DELF: A Dual-Embedding based Deep Latent Factor Model for Recommendation

Weiyu Cheng, Yanyan Shen, Yanmin Zhu, Linpeng Huang
Speaker: Weiyu Cheng

Department of Computer Science and Engineering
Shanghai Jiao Tong University

Outline

- Background & Motivation
- Proposed Method
- Experimental Results
- Conclusion

Background

- Collaborative filtering (CF) exploits past user-item interactions for recommendation.

- Among various CF methods, Latent factor models are widely used and considered to be the state-of-the-art solutions to recommendation.

- Recommendation based on implicit feedback has drawn more attention recently, which is easier to collect than explicit ratings.

Motivation

$$\begin{array}{cc} & i \\ u & \begin{array}{|c|c|c|c|} \hline 1 & ? & ? & 1 \\ \hline ? & 1 & ? & 1 \\ \hline ? & 1 & ? & ? \\ \hline 1 & ? & 1 & ? \\ \hline \end{array} \end{array}$$

- One of the most important problem in recommendation with implicit datasets: lack of negative feedback (also known as the one-class problem).
- Previous solutions:
  - Treating all unobserved interactions as negative feedback.
  - Using non-uniform weighting schemes to filter negative feedback from unobserved interactions.
- However, since not all the unobserved entries are true negative instances, previous solutions relying on noisy negative feedback may still hinder useful information from the limited observed interactions.

Motivation

- We notice that NSVD was proposed to directly parameterize users according to the items that they have rated.

- In NSVD, a user embedding is determined by the items that have interacted with the user, which is not easily affected by negative feedback and robust to the number of user interactions.

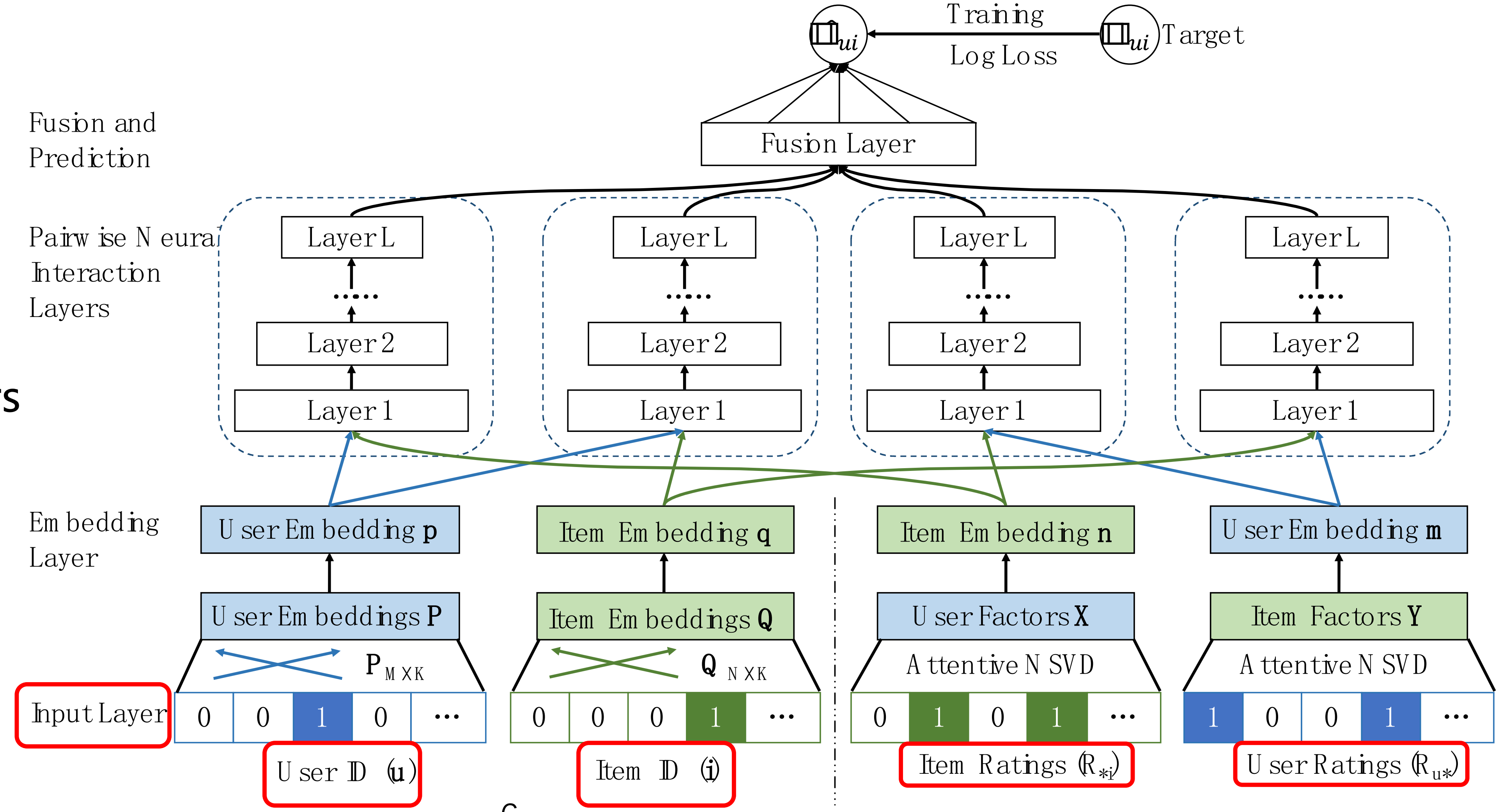
- In addition to learning a primitive embedding for a user (resp. item), we represent each user (resp. item) with an additional embedding from the perspective of the interacted items (resp. users) for recommendation with implicit feedback.

