When Multi-Level Meets Multi-Interest: A Multi-Grained Neural Model for Sequential Recommendation

Yu Tian[1], Jianxin Chang[2], Yanan Niu[2], Yang Song[2], Chenliang Li[1†]
[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, 430072, China
s.braylon1002@gmail.com; cllee@whu.edu.cn
[2]Kuaishou Technology Co., Ltd., Beijing, 10010, China
{changjianxin,niuyanan,yangsong}@kuaishou.com

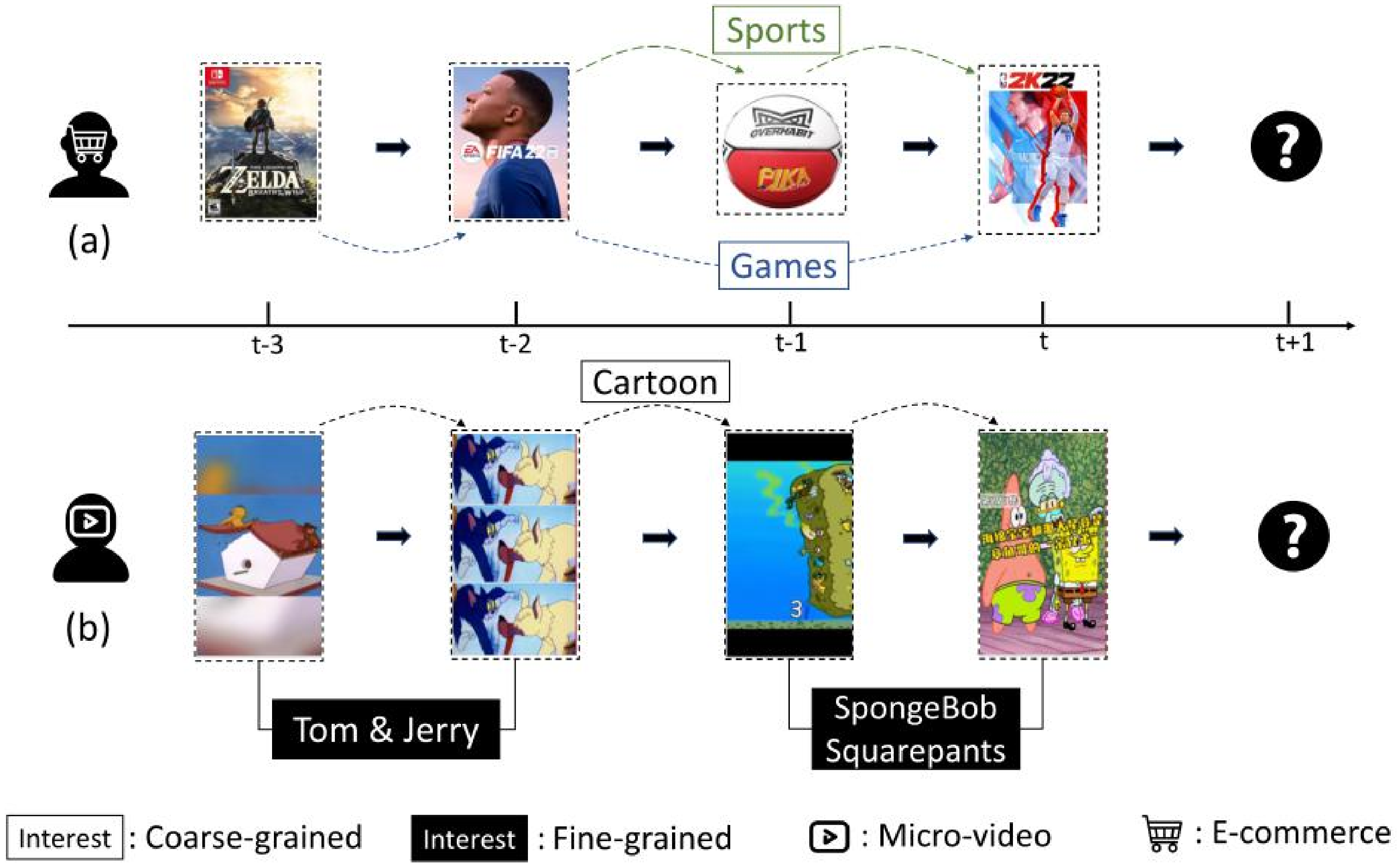
SIGIR 2022

2022. 5. 7 • ChongQing

gesis
Leibniz-Institut für Sozialwissenschaften

