Pontificia Universidad Católica de Chile

Denoising Self-Attentive Sequential Recommendation

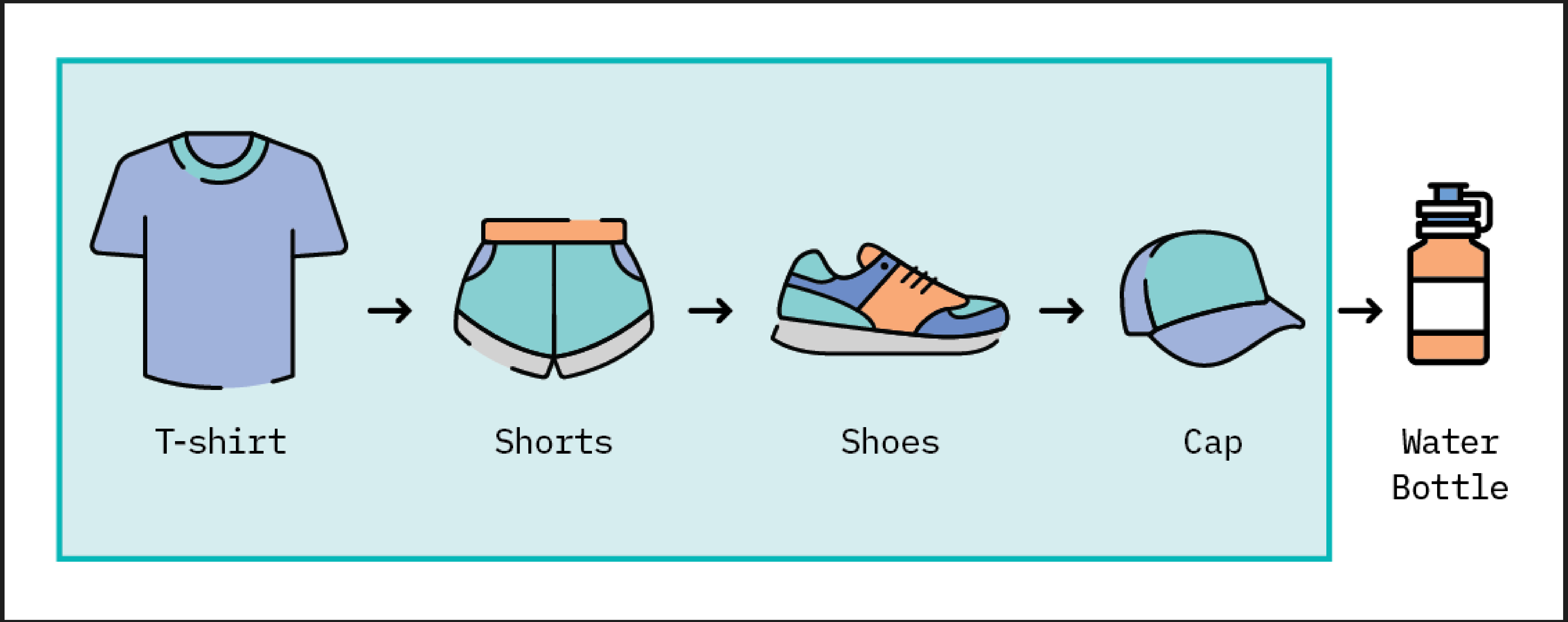
HUIYUAN CHEN et al.

Maureen Cooper, Lucía De Pineda y Anna Ramon

01 INTRODUCTION

CONTEXT

SEQUENTIAL RECOMMENDATION



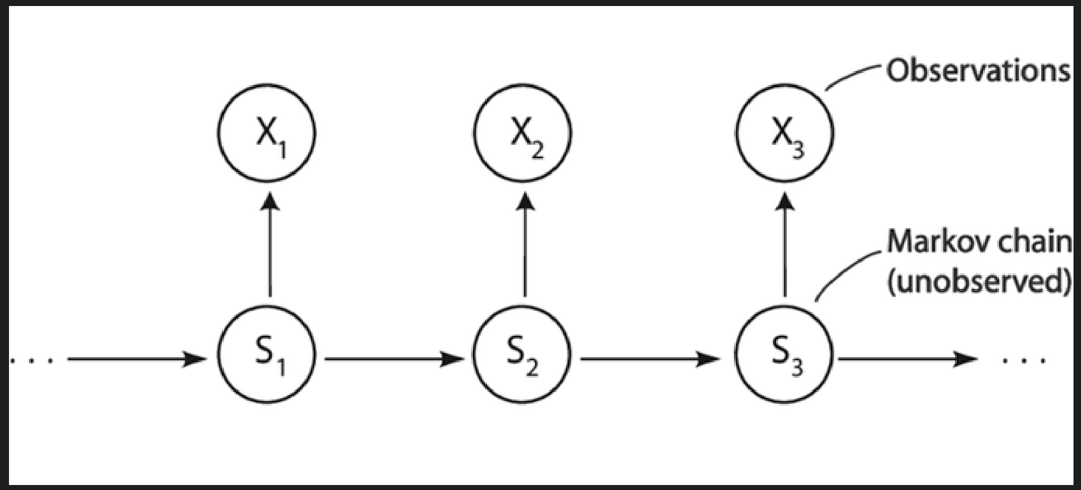
Historical Profile

Predicted recommendation

EXISTING MODELS

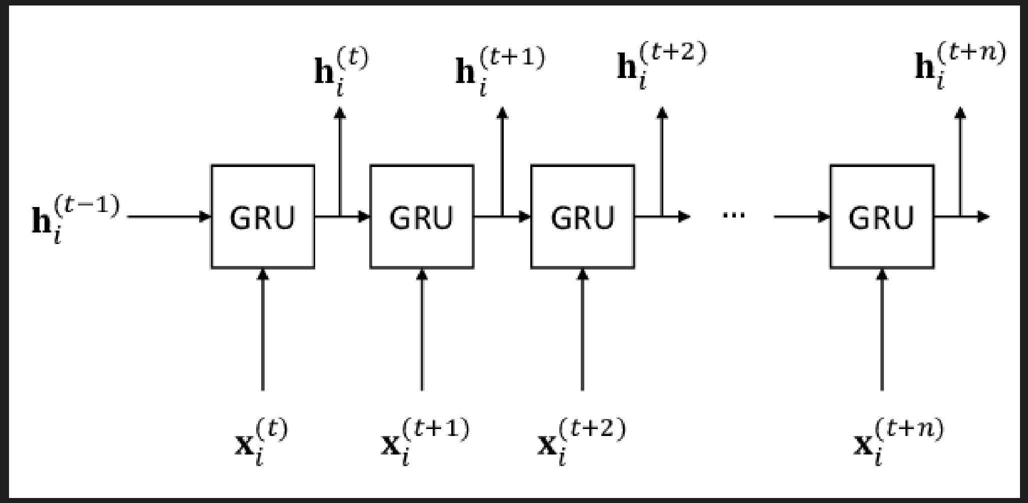
Capturation of user's **historical** actions:

Markov Chain Models



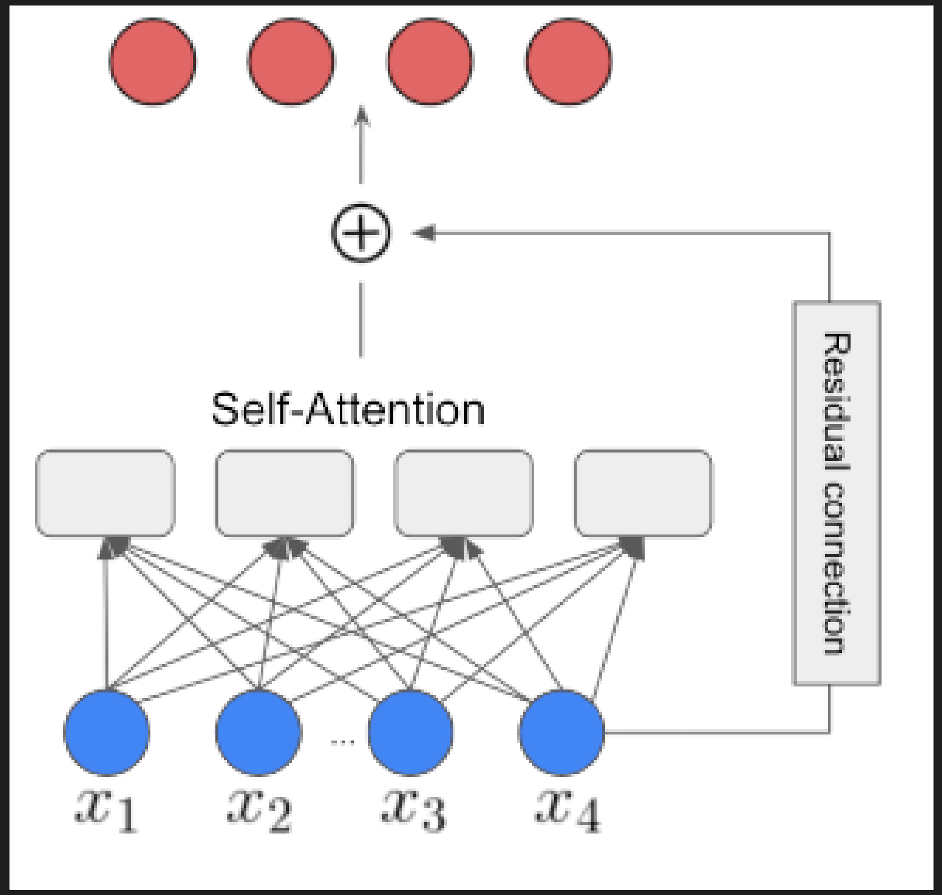
Short-Term memory

RNN

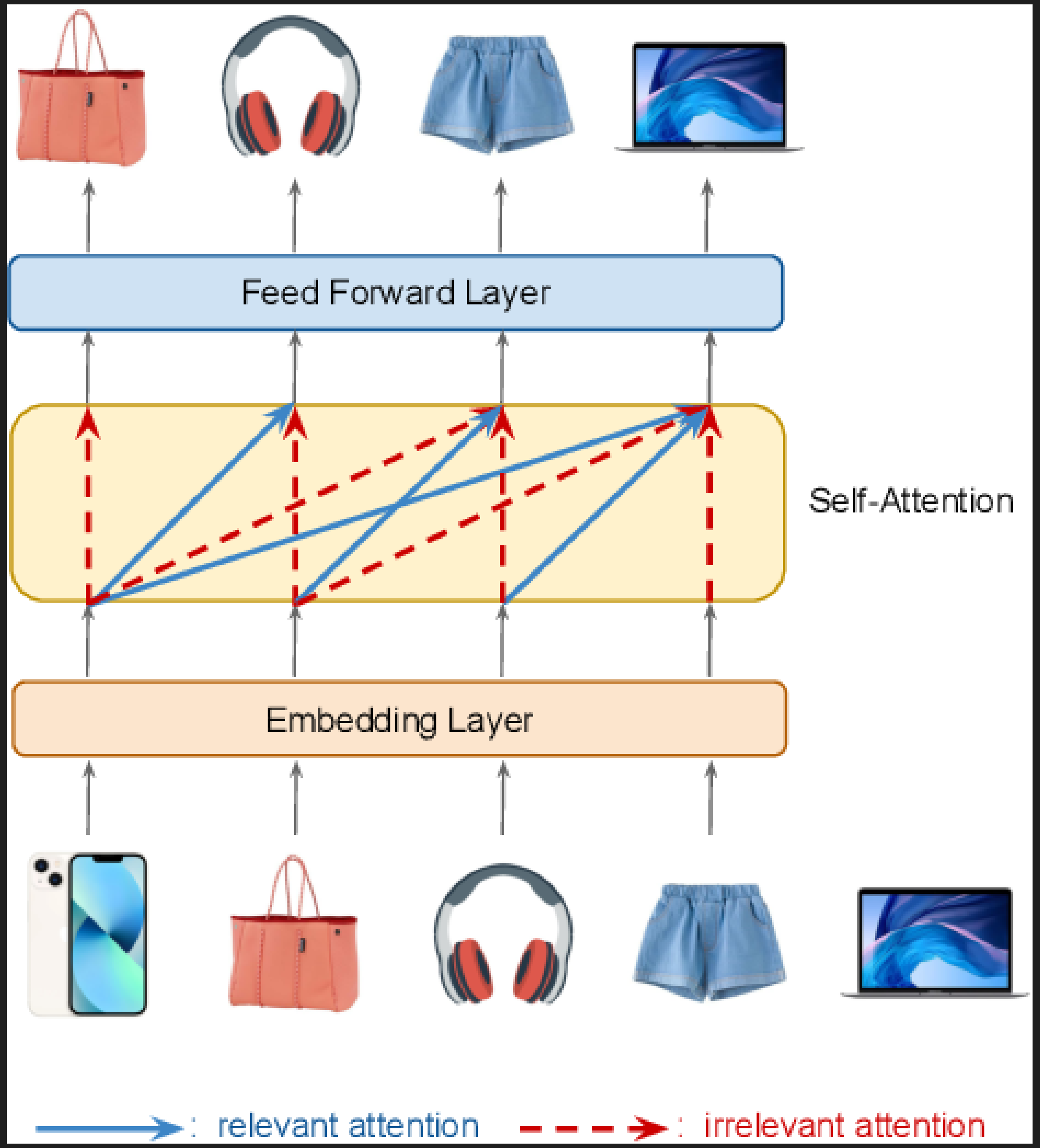


High cost to train

Transformers

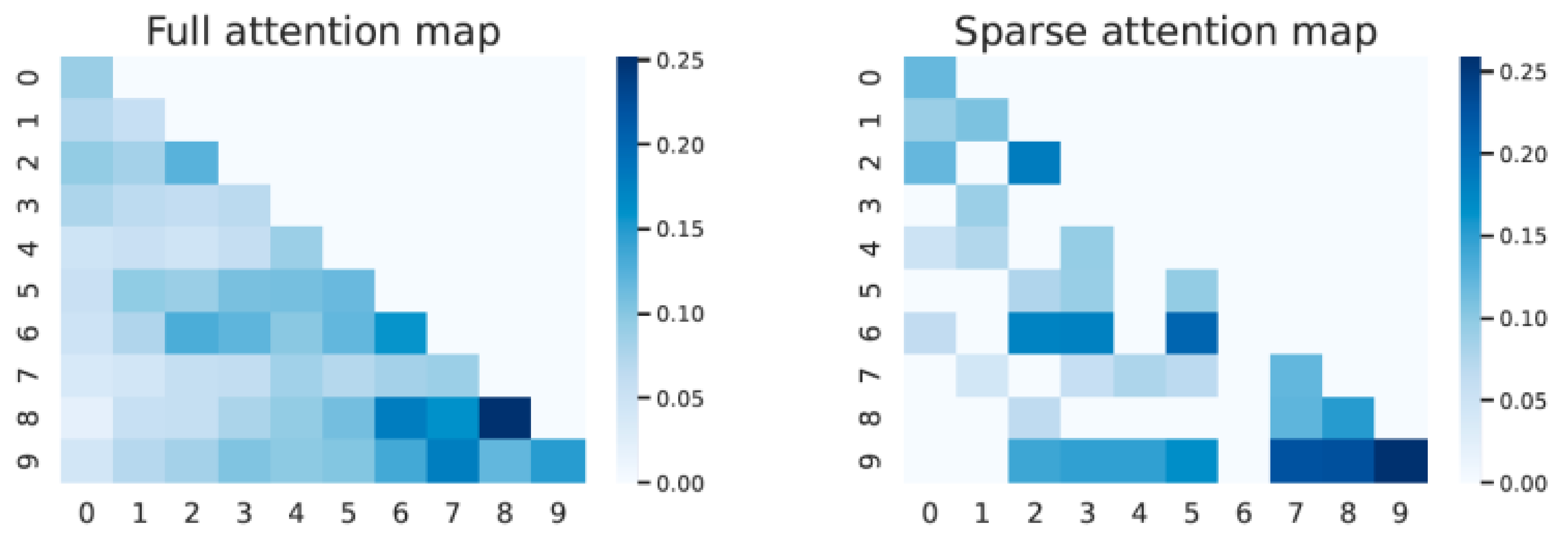


CHALLENGE How to detect irrelevant items?