Proposed Method

Input Layer

User/Item ID: One-hot encoded vectors

User/Item Ratings: Multi-hot binary vectors



Proposed Method

Embedding Layer

Primitive embeddings:

$$\boldsymbol{p}_u = \boldsymbol{P}^T \boldsymbol{u}$$
$$\boldsymbol{q}_i = \boldsymbol{Q}^T \boldsymbol{i}$$

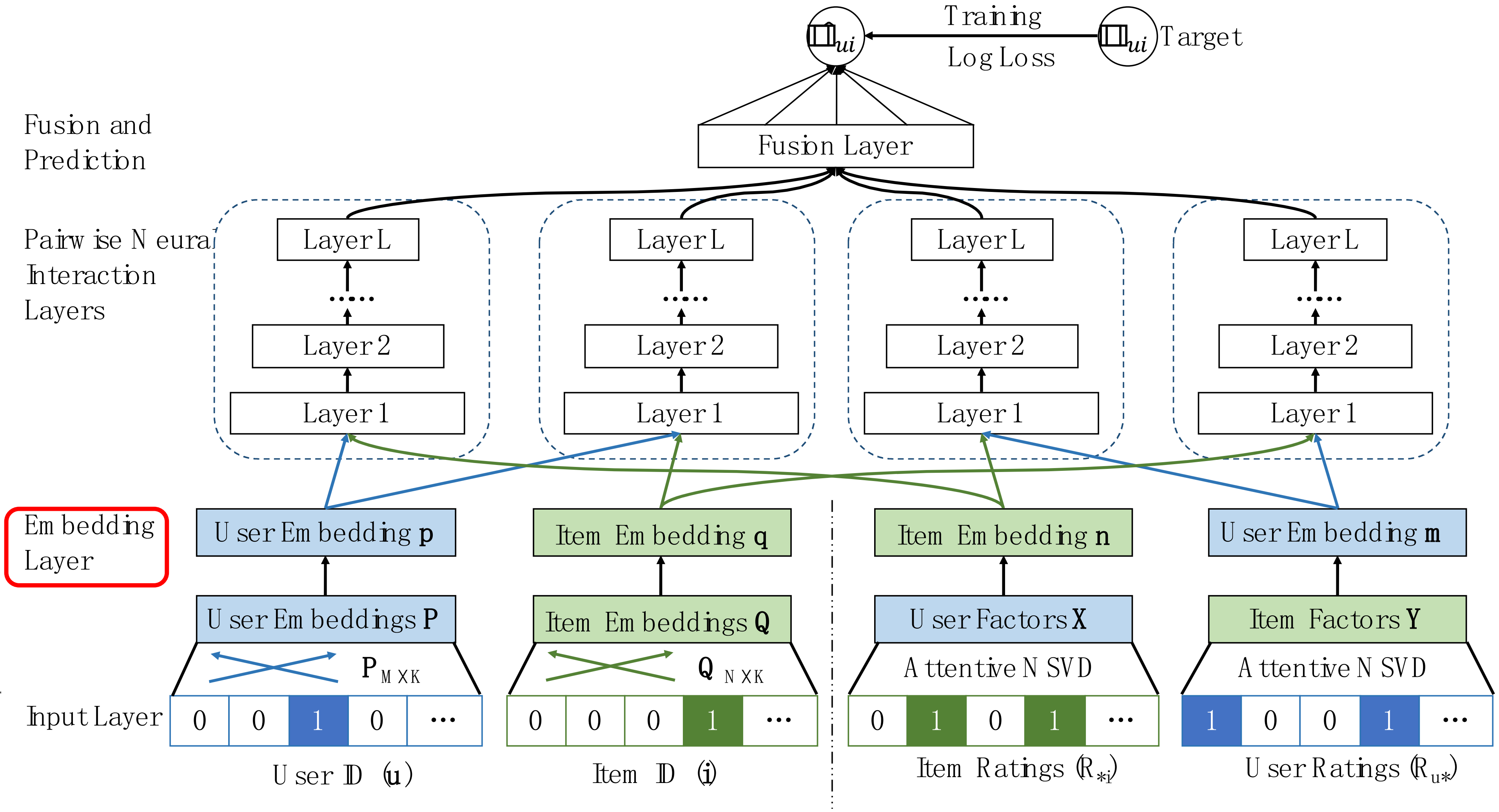
Additional embeddings:

($\mathbf{n}_i$ is omitted here)

$$\mathbf{m}_u = \sum_{i \in R(u)} \alpha_i \mathbf{y}_i$$

$$\mathbf{h}_i = \tanh(\mathbf{W}_a \mathbf{y}_i + \mathbf{b}_a)$$

$$\alpha_i = \frac{\exp(\mathbf{h}_i^T \mathbf{h}_a)}{\sum_{i \in R(u)} \exp(\mathbf{h}_i^T \mathbf{h}_a)}$$



Proposed Method

Pairwise Neural Interaction Layers

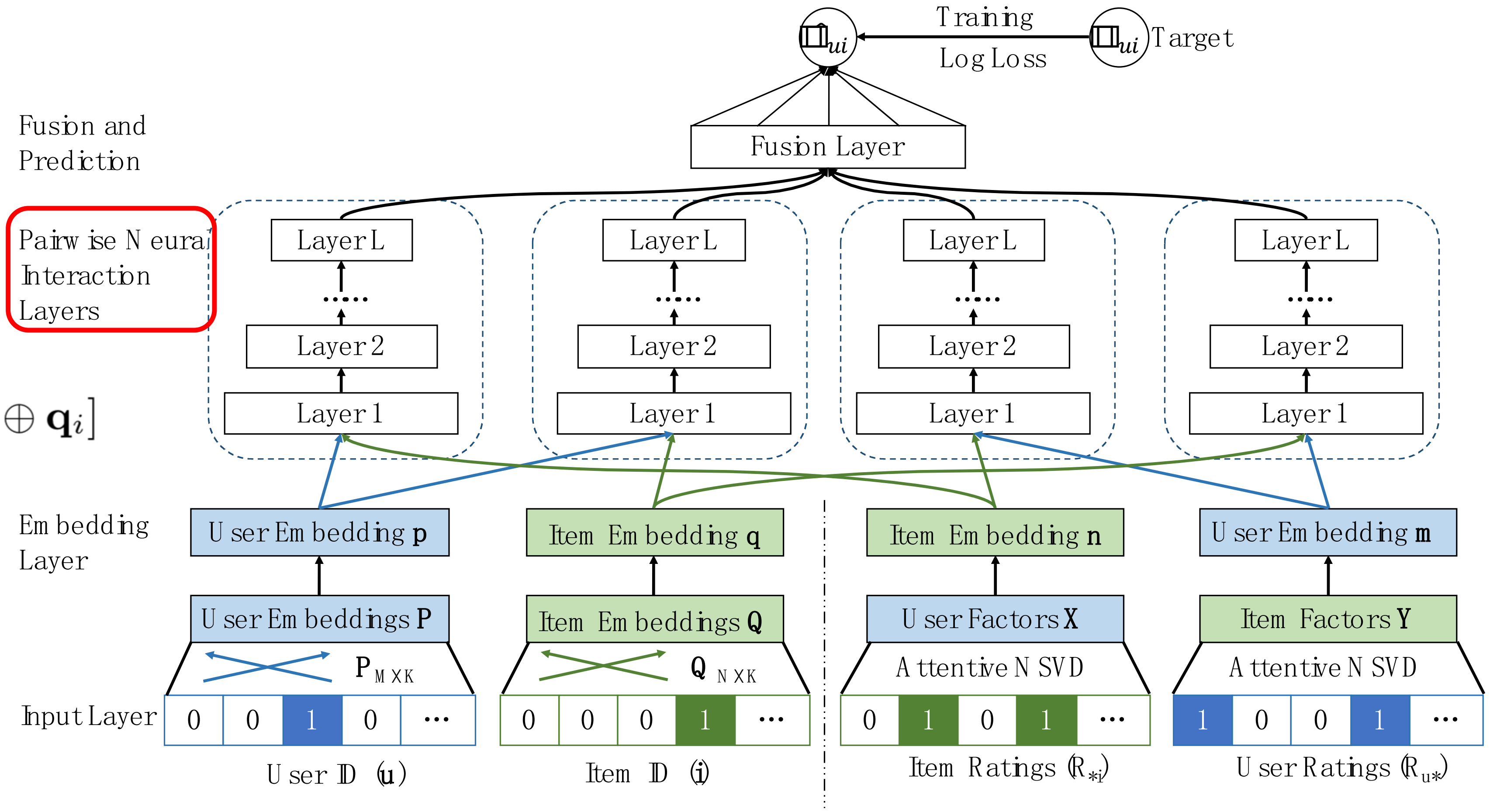
Four groups of concatenated embedding pairs:

$$\mathbf{z}_0 = [\mathbf{p}_u \oplus \mathbf{n}_i, \mathbf{p}_u \oplus \mathbf{q}_i, \mathbf{m}_u \oplus \mathbf{n}_i, \mathbf{m}_u \oplus \mathbf{q}_i]$$

Neural interaction layers:

$$\phi_l^j = \delta_l^j(\mathbf{W}_l^j \mathbf{z}_{l-1}^j + \mathbf{b}_l^j), \quad l \in [1, L]$$

$$\mathbf{h}^j = \phi_L^j(\ldots\phi_2^j(\phi_1^j(\mathbf{z}_0[j]))\ldots)$$



Proposed Method

Fusion and Prediction

Fusion with MLP:

$$\mathbf{h}_f = \delta_f(\mathbf{W}_f \mathbf{z}_f + \mathbf{b}_f)$$

$$\mathbf{z}_f = \mathbf{h}^1 \oplus \mathbf{h}^2 \oplus \mathbf{h}^3 \oplus \mathbf{h}^4$$

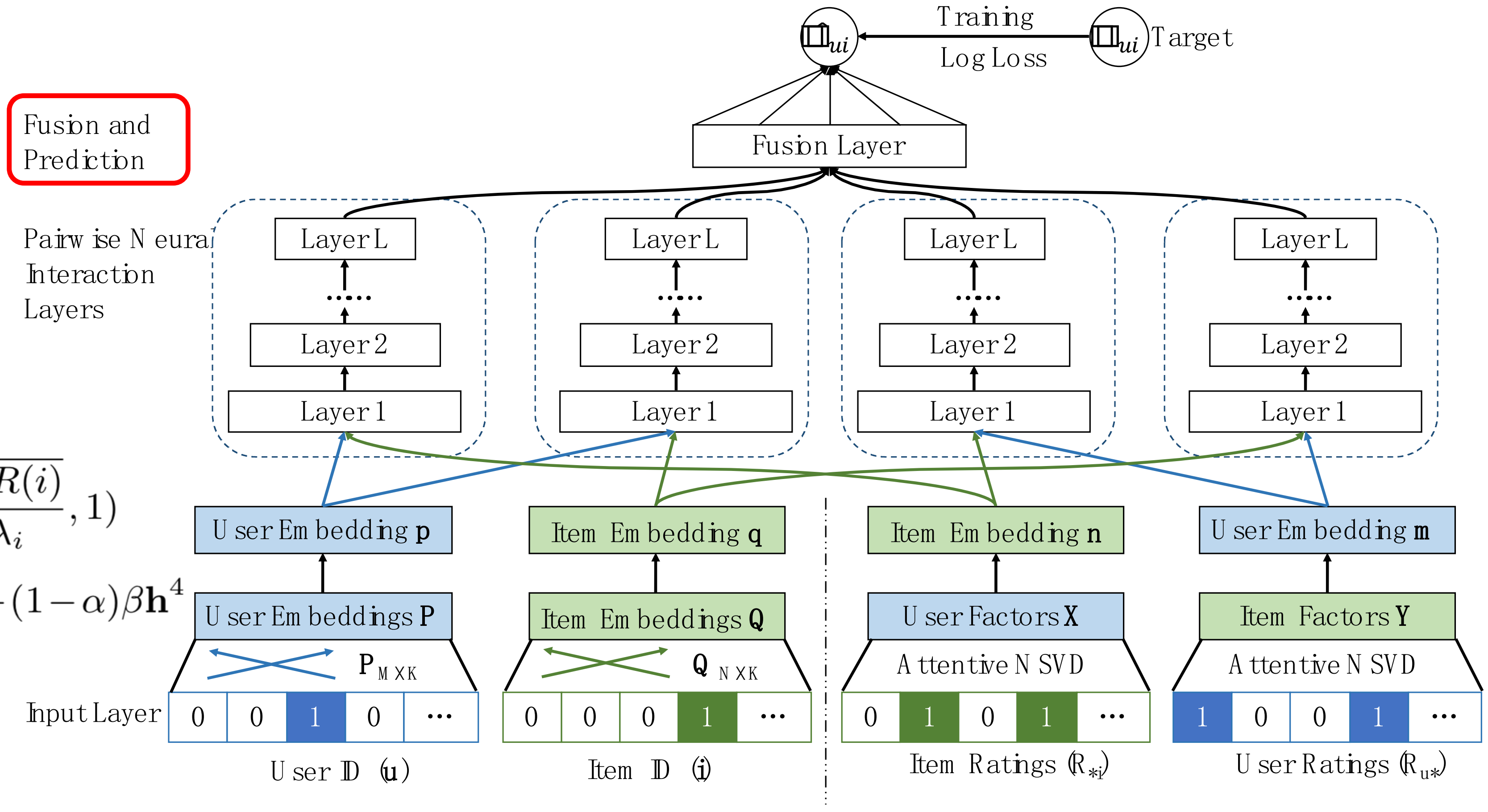
Fusion with empirical formula:

$$\alpha = \min(\frac{\sqrt{R(u)}}{\lambda_u}, 1), \quad \beta = \min(\frac{\sqrt{R(i)}}{\lambda_i}, 1)$$

$$\mathbf{h}_f = \alpha(1-\beta)\mathbf{h}^1 + \alpha\beta\mathbf{h}^2 + (1-\alpha)(1-\beta)\mathbf{h}^3 + (1-\alpha)\beta\mathbf{h}^4$$

Prediction:

$$\hat{R}_{ui} = \delta_p(\mathbf{W}_p \mathbf{h}_f + b_f)$$



Experimental Results

- **Datasets**: Movielens 1M & Amazon Music (transferred to implicit)
- **Evaluation Protocol**: leave-one-out
- **Metrics**: Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG)
- **Compared Methods**:
    - ItemPop
    - eALS [He *et al.*, 2016]
    - BPR [Rendle *et al.*, 2009]
    - MLP [He *et al.*, 2017]
    - NeuMF [He *et al.*, 2017]
    - DMF [Xue *et al.*, 2017]

Experimental Results

DELF methods achieve the best overall performance on both datasets.

(Similar performance to NeuMF in Movielens, and >10% improvements to all baselines in Amazon.)

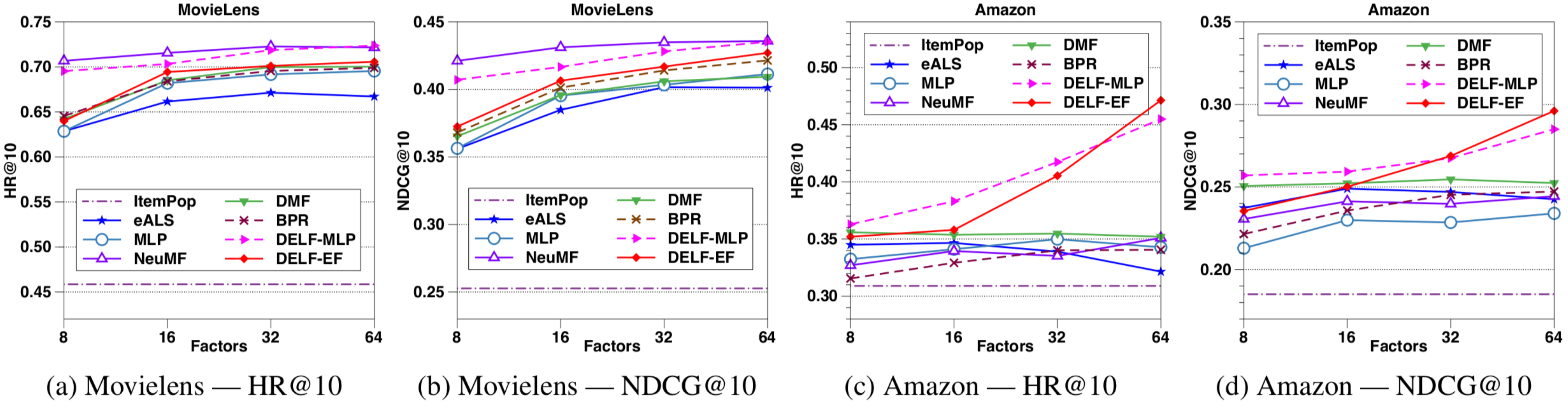


Fig2: Performance of HR@10 and NDCG@10 w.r.t. the number of predictive factors on the two datasets

