Personalized Re-ranking for Recommendation

Changhua Pei, Yi Zhang, Yongfeng Zhang,
Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang,
Junfeng Ge, Wenwu Ou, Dan Pei

Outline

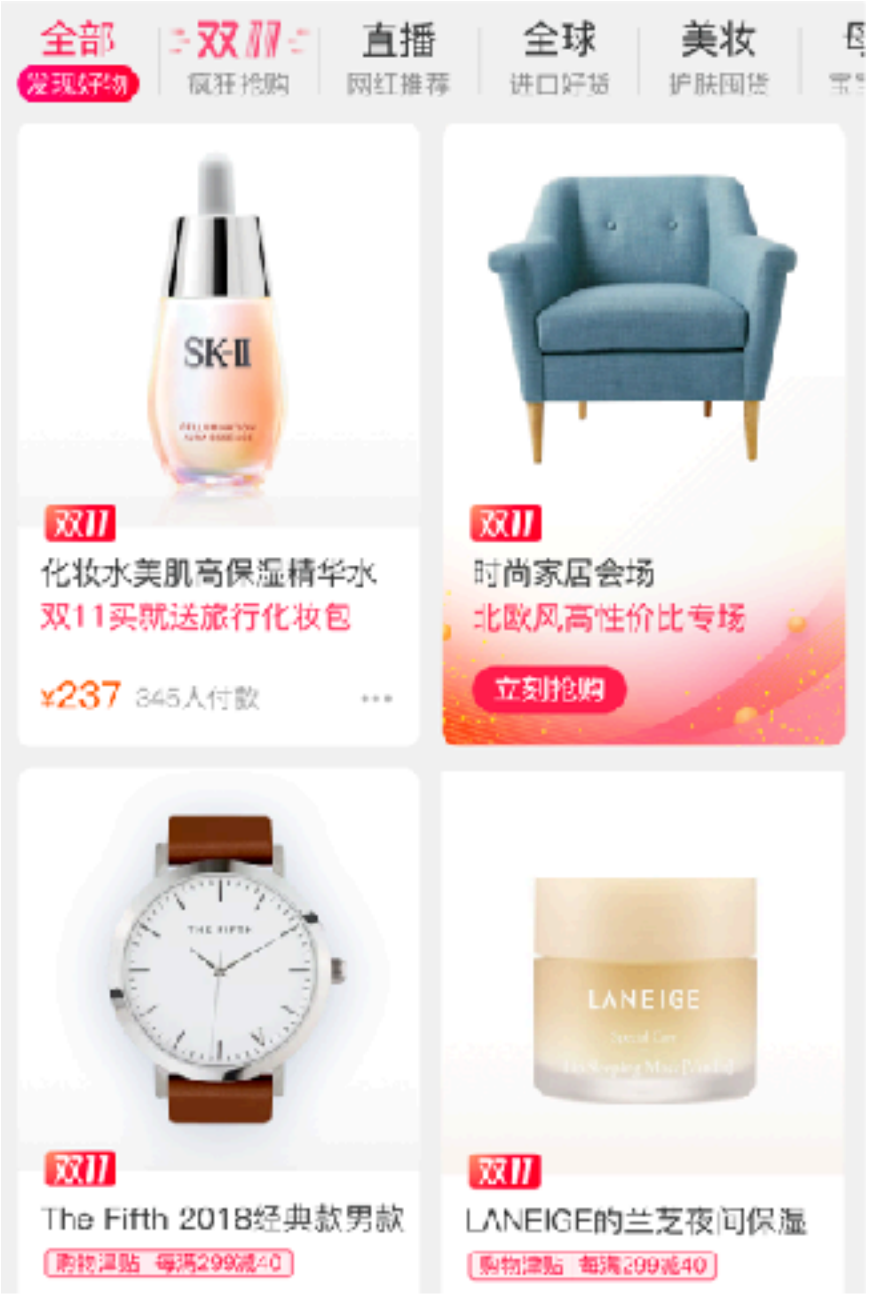
Formulation of re-ranking for recommendation

Model: How do we solve the formulated re-ranking problem?

Experiments: The performance of our model.