CONTRIBUTIONS

1. Denoising item sequences
2. Differentiable Masks: Prune irrelevant information



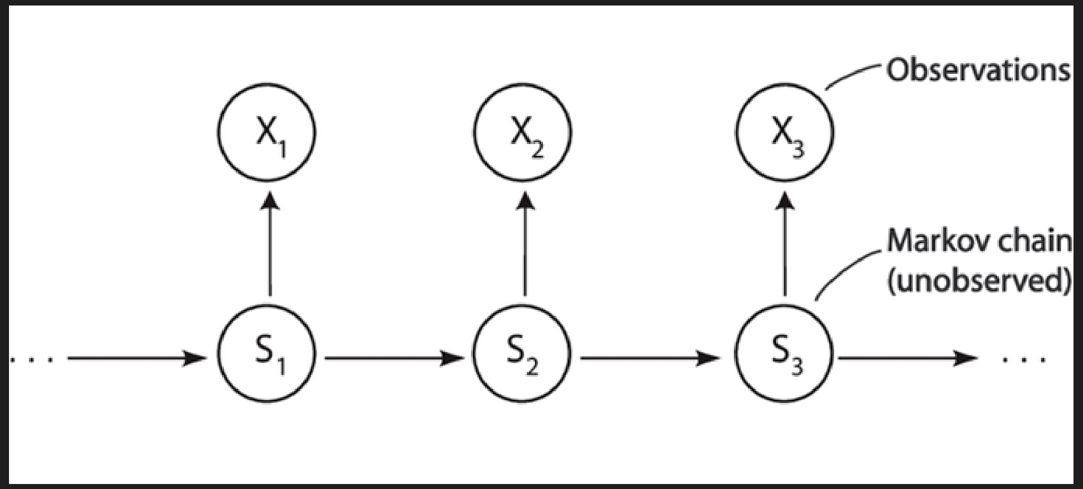
(a) SASRec (b) SASRec+Denoiser

02 RELATED WORK

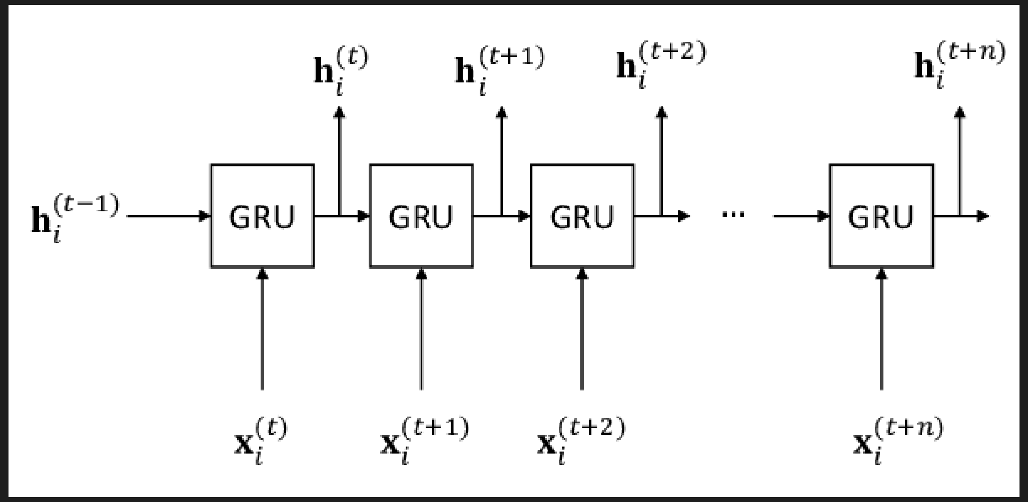
SEQUENTIAL RECOMMENDATION

Leveraging **sequences** of **user-item interactions**

Markov Chain Models

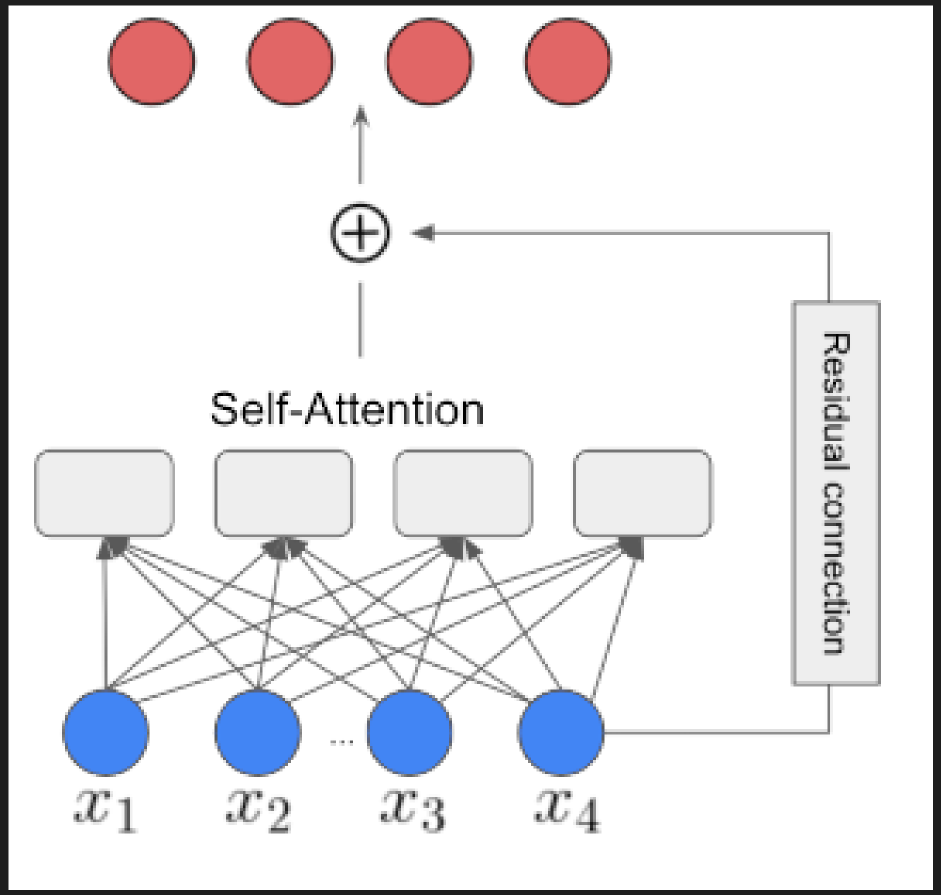