Experimental Results

The key components in DELF are useful for improving recommendation results
(i.e., attentive module, pairwise neural interaction layers)

Factors	DELF-MLP	DELF-EF	DELF-ni	DELF-na
Movielens				
8	**0.695**	0.640	0.672	0.680
16	**0.703**	0.694	0.693	0.688
32	**0.719**	0.701	0.704	0.705
64	**0.724**	0.706	0.713	0.709
Amazon				
8	**0.363**	0.352	0.356	0.355
16	**0.383**	0.358	0.362	0.374
32	**0.417**	0.405	0.390	0.409
64	0.455	**0.471**	0.423	0.442

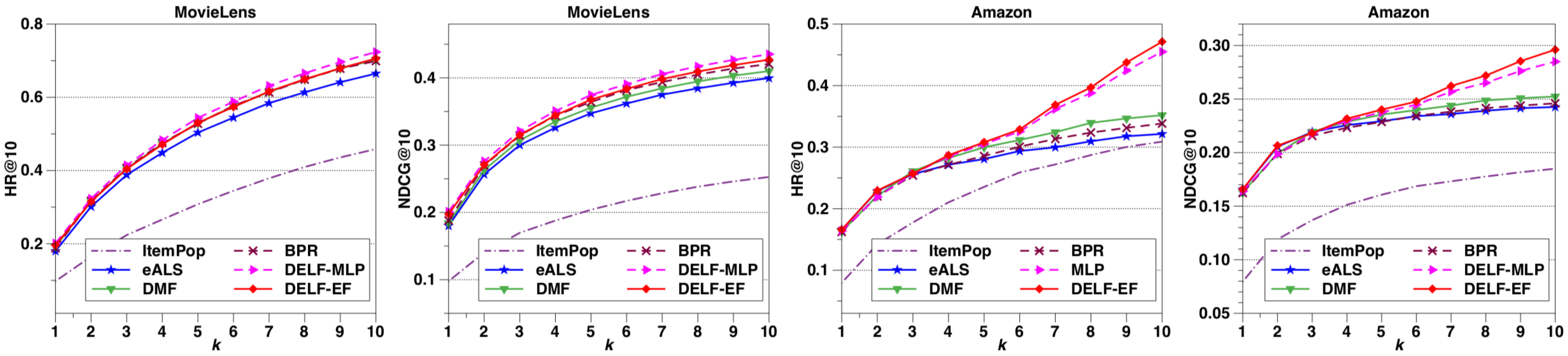
Table 2: HR@10 of different variants of DELF

Factors	DELF-MLP	DELF-EF	DELF-ni	DELF-na
Movielens				
8	**0.407**	0.373	0.389	0.398
16	**0.417**	0.407	0.409	0.407
32	**0.428**	0.417	0.416	0.419
64	**0.435**	0.424	0.426	0.426
Amazon				
8	**0.257**	0.235	0.252	0.250
16	**0.259**	0.250	0.247	0.249
32	0.268	**0.269**	0.259	0.258
64	0.285	**0.296**	0.264	0.277

Table 3: NDCG@10 of different variants of DELF

Experimental Results

DELF keeps consistent improvements over other methods with all the values of k (1~10) for Top-k recommendation.



(a) Movielens — HR@k (b) Movielens — NDCG@k (c) Amazon — HR@k (d) Amazon — NDCG@k

Figure 3: Evaluation of Top-k item recommendation where k ranges from 1 to 10 on the two datasets

Conclusion

- In addition to the primary embeddings, we propose to obtain additional embeddings for users and items based on their interaction vectors with an attentive neural method.
- We introduce a neural network architecture to learn deep representations for pairwise interactions among dual user/item embeddings.
- Extensive experiments on two real-world datasets demonstrate the superior performance of our proposed method.

Thanks for your attention!

Q&A

Contact: weiyu_cheng@sjtu.edu.cn