Conclusion

Ranking is critical for Recommendation



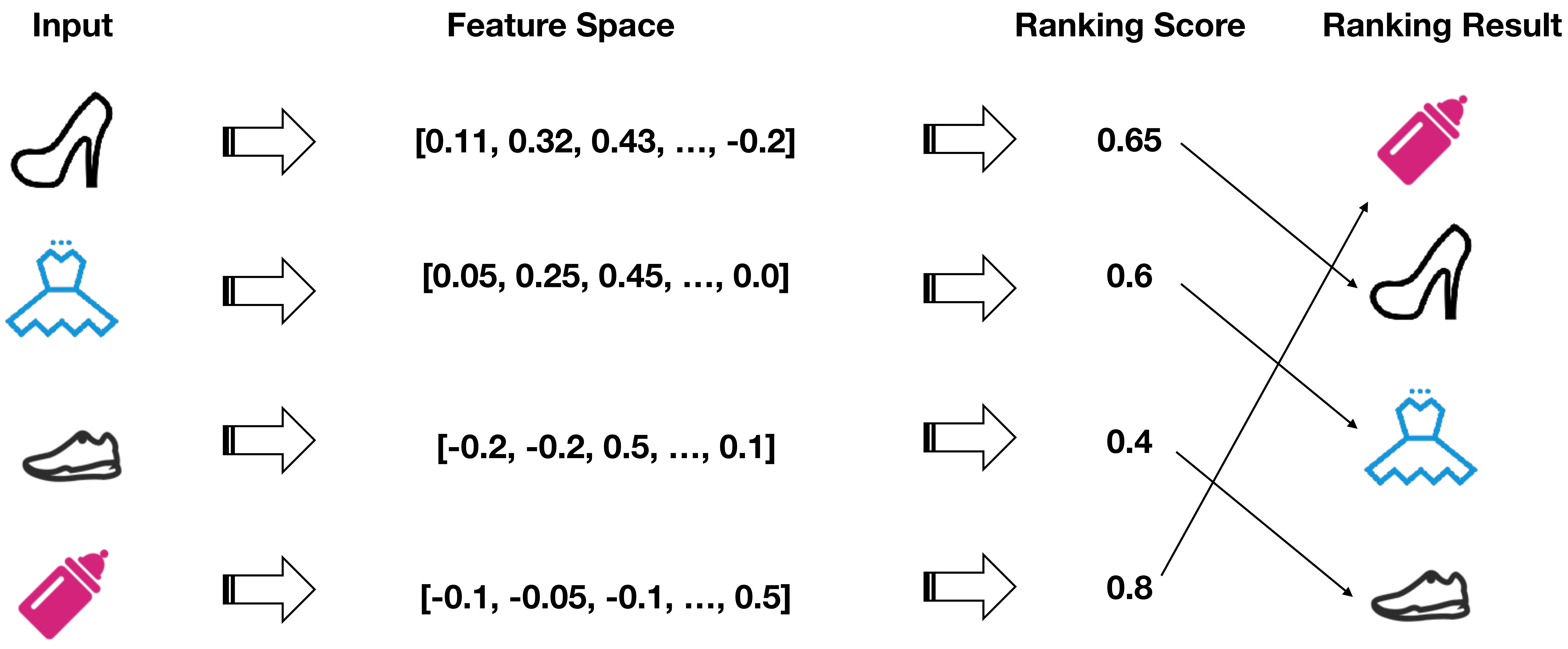
Auction

News

Restaurant

Video

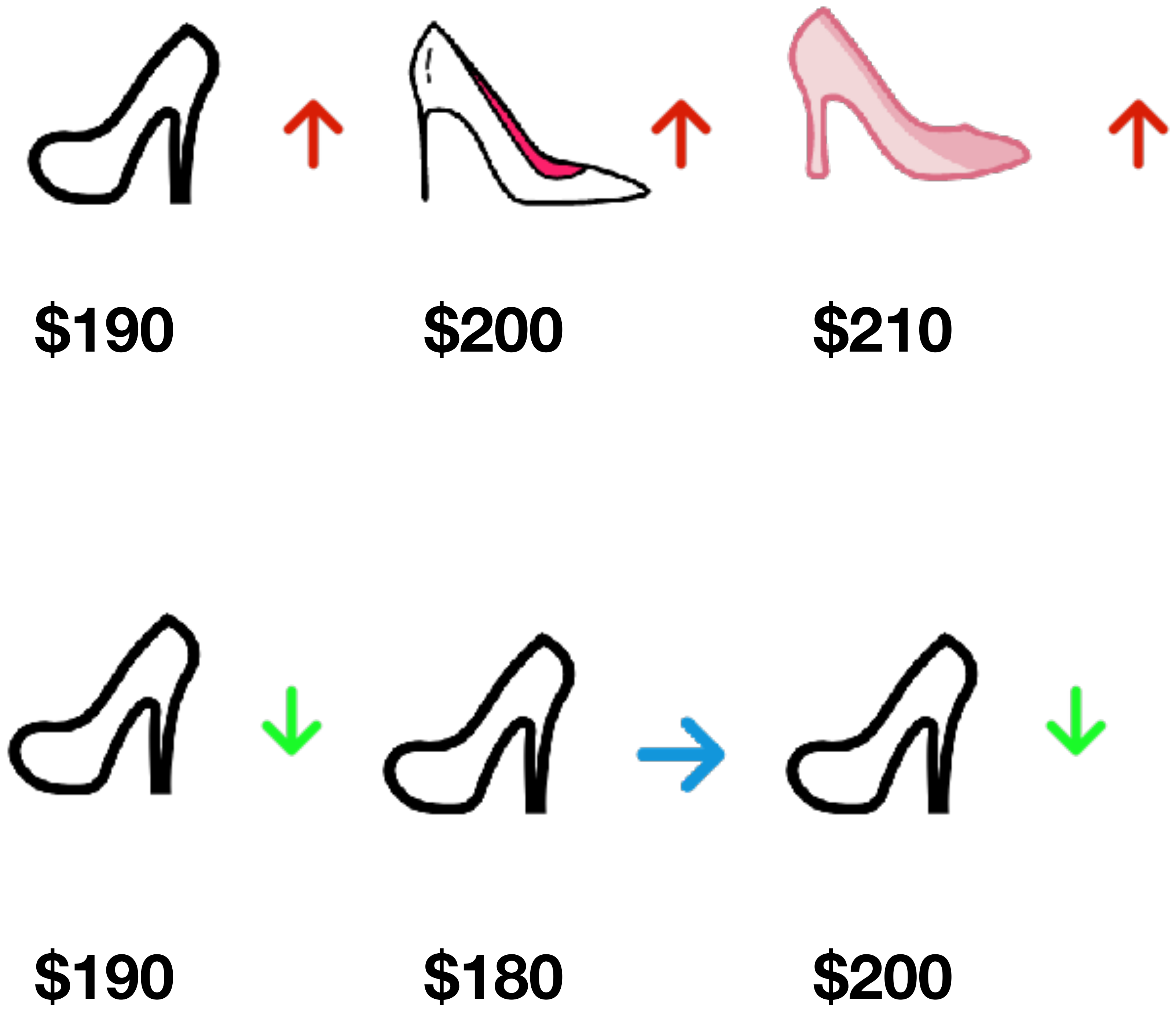
Learning to rank



The scoring process is independent for each item!

Mutual Influences between Items

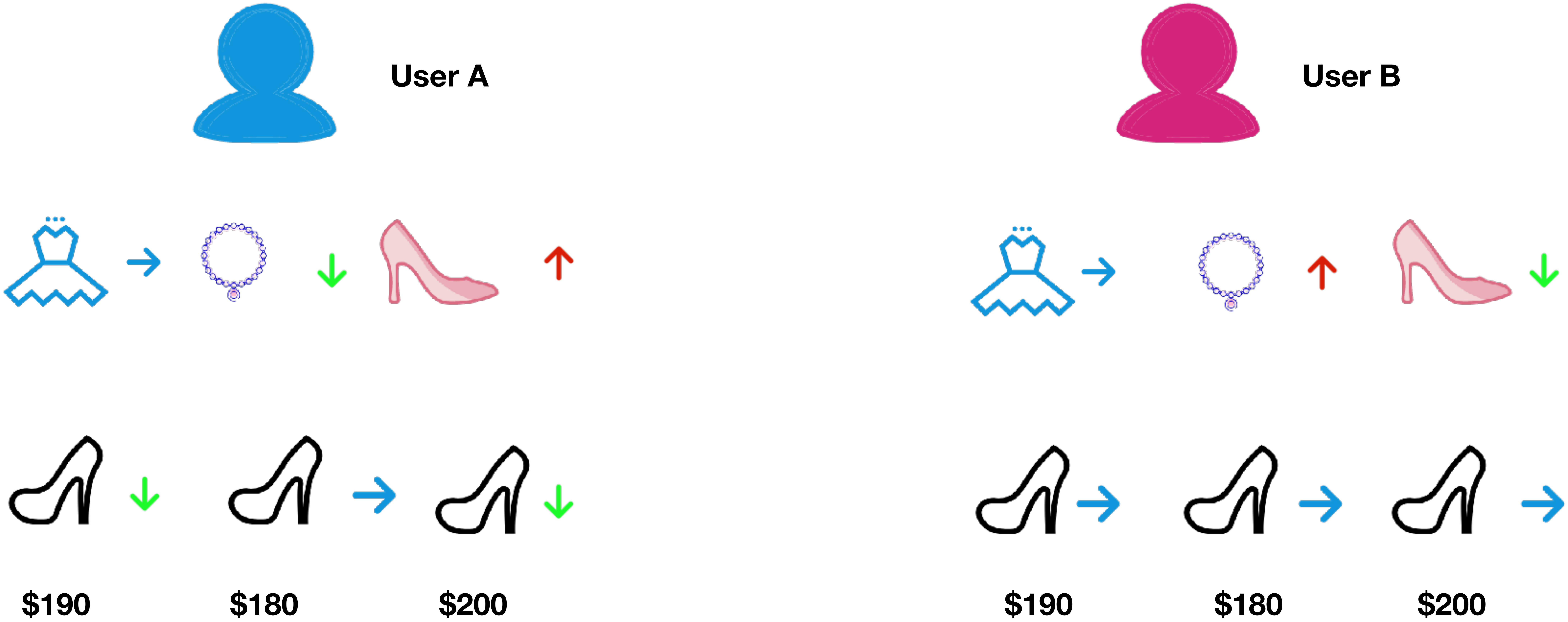
Redundancy



Supplement

The user's decision may be affected by the items placed alongside it! Either negative or positive

Personalized Mutual Influences



Different people reacts differently for mutual influences!