Deep Neural Networks



- **GRU4REC**
- **Caser**
- **MANN**
- **SR-GNN**

Transformers

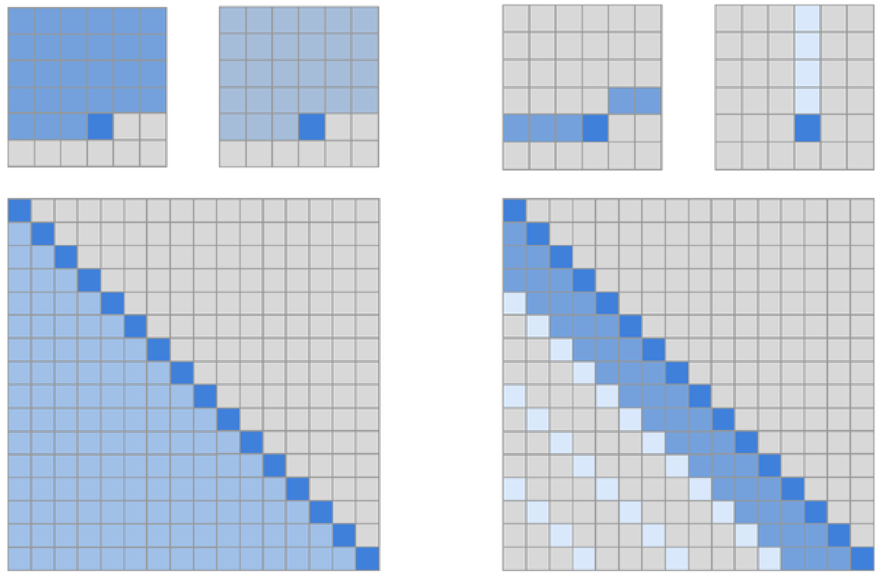


- **SASRec**

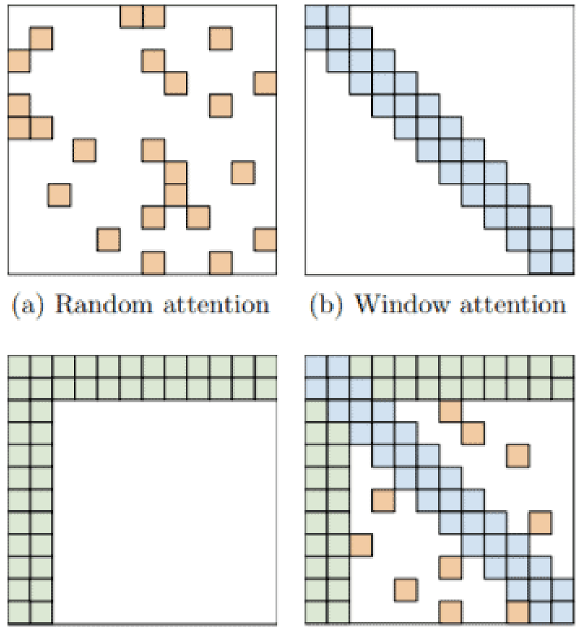
SPARSE TRANSFORMER

Seek to achieve sparse attention maps

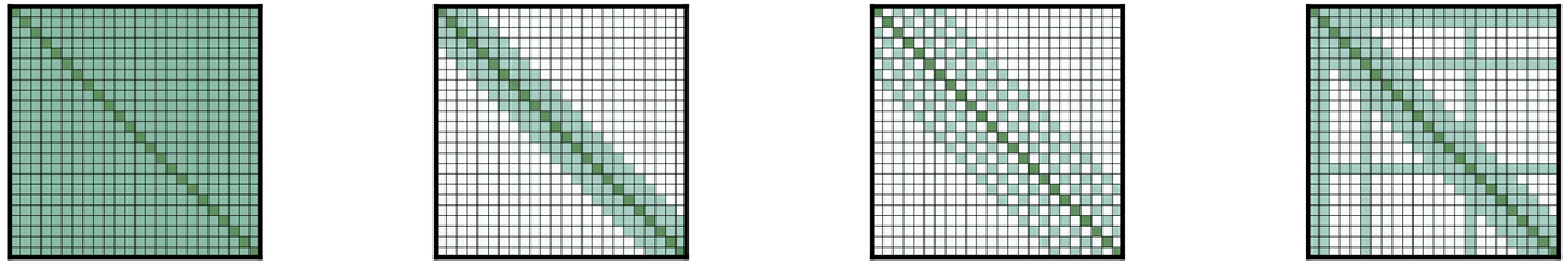
State-of-the-art performance



(a) Transformer (b) Sparse Transformer (strided)



