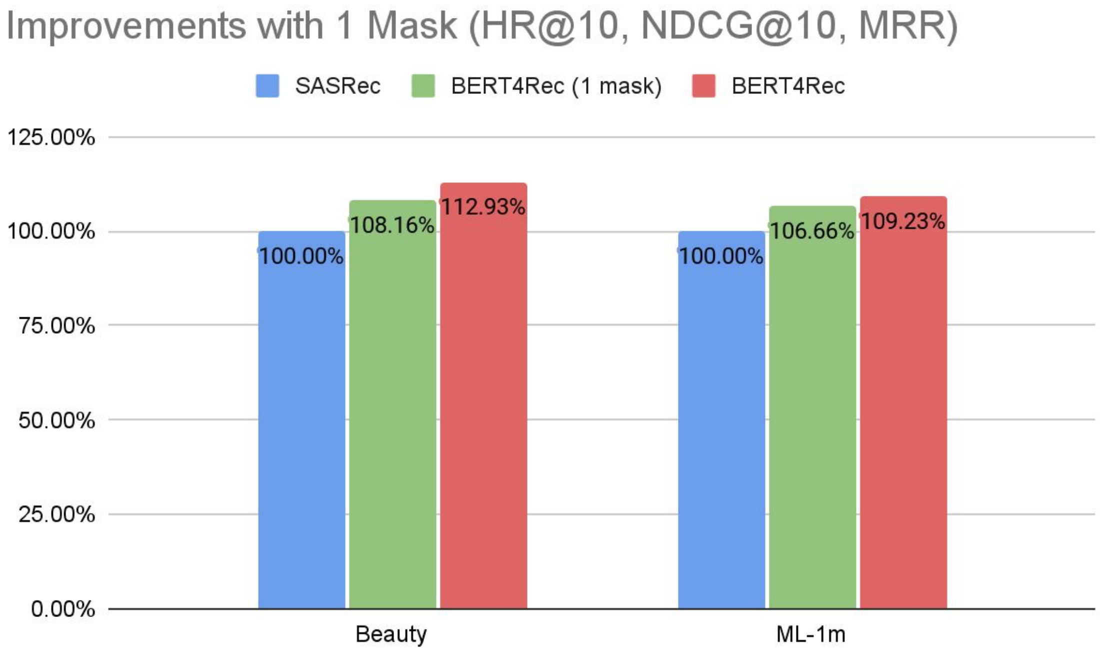


Why Bidirectional Encoding?



(a) *Beauty*, Layer 1 (b) *ML-1M*, Layer 1, w/o PE (c) *ML-1M*, Layer 1 (d) *ML-1M*, Layer 2

Isolation from Cloze Task



Thanks for being here

References

- BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer - Fei Sun
- Self-Attentive Sequential Recommendation - Wang-Cheng Kang