Our Model

Learning to rank

$$\mathcal{L} = \sum_{r \in \mathcal{R}} \ell\left(\{y_i, P(y_i | \boldsymbol{x_i}; \theta) | i \in \mathcal{I}_r\}\right)$$

Personalized Re-ranking

feature matrices of candidate items

$$\mathcal{L} = \sum_{r \in \mathcal{R}} \ell\left(\{y_i, P(y_i | \boldsymbol{X}, \boldsymbol{PV}; \hat{\theta}) | i \in \mathcal{S}_r\}\right)$$

matrices of personalized vector

sorted list of the items generated by the initial ranker

x_i **feature vector of item**

y_i **click labels of item**

I_r **candidate sets of items**

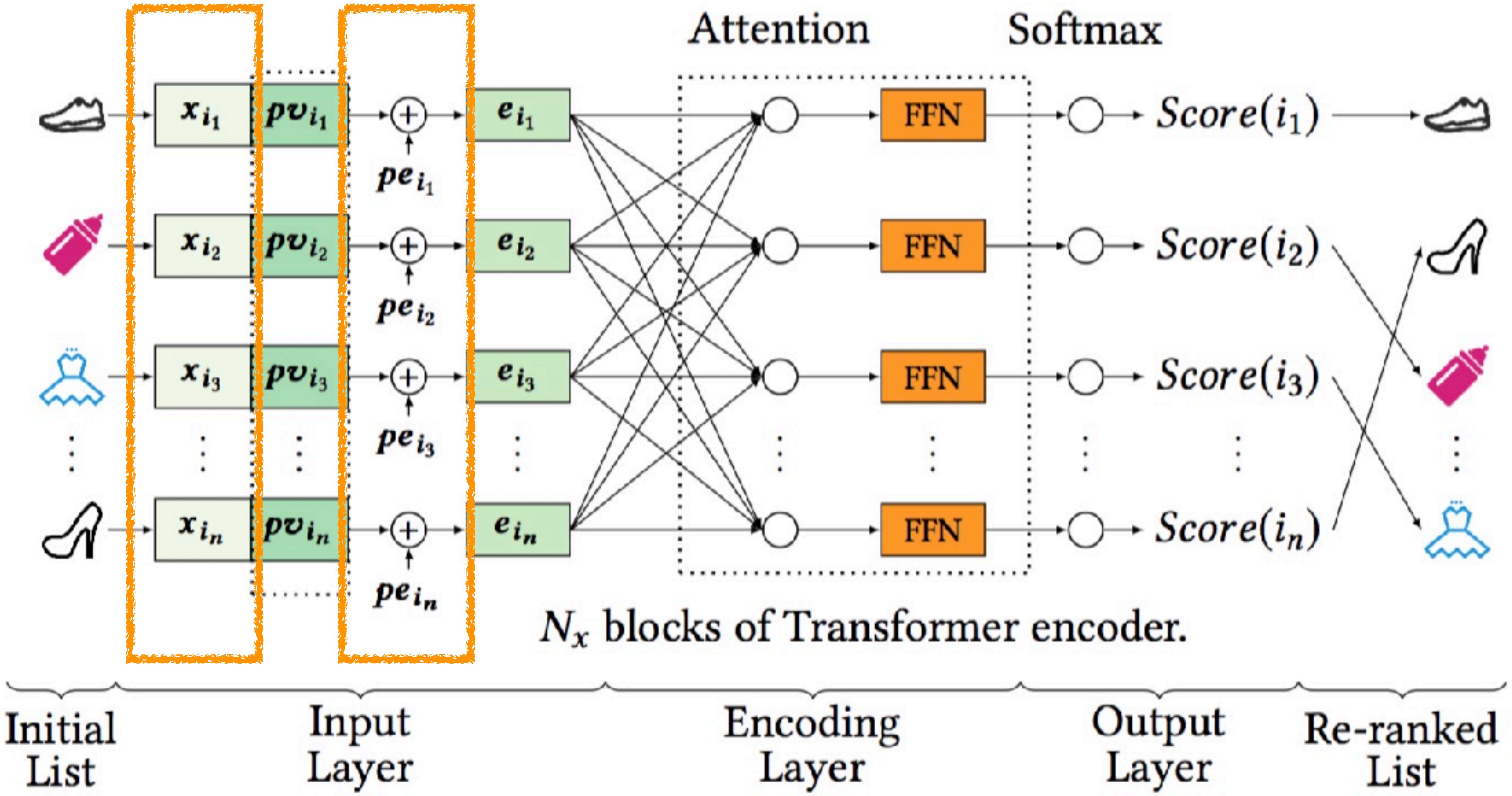
Re-ranking: act as a refinement of the learning to rank method.

Our Model

- **Modeling item-dependency**
  - Extend feature space
  - global representation
    - RNN-based
  - **Modeling relationship directly between any two items**
    - Attention-based
- **Personalized item-dependency**
  - user context features
  - **Pre-trained user preference vectors**