(a) Random attention (b) Window attention (c) Global Attention (d) BIGBIRD



(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

- Reformer
- Star Transformer
- Sparse Transformer
- Longformer
- BigBird

03 DATA

DATASETS

5 benchmark datasets:

MovieLens, Amazon (Beauty, Games, Movies&TV), Steam

Dataset	#Users	#Items	Avg actions/user	#Actions
MovieLens	6,040	3,416	163.5	0.987M
Beauty	51,369	19,369	4.39	0.225M
Games	30,935	12,111	6.46	0.2M
Movies&TV	40,928	37,564	25.55	1.05M
Steam	114,796	8,648	7.58	0.87M

04 MODEL

REC-DENOISER

DIFFERENTIABLE MASKS

- Learnable Sparse attentions

$$\mathbf{A}^{(l)} = \operatorname{softmax}\left(\frac{\mathbf{Q}^{(l)}\mathbf{K}^{(l)T}}{\sqrt{d}}\right),$$

$$\mathbf{M}^{(l)} = \mathbf{A}^{(l)} \odot \mathbf{Z}^{(l)} \longrightarrow \mathbf{Z}^{(l)}_{u,v} \sim \operatorname{Bern}(\Pi^{(l)}_{u,v})$$

$$\operatorname{Attention}(\mathbf{Q}^{(l)}, \mathbf{K}^{(l)}, \mathbf{V}^{(l)}) = \mathbf{M}^{(l)}\mathbf{V}^{(l)},$$

$$\mathcal{R}_M = \sum_{l=1}^{L} \left\| \mathbf{Z}^{(l)} \right\|_0 = \sum_{l=1}^{L}\sum_{u=1}^{n}\sum_{v=1}^{n} \mathbb{I}\left[\mathbf{Z}^{(l)}_{u,v} \neq 0\right].$$

JACOBIAN REGULARIZATION

- Robust learning

$$f_i^{(l)}(\mathbf{x}+\epsilon) - f_i^{(l)}(\mathbf{x}) \approx \left[\frac{\partial f_i^{(l)}(\mathbf{x})}{\partial \mathbf{x}}\right]^{\top}\epsilon.$$

Transformer block

Jacobian

$$\mathcal{R}_J = \sum_{l=1}^{L} \|\mathbf{J}^{(l)}\|_F^2.$$

$$\|\mathbf{J}^{(l)}\|_F^2 = \operatorname{Tr}\left(\mathbf{J}^{(l)}\mathbf{J}^{(l)\top}\right) = \mathbb{E}_{\eta\in\mathcal{N}(0,\mathbf{I}_n)}\left[\left\|\eta^{\top}\mathbf{J}^{(l)}\right\|_F^2\right],$$

REC-DENOISER

Optimization

$$\mathcal{L}_{\text{Rec-Denoiser}} = \mathcal{L}_{BCE} + \beta \cdot \mathcal{R}_M + \gamma \cdot \mathcal{R}_J,$$