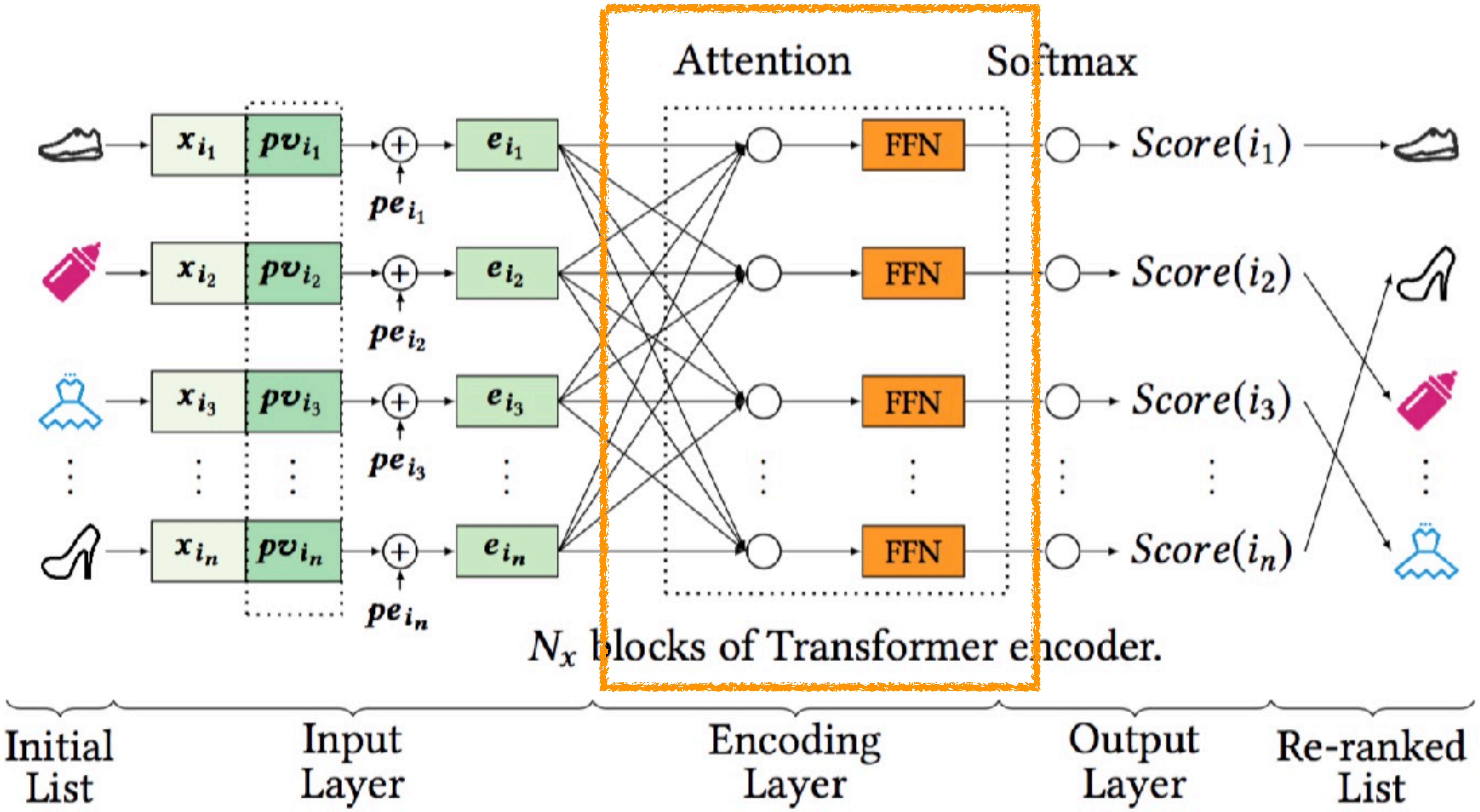
PRM: personalized re-ranking model

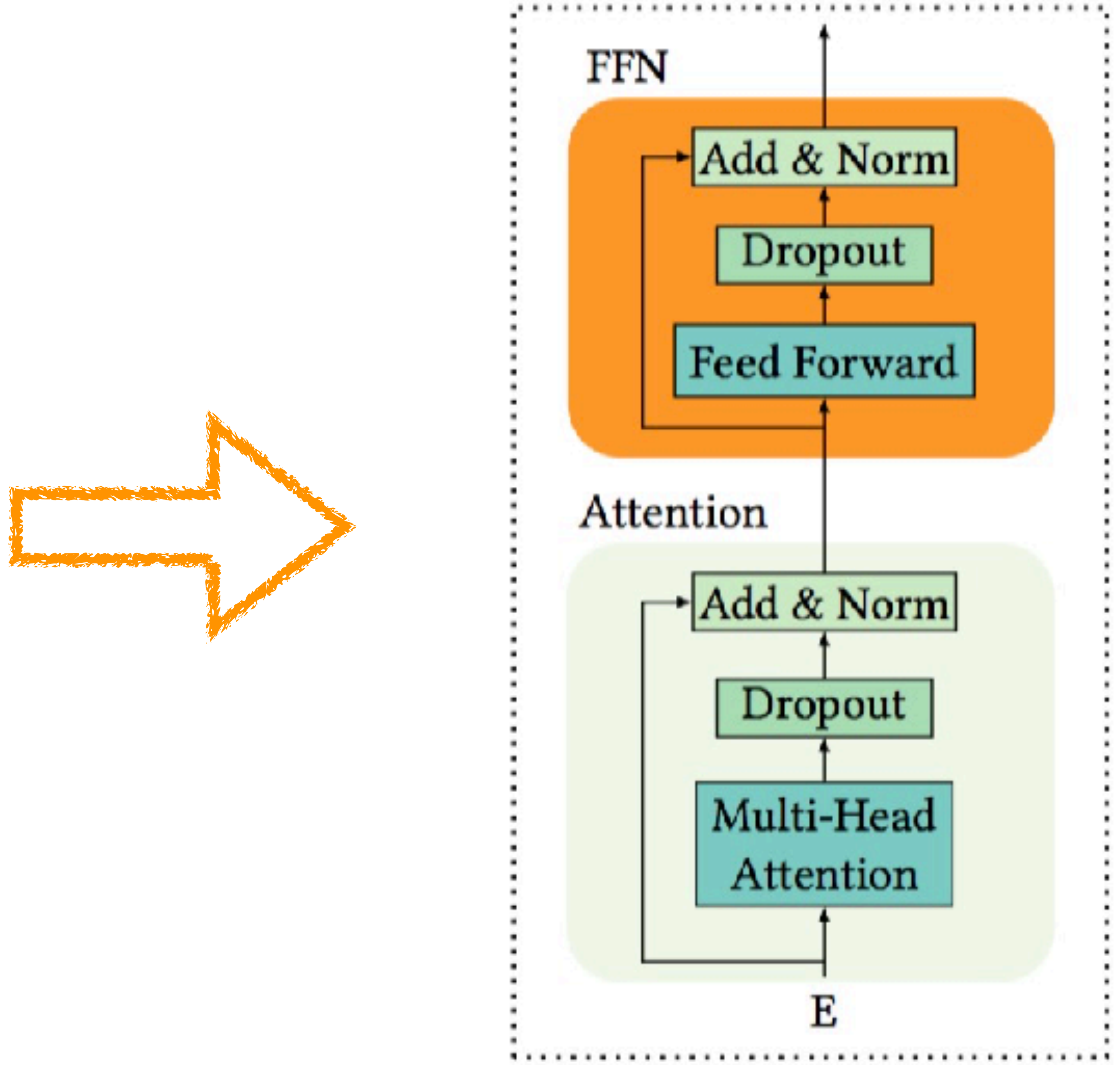
Feature vector.



Learnable position embedding

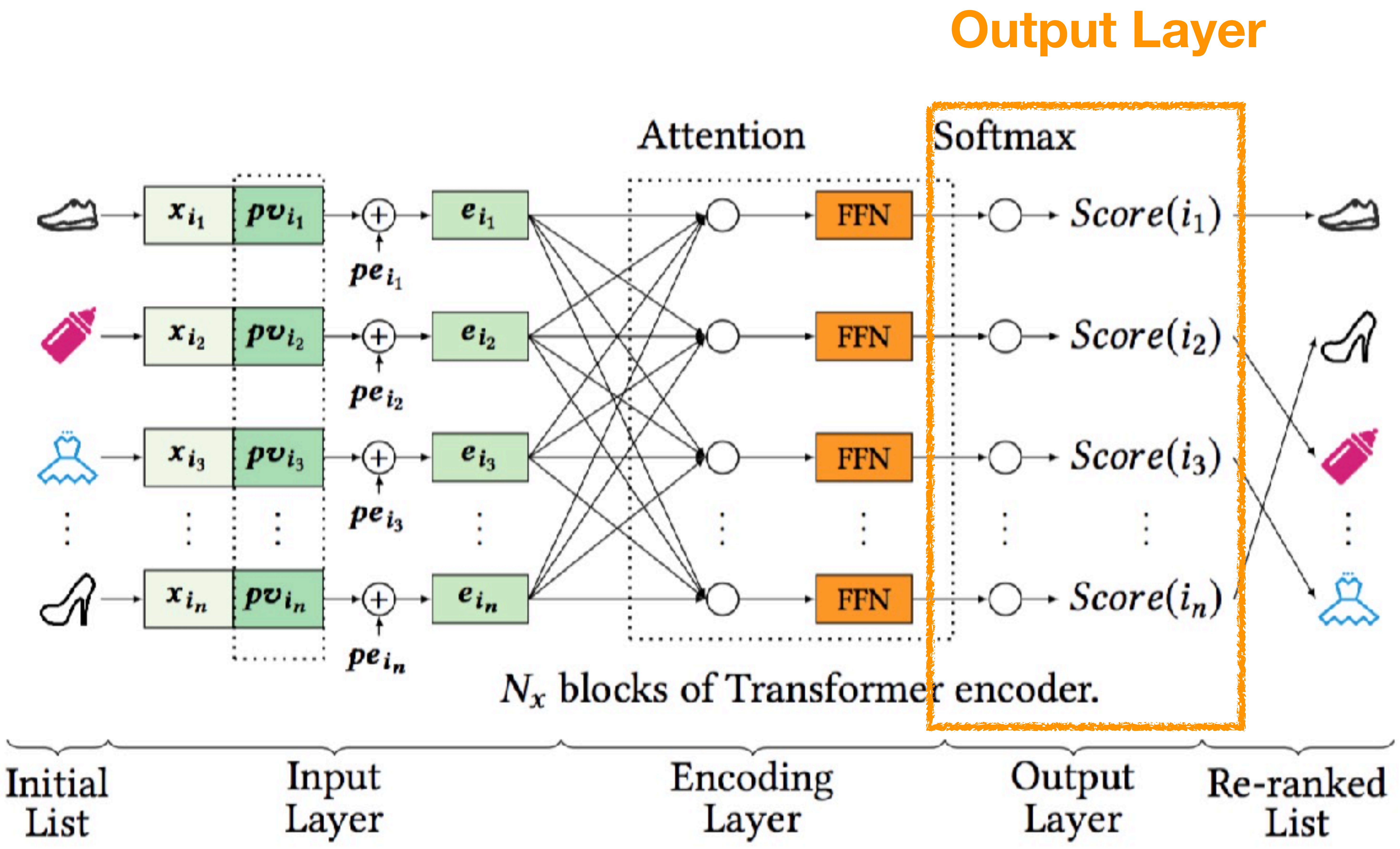
PRM: personalized re-ranking model





(a) One block of Transformer encoder.