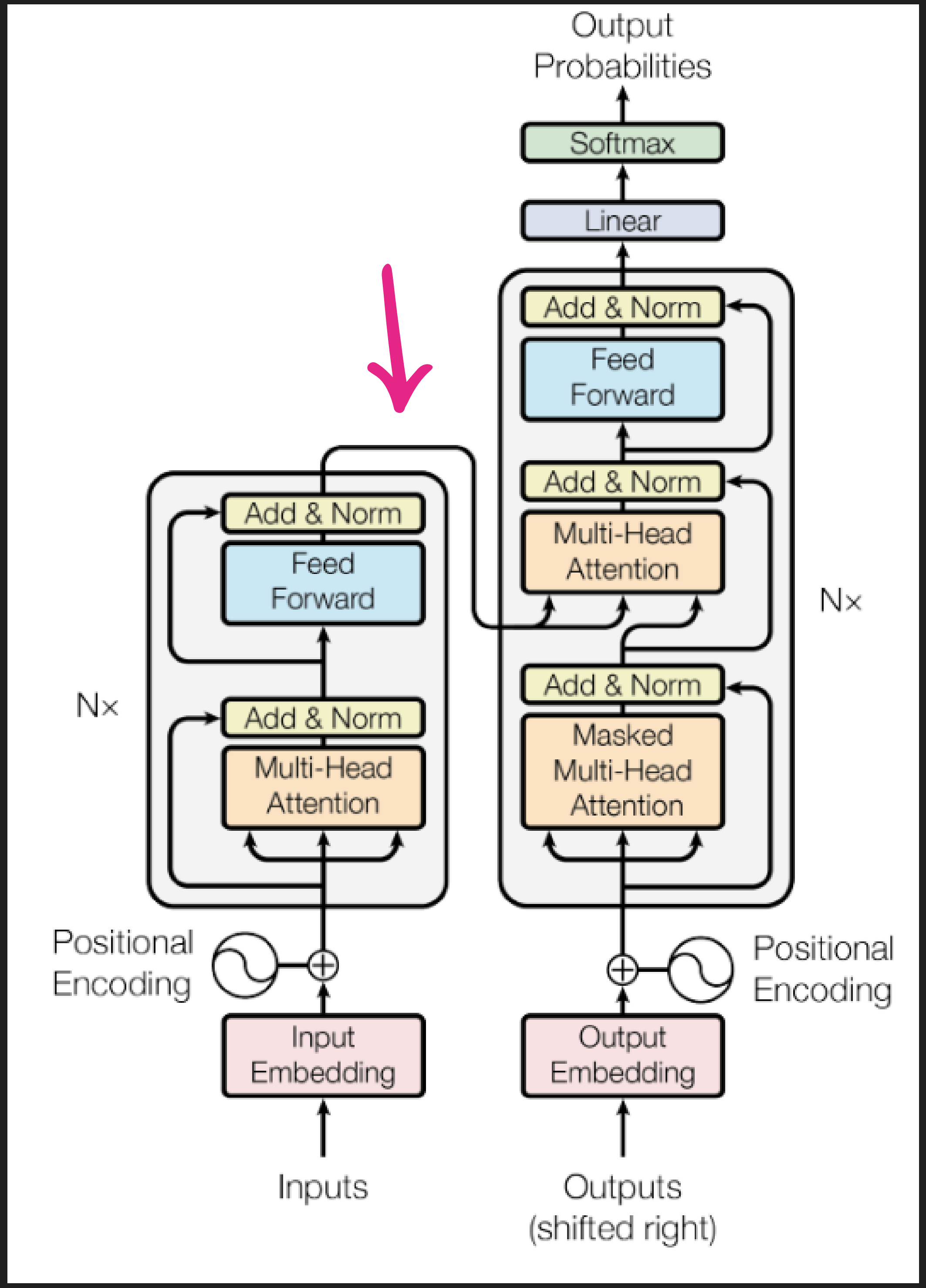
Original loss fn. for transformers

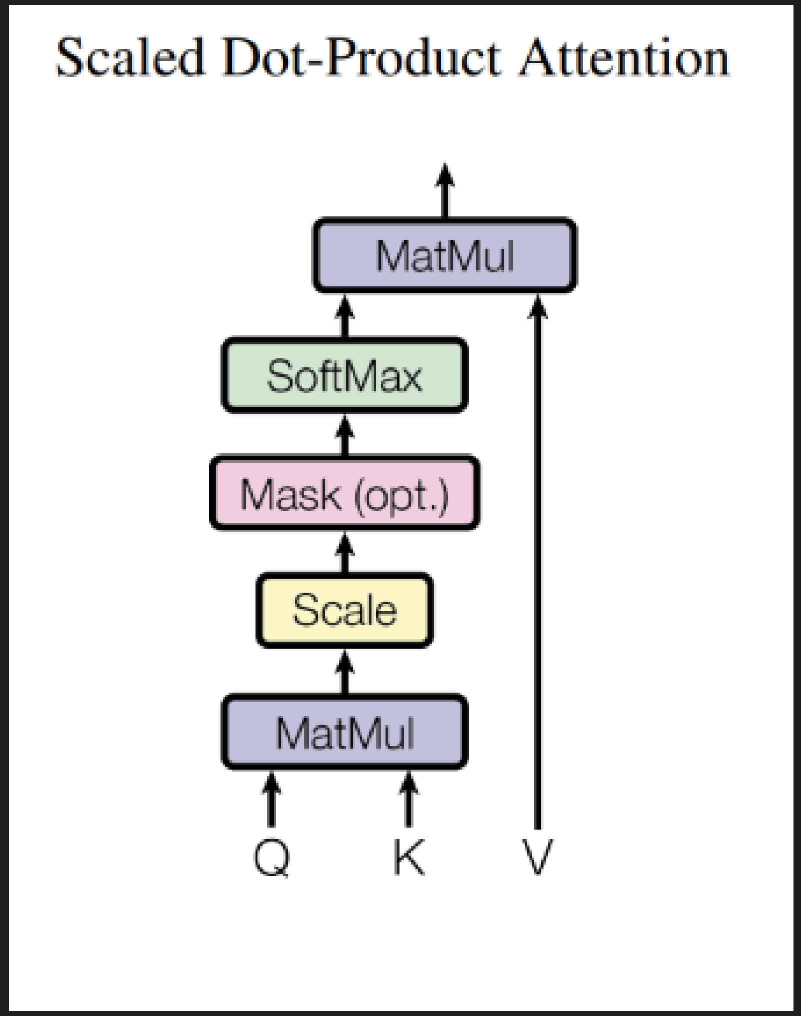
Mask fn. sparse paterns

Jacobian Regularizations

ARCHITECTURE

SASRec: Transformer-based Model





$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

05 RESULTS

Research Questions

RQ1: How effective is the proposed Rec-Denoiser compared to the state-of-the-art sequential recommenders?

RQ2: How can Rec-Denoiser reduce the negative impacts of noisy items in a sequence?

RQ3:How do different components (e.g., differentiable masks and Jacobian regularization) affect the overall
performance of Rec-Denoiser?

RQ 1: How effective is the proposed Rec-Denoiser?

Table 2. Overall Performance of different models. "RI" denotes the relative improvement of Rec-Denoisers over their backbones. The best performing results are boldfaced, and the second best ones are underlined.

Dataset	MovieLens		Beauty		Games		Movies&TV		Steam	
Metrics	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10
FPMC [39]	0.7478	0.4889	0.2810	0.1792	0.5231	0.3410	0.4806	0.3174	0.6012	0.4084
GRU4Rec [20]	0.5582	0.3383	0.2123	0.1205	0.2943	0.1939	0.4210	0.2343	0.4184	0.2687
Caser [42]	0.7213	0.4672	0.2670	0.1531	0.4315	0.2652	0.4987	0.3120	0.7137	0.4810
SASRec [26]	0.7434	0.5012	0.4345	0.2765	0.6748	0.4622	0.6521	0.4093	0.7723	0.5514
SASRec+Denoiser	0.7980	0.5610	0.4783	0.3025	0.7391	<u>0.5439</u>	<u>0.7056</u>	<u>0.4718</u>	0.8345	0.5946
+RI (%)	7.34%	11.93%	10.08%	9.40%	9.53%	17.68%	8.20%	15.27%	5.05%	7.83%
BERT4Rec [41]	0.7549	0.5245	0.4528	0.3013	0.6812	0.4815	0.6701	0.4216	0.7901	0.5641
BERT4Rec+Denoiser	**0.8045**	**0.5814**	0.4883	**0.3348**	**0.7415**	0.5310	**0.7212**	**0.4875**	<u>0.8410</u>	<u>0.6223</u>
+RI (%)	6.57%	10.85%	7.84%	11.12%	8.85%	10.28%	7.63%	15.63%	6.45%	10.32%
TiSASRec [30]	0.7365	0.5164	0.4532	0.2911	0.6613	0.4517	0.6412	0.4034	0.7704	0.5517
TiSASRec+Denoiser	0.7954	0.5582	**0.4962**	<u>0.3312</u>	0.7331	0.4984	0.6914	0.4671	**0.8414**	**0.6320**
+RI (%)	7.80%	8.10%	9.49%	13.78%	10.86%	10.34%	7.93%	15.79%	9.22%	14.56%
SSE-PT [50]	0.7413	0.5041	0.4326	0.2731	0.6810	0.4713	0.6378	0.4127	0.7641	0.5703
SSE-PT+Denoiser	<u>0.8010</u>	<u>0.5712</u>	<u>0.4952</u>	0.3265	<u>0.7396</u>	**0.5152**	0.6972	0.4571	0.8310	0.6133
+RI (%)	8.01%	13.31%	14.47%	19.55%	8.61%	11.68%	9.31%	10.76%	8.76%	13.51%

- Proposed Rec- denoisers consistently obtain the best performance for all datasets.
- Average of 8.04% improvement in Hit@10 and 12.42% in NDCG@10
- Self-attentive models generally perform better than other types of models.

RQ 1: How effective is the proposed Rec-Denoiser?

Table 2. Overall Performance of different models. "RI" denotes the relative improvement of Rec-Denoisers over their backbones. The best performing results are boldfaced, and the second best ones are underlined.

Dataset	MovieLens		Beauty		Games		Movies&TV		Steam	
Metrics	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10
FPMC [39]	0.7478	0.4889	0.2810	0.1792	0.5231	0.3410	0.4806	0.3174	0.6012	0.4084
GRU4Rec [20]	0.5582	0.3383	0.2123	0.1205	0.2943	0.1939	0.4210	0.2343	0.4184	0.2687
Caser [42]	0.7213	0.4672	0.2670	0.1531	0.4315	0.2652	0.4987	0.3120	0.7137	0.4810
SASRec [26]	0.7434	0.5012	0.4345	0.2765	0.6748	0.4622	0.6521	0.4093	0.7723	0.5514
SASRec+Denoiser	0.7980	0.5610	0.4783	0.3025	0.7391	0.5439	0.7056	0.4718	0.8345	0.5946
+RI (%)	7.34%	11.93%	10.08%	9.40%	9.53%	17.68%	8.20%	15.27%	5.05%	7.83%
BERT4Rec [41]	0.7549	0.5245	0.4528	0.3013	0.6812	0.4815	0.6701	0.4216	0.7901	0.5641
BERT4Rec+Denoiser	**0.8045**	**0.5814**	0.4883	**0.3348**	**0.7415**	0.5310	**0.7212**	**0.4875**	0.8410	0.6223
+RI (%)	6.57%	10.85%	7.84%	11.12%	8.85%	10.28%	7.63%	15.63%	6.45%	10.32%
TiSASRec [30]	0.7365	0.5164	0.4532	0.2911	0.6613	0.4517	0.6412	0.4034	0.7704	0.5517
TiSASRec+Denoiser	0.7954	0.5582	**0.4962**	0.3312	0.7331	0.4984	0.6914	0.4671	**0.8414**	**0.6320**
+RI (%)	7.80%	8.10%	9.49%	13.78%	10.86%	10.34%	7.93%	15.79%	9.22%	14.56%
SSE-PT [50]	0.7413	0.5041	0.4326	0.2731	0.6810	0.4713	0.6378	0.4127	0.7641	0.5703
SSE-PT+Denoiser	0.8010	0.5712	0.4952	0.3265	0.7396	**0.5152**	0.6972	0.4571	0.8310	0.6133
+RI (%)	8.01%	13.31%	14.47%	19.55%	8.61%	11.68%	9.31%	10.76%	8.76%	13.51%

- Proposed Rec- denoisers consistently obtain the best performance for all datasets.
- Average of 8.04% improvement in Hit@10 and 12.42% in NDCG@10
- Self-attentive models generally perform better than other types of models.

RQ 1: How effective is the proposed Rec-Denoiser?

Table 2. Overall Performance of different models. "RI" denotes the relative improvement of Rec-Denoisers over their backbones. The best performing results are boldfaced, and the second best ones are underlined.

Dataset	MovieLens		Beauty		Games		Movies&TV		Steam	
Metrics	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10
FPMC [39]	0.7478	0.4889	0.2810	0.1792	0.5231	0.3410	0.4806	0.3174	0.6012	0.4084
GRU4Rec [20]	0.5582	0.3383	0.2123	0.1205	0.2943	0.1939	0.4210	0.2343	0.4184	0.2687
Caser [42]	0.7213	0.4672	0.2670	0.1531	0.4315	0.2652	0.4987	0.3120	0.7137	0.4810
SASRec [26]	0.7434	0.5012	0.4345	0.2765	0.6748	0.4622	0.6521	0.4093	0.7723	0.5514
SASRec+Denoiser	0.7980	0.5610	0.4783	0.3025	0.7391	0.5439	0.7056	0.4718	0.8345	0.5946
+RI (%)	7.34%	11.93%	10.08%	9.40%	9.53%	17.68%	8.20%	15.27%	5.05%	7.83%
BERT4Rec [41]	0.7549	0.5245	0.4528	0.3013	0.6812	0.4815	0.6701	0.4216	0.7901	0.5641
BERT4Rec+Denoiser	**0.8045**	**0.5814**	0.4883	**0.3348**	**0.7415**	0.5310	**0.7212**	**0.4875**	0.8410	0.6223
+RI (%)	6.57%	10.85%	7.84%	11.12%	8.85%	10.28%	7.63%	15.63%	6.45%	10.32%
TiSASRec [30]	0.7365	0.5164	0.4532	0.2911	0.6613	0.4517	0.6412	0.4034	0.7704	0.5517
TiSASRec+Denoiser	0.7954	0.5582	**0.4962**	0.3312	0.7331	0.4984	0.6914	0.4671	**0.8414**	**0.6320**
+RI (%)	7.80%	8.10%	9.49%	13.78%	10.86%	10.34%	7.93%	15.79%	9.22%	14.56%
SSE-PT [50]	0.7413	0.5041	0.4326	0.2731	0.6810	0.4713	0.6378	0.4127	0.7641	0.5703
SSE-PT+Denoiser	0.8010	0.5712	0.4952	0.3265	0.7396	**0.5152**	0.6972	0.4571	0.8310	0.6133
+RI (%)	8.01%	13.31%	14.47%	19.55%	8.61%	11.68%	9.31%	10.76%	8.76%	13.51%

- **Proposed Rec- denoisers consistently obtain the best performance for all datasets.**
- **Average of 8.04% improvement in Hit@10 and 12.42% in NDCG@10**
- **Self-attentive models generally perform better than other types of models.**

RQ 2: How can Rec-Denoiser reduce the negative impacts of noisy items?

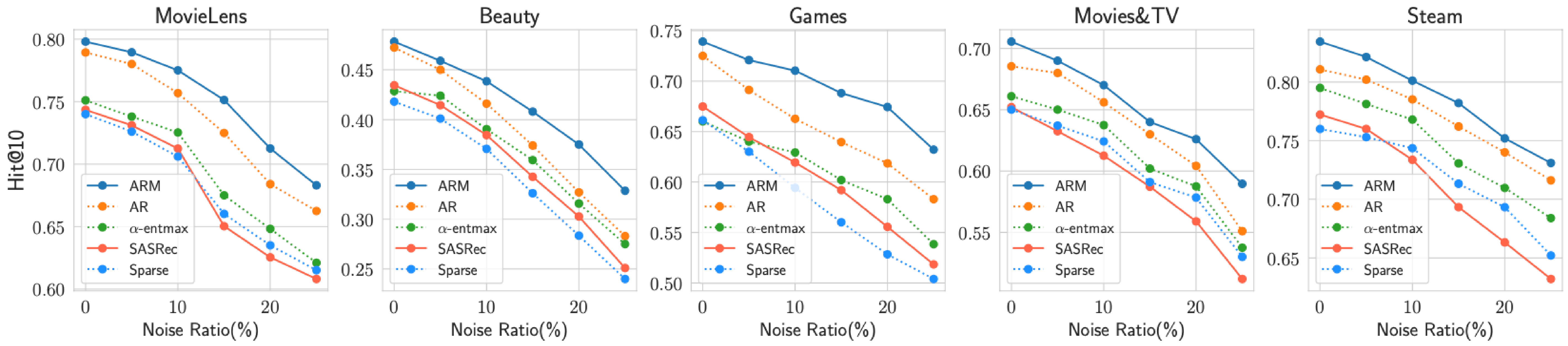


Fig. 2. Overall performance on the training data are corrupted by synthetic noises.