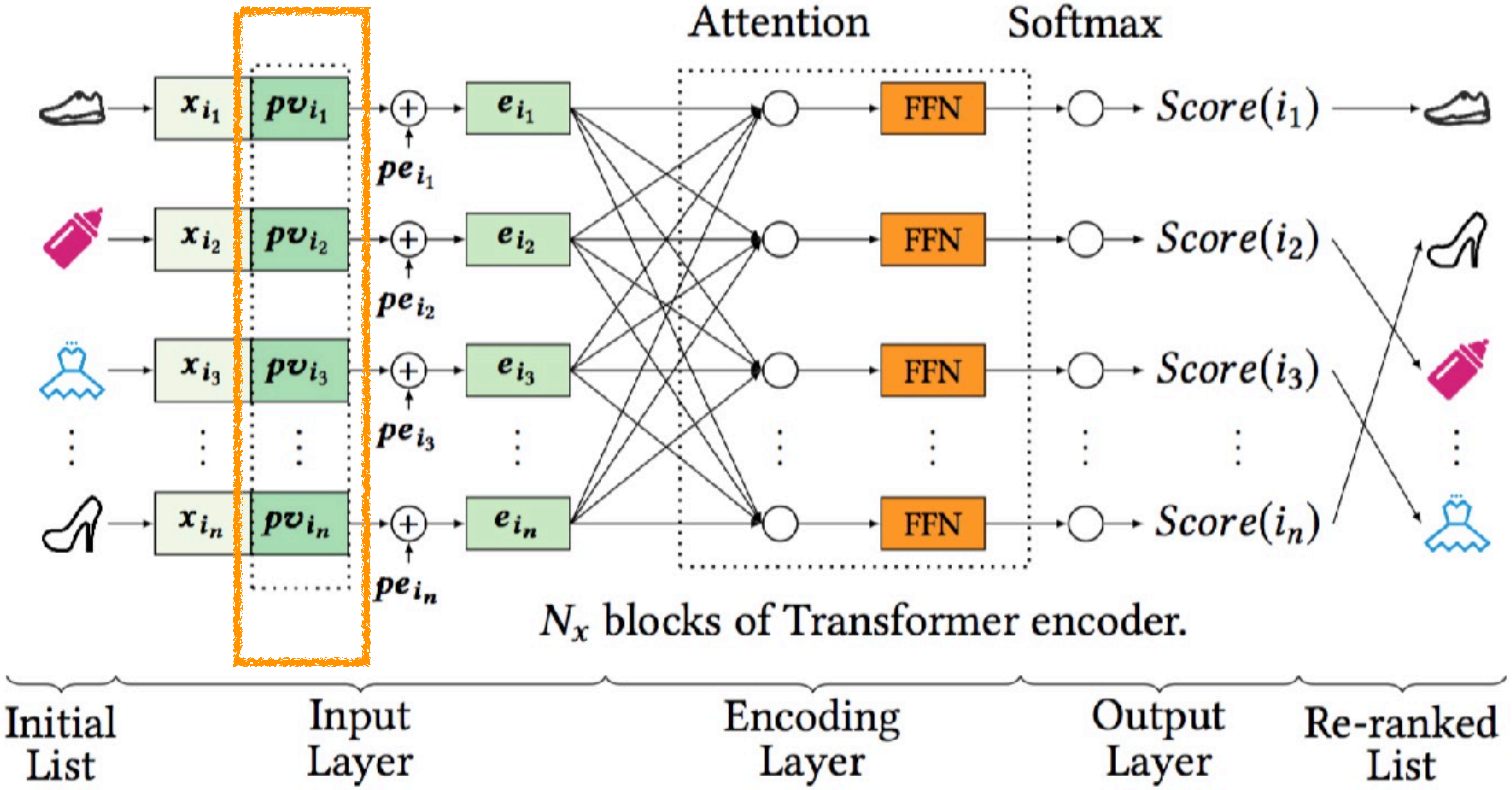
PRM: personalized re-ranking model



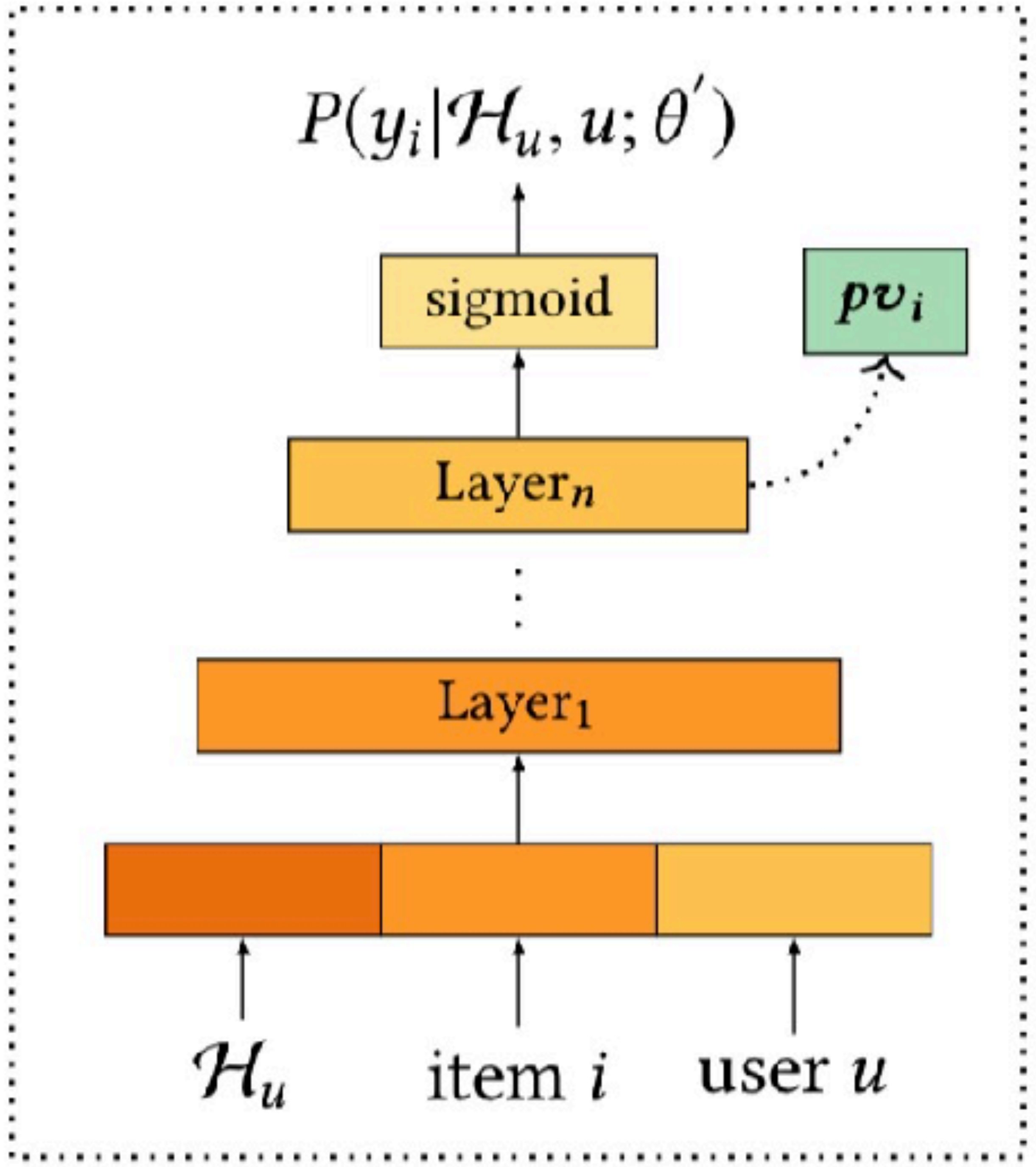
Loss function:

$$\mathcal{L} = -\sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{S}_r} y_i \log(P(y_i|\boldsymbol{X},\boldsymbol{PV};\hat{\theta}))$$

PRM: personalized re-ranking model



Pre-trained Model: click or not



1. hidden vector contains more information.
2. Pre-trained model can utilize the user click history.