- **Rec-Denoiser shows robustness against noisy data in sequences, outperforming SASRec.**
- **Training data corruption strategy: up to 25% items replaced with random, unrelated items.**
- **Superior performance is maintained even with up to 25% corrupted data.**
- **Differentiable masks and Jacobian regularization contribute to model resilience.**

*Sólo se reportan los resultados de SASRec y SASRec-Denoiser por temas de espacio del paper, el desempeño de los otros modelos fue similar pero se omite por limitación de páginas.

RQ 3: How can Rec-Denoiser reduce the negative impacts of noisy items?

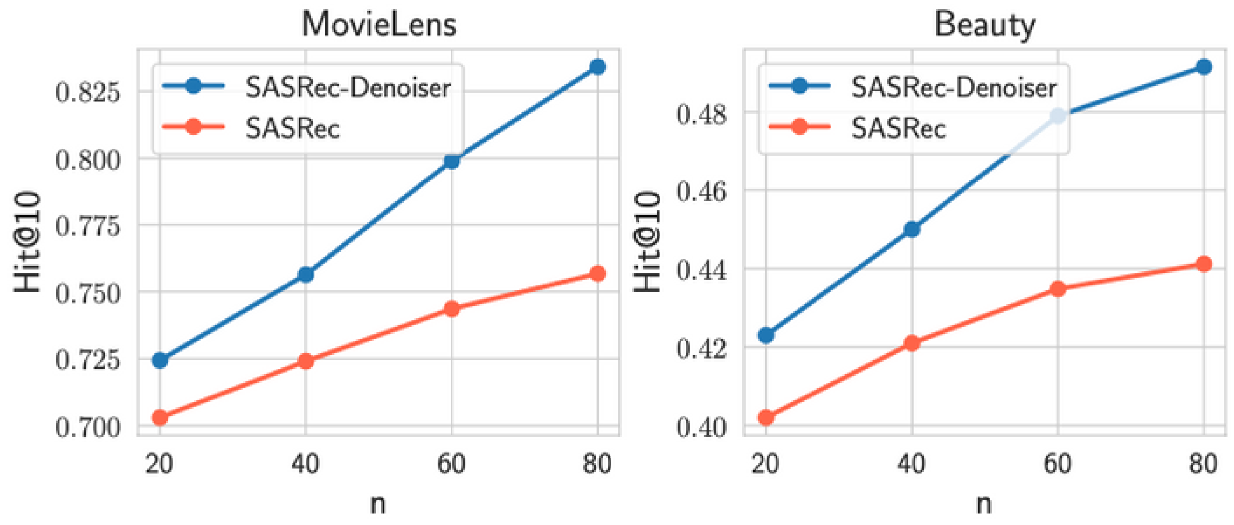


Fig. 3. Effect of maximum length n on ranking performance (Hit@10).

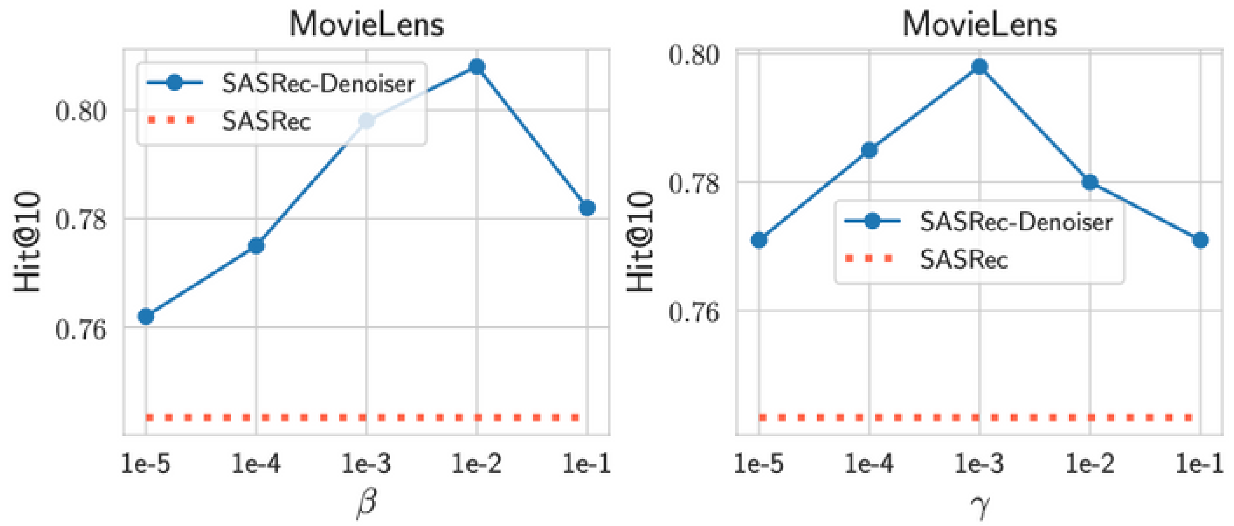


Fig. 4. Effect of regularizers β and γ on ranking performance (Hit@10).

- **Self-attentive models typically benefit from small values of H (number of heads) and L (number of blocks)**
- **Rec-Denoiser shows improved performance with longer sequences, ideal for dense datasets.**
- **Parameters β and γ for sparsity and gradient smoothness, respectively, performance is relatively stable with respect to different settings**

- The paper mentions they only did tests on the densest and sparsest datasets: MovieLeans and Beauty, because they are bigger, therefore have a higher probability that the sequence contains noisy items.

Conclusions and future work

- Rec-Denoiser effectively mitigates the impact of noisy items in recommendations.
- Differentiable masks and Jacobian regularization improve robustness and generalization.
- Model demonstrated effectiveness across multiple real-world datasets.
- Future work will explore further applications of Rec-Denoiser beyond recommendations.

Pontificia Universidad Católica de Chile

Denoising Self-Attentive Sequential Recommendation

HUIYUAN CHEN et al.

Maureen Cooper, Lucía De Pineda y Anna Ramon