Loss function:

$$\mathcal{L} = \sum_{i \in \mathcal{D}} (y_i \log(P(y_i|\mathcal{H}_u, u; \theta')) + (1 - y_i) \log(1 - P(y_i|\mathcal{H}_u, u; \theta')),$$

Experiments

- Evaluation datasets.
  - Yahoo Letor Dataset.
  - Our own dataset.
    - For personalization.

Table 2: Overview of the datasets.

	Yahoo Letor Dataset	E-commerce Re-ranking Dataset
#Users	-	743,720
#Docs/Items	709,877	7,246,323
#Records	29,921	14,350,968
Relavance/Feedback	{0,1,2,3,4}	{0,1}

We release this dataset to public!

https://github.com/rank2rec/rerank

Experiments

- Baselines
  - SVMRank
  - LambdaMart
  - DNN-based LTR
  - **DLCM (Re-ranking Method)**
  - **PRM: personalized re-ranking model.**

- Metrics
  - Offline metrics.

$$Precision@k = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \frac{\sum_{i=1}^{k} \mathbf{I}(\mathcal{S}_r(i))}{k}$$

$$MAP@k = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \frac{\sum_{i=1}^{k} Precision@i * \mathbf{I}(\mathcal{S}_r(i))}{k}$$

  - Online metrics.
    - PV/IPV/CTR/GMV

Experiments

- Research Questions:
  - RQ0: Does our PRM model outperform the state-of-the-art methods and why?
  - RQ1: Does the performance vary according to initial lists generated by different LTR approaches?
  - RQ2: What is the performance of our PRM model equipped with personalized module?
  - RQ3: Can self-attention mechanism learn meaningful information with respect to different aspects, for example, positions and characteristics of items?

Experiments

- RQ0: Does our PRM model outperform the state-of-the-art methods and why?
- RQ1: Does the performance vary according to initial lists generated by different LTR approaches?

Table 3: Offline evaluation results on Yahoo Letor dataset.

Init. List	Reranking	Yahoo Letor dataset. Precision@5(%)	Precision@10(%)	MAP@5(%)	MAP@10(%)	MAP(%)
SVMRank	SVMRank	50.42	42.25	73.71	68.28	62.14
	LambdaMART	51.35	43.08	74.94	69.54	63.38
	DLCM	52.54	43.26	76.52	70.86	64.50
	PRM-BASE	**53.29** +1.4%	**43.66**	**77.62**	**72.02**	**65.60** +1.7%
LambdaMART	SVMRank	50.41	42.34	73.82	68.27	62.13
	LambdaMART	52.04	43.00	75.77	70.49	64.04
	DLCM	52.54	43.16	77.81	71.88	65.24
	PRM-BASE	**53.63** +2.1%	**43.41**	**78.62**	**72.67**	**65.72** +0.7%

1. **Our PRM-BASE achieves stable and significant performance improvements comparing with all baselines, regardless of the initial list.**
2. **The performance gain over DLCM mainly comes from the powerful encoding ability of Transformer.**

Experiments

- RQ3: What is the performance of our PRM model equipped with personalized module?

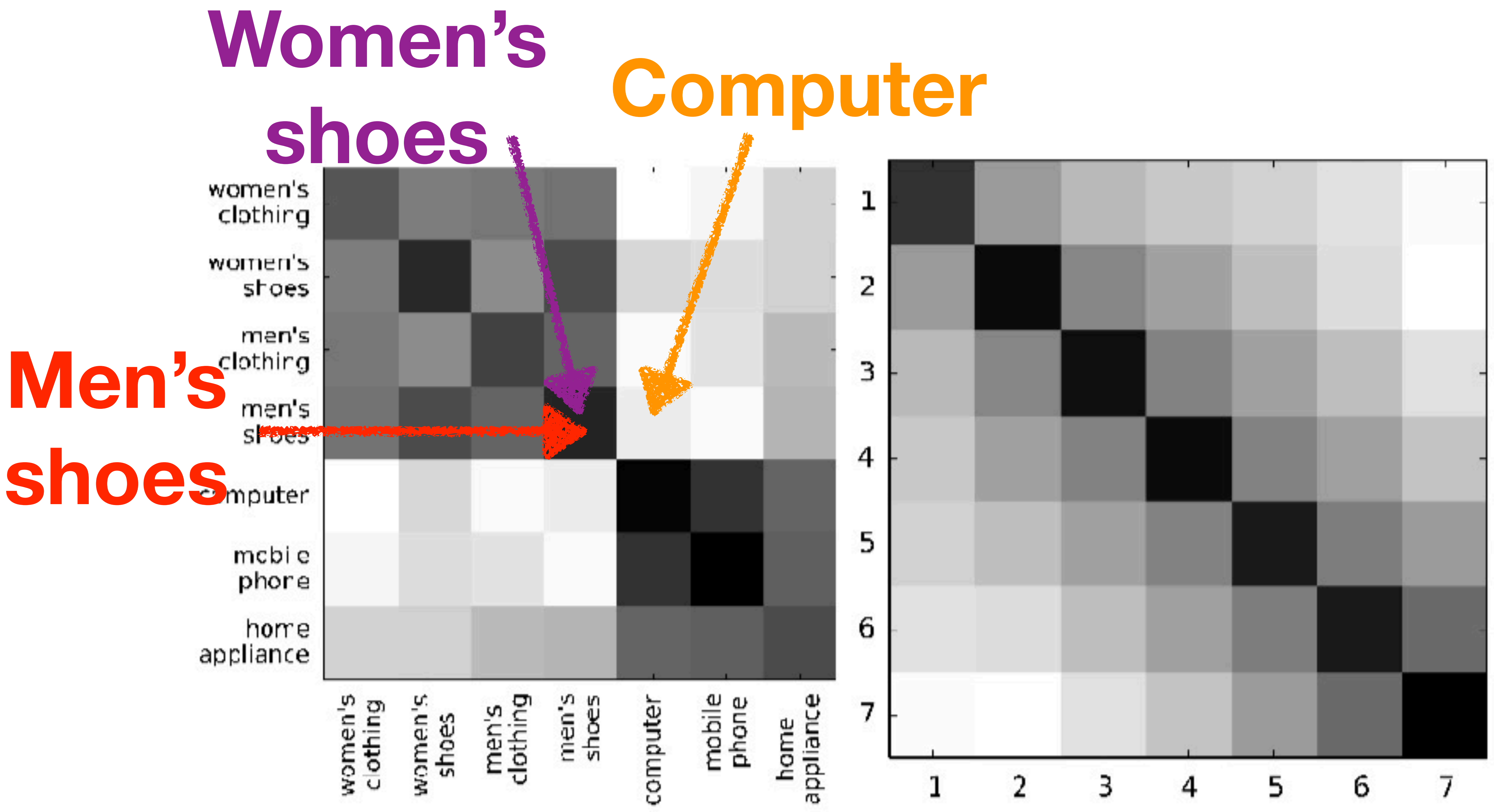
Table 5: Offline evaluation results on E-commerce Re-ranking dataset.

Init. List	Re-ranking	E-commerce Re-ranking dataset.					
		Precision@5		Precision@10	MAP@5(%)	MAP@10(%)	MAP(%)
DNN-based LTR	DLCM	12.21		9.73	29.32	30.28	28.19
	PRM-BASE	12.71	+4.1%	9.99	29.80	30.83	28.85 +2.3%
	PRM-Personalized-Pretrain	**13.58**	+9.0%	**10.52**	**31.18**	**32.12**	**30.15** +4.5%

1. **The performance gain on our E-commerce Re-ranking dataset is much larger than on Yahoo Letor dataset.**
2. **Our PRM-Personalized-Pretrain achieves significant performance improvements comparing with PRM-BASE.**

Experiments

- RQ4: Can self-attention mechanism learn meaningful information with respect to different aspects, for example, positions and characteristics of items?



(a) Category. (b) Price.

Figure 2: Average attention weights related to items' attributes.

Attention mechanism can successfully capture mutual-influences in different categories or price-level.

Conclusion

- We proposed a personalized re-ranking model (PRM) to refine the initial list given by state-of-the-art learning to rank methods.
- We used Transformer network to encode both the dependencies among items and the interactions between the user and items.
- The personalized vector can bring further performance improvements to the re-ranking model.
- Both the online and offline experiments demonstrated that our PRM model can greatly improve the ranking performance.
- Our released real-world dataset can enable researchers to study the ranking/re-ranking algorithms for recommendation systems.

Thank You!

Our Model

- Modeling item-dependency
  - **Extend feature space**
  - Global representation
    - RNN-based
  - relationship between any two items
    - Attention-based

The representation of manual features is limited!

$$f_{\min} = \min_{1 \leq j \leq N} f_l(j)$$
$$f_{\max} = \max_{1 \leq j \leq N} f_l(j)$$
$$f_g(i) = \frac{f_l(i) - f_{\min}}{f_{\max} - f_{\min}}$$

[…, 190 or 180 or 200]

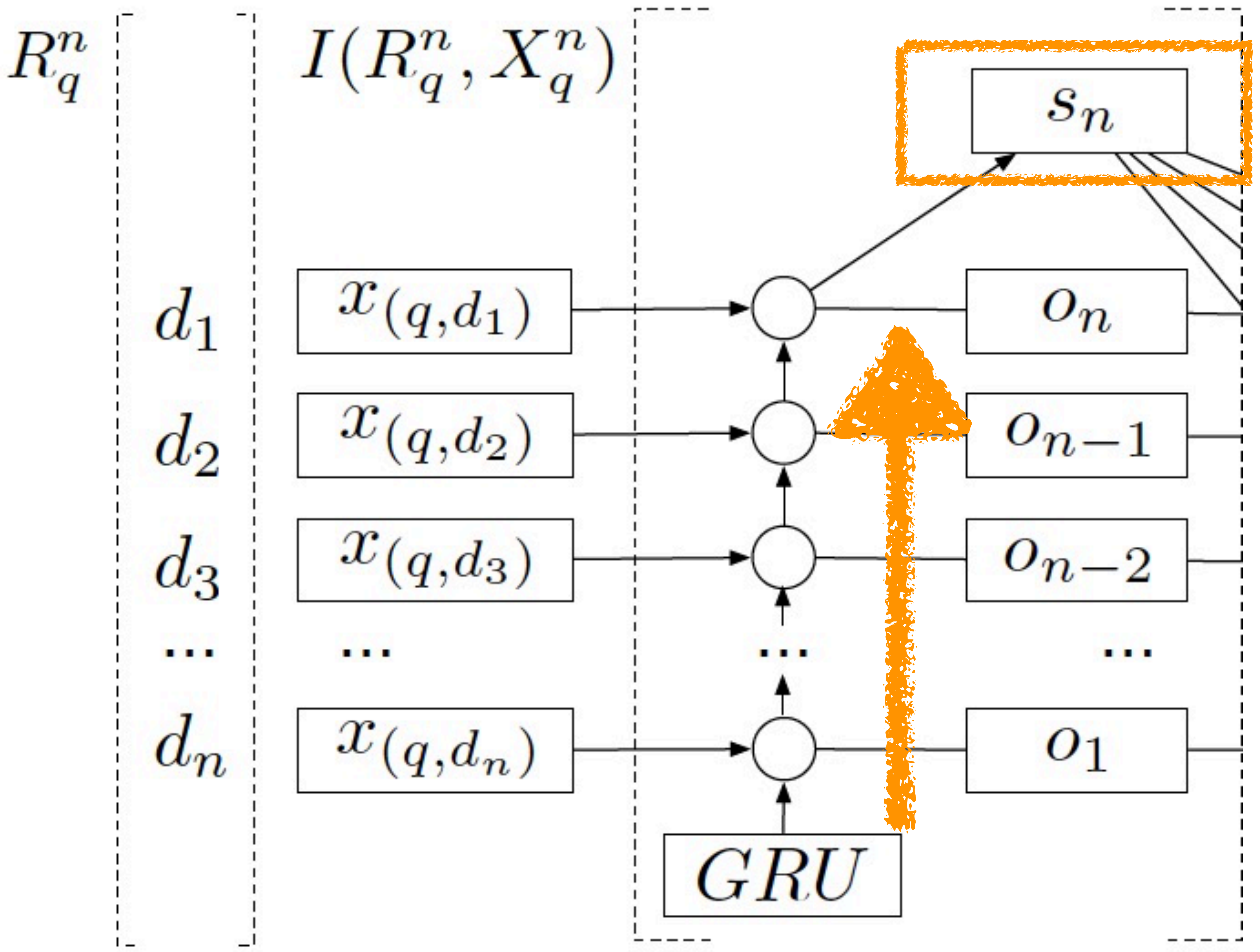
$190 **$180** **$200**

[…, 190 or 180 or 200, 0.5 or 0 or 1]

$$\frac{\$190 - \$180}{\$200 - \$180} = 0.5 \quad \frac{\$180 - \$180}{\$200 - \$180} = 0 \quad \frac{\$200 - \$180}{\$200 - \$180} = 1$$

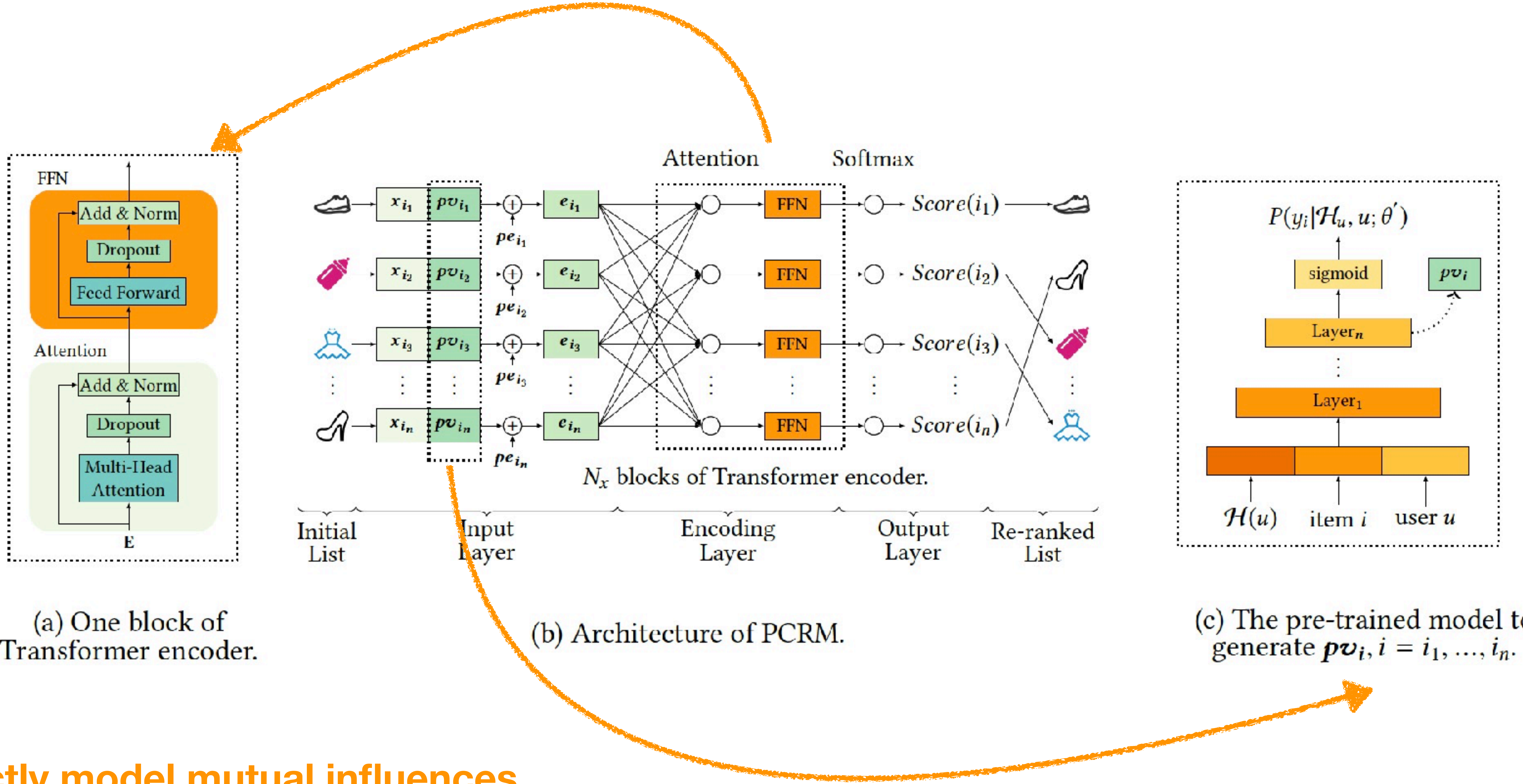
Our Model

- Modeling item-dependency
  - Extend feature space
  - **Global representation**
    - RNN-based
  - Modeling relationship between any two items
    - Attention-based



The feature information degrades along with the encoding distance.

PRM: personalized re-ranking model



(a) One block of Transformer encoder.

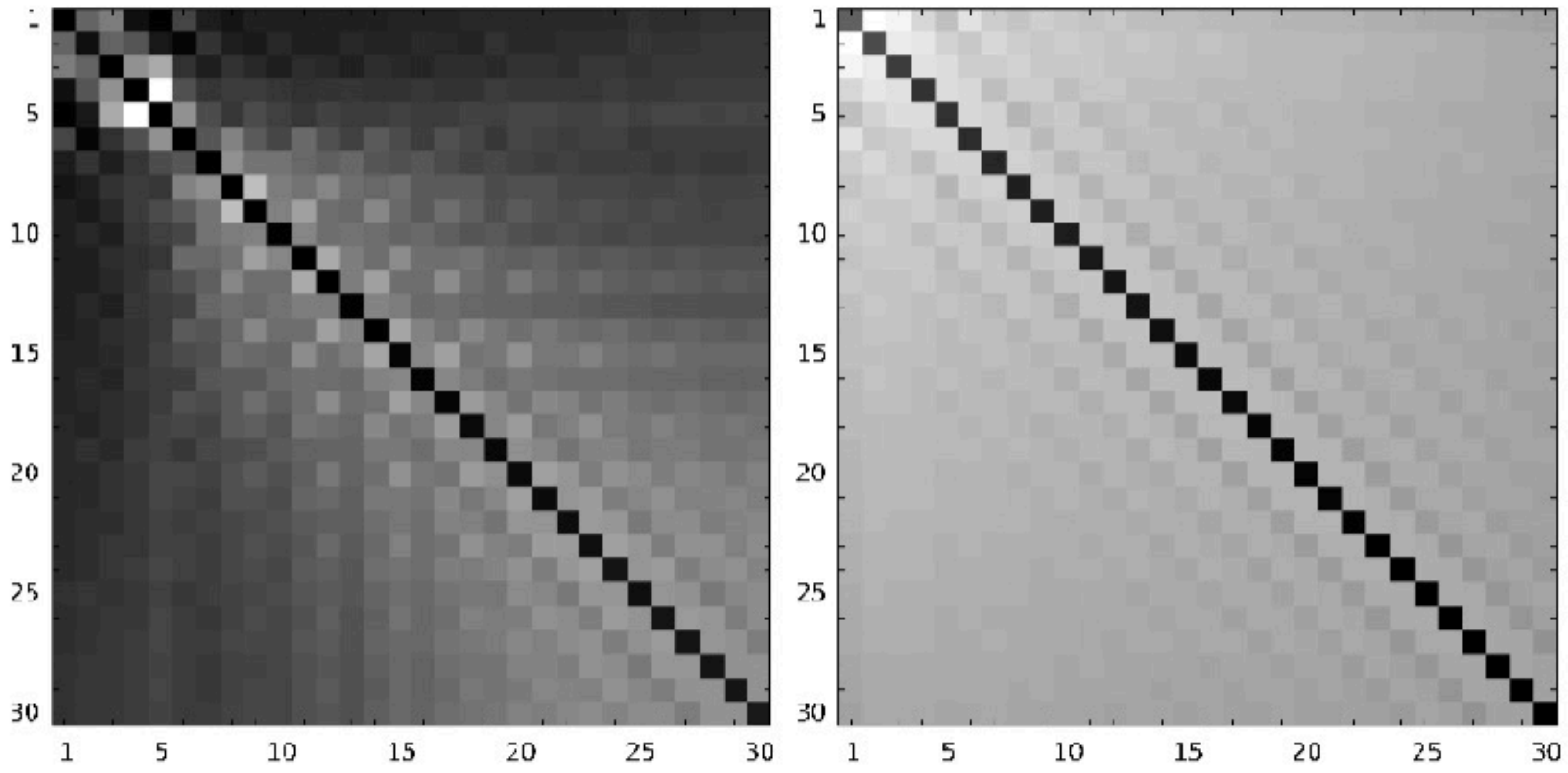
(b) Architecture of PCRM.

(c) The pre-trained model to generate $pv_i, i = i_1, \ldots, i_n$.

1. **directly model mutual influences.**
2. **enable personalized re-ranking in a flexible way.**

Experiments

- RQ3: Can self-attention mechanism learn meaningful information with respect to different aspects, for example, positions and characteristics of items?



(a) With position embedding. (b) Without position embedding.

Figure 3: Average attention weights on positions in the initial list of two PRM models: w/o position embedding.

Self-attention mechanism in our model can capture the mutual influences regardless of the encoding distances as well as the position bias in recommendation list.

Experiments

- RQ2: Which part of our model contributes most to the performance?

Table 4: Ablation study of PRM-BASE on Yahoo Letor datasets with the initial list generated by SVMRank. All the numbers in the table are multiplied by 100.

	Yahoo Letor dataset				
	P@5	P@10	MAP@5	MAP@10	MAP
DLCM	52.54	43.26	76.52	70.86	64.50
Default(b=4,h=3)	53.29	43.66	77.62	72.02	65.60
Remove PE	**52.55**	**43.56**	76.11	70.74	**64.73**
Remove RC	53.24	43.63	77.52	71.92	65.52
Remove Dropout	53.17	43.42	77.41	71.80	65.17
Block(b=1)	53.12	43.59	77.58	71.91	65.49
Block(b=2)	53.19	43.58	77.51	71.86	65.49
Block(b=6)	53.22	43.63	77.64	72.02	65.61
Block(b=8)	**52.85**	**43.32**	**77.43**	**71.65**	**65.14**
Multiheads(h=1)	53.17	43.67	77.65	71.96	65.55
Multiheads(h=2)	53.29	43.60	77.68	72.00	65.57
Multiheads(h=4)	53.20	43.61	77.72	72.00	65.58

1. **The performance of our model degrades greatly after removing position embedding.**
2. **No significant improvements are observed for different number of attention heads.**