Reported by Gu Tang

Introduction



Method

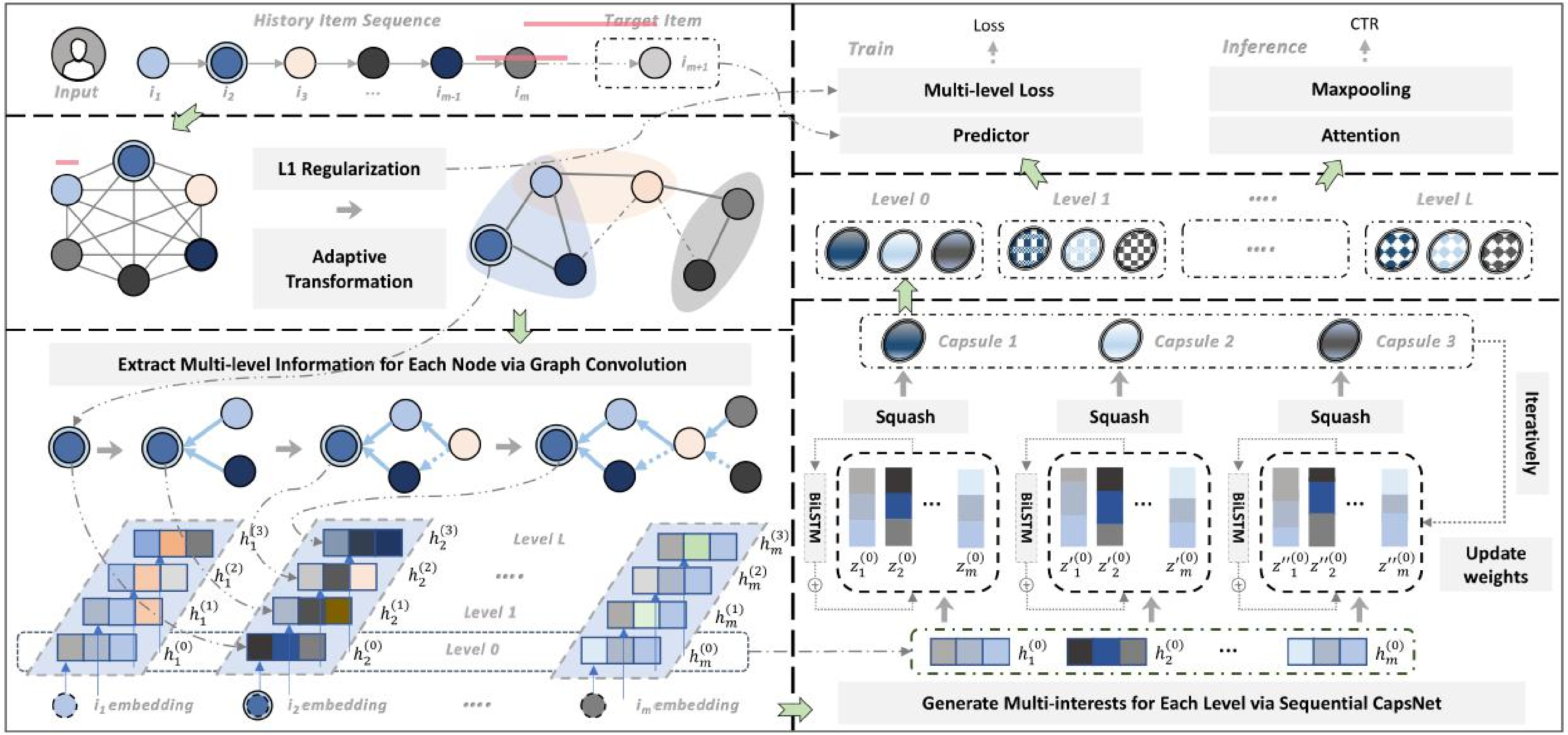
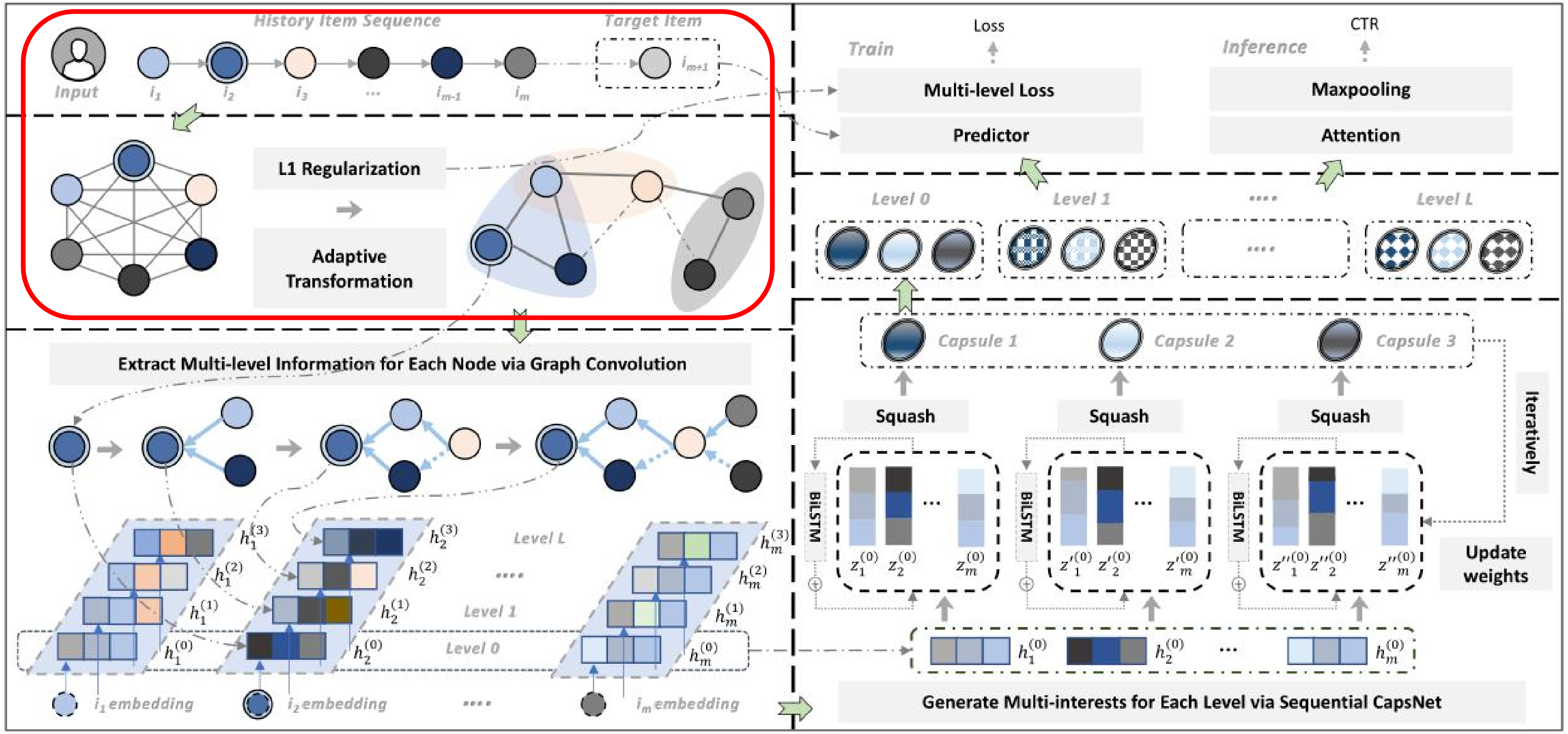


Figure 2: The network architecture of our proposed MGNM. The raw sequence is the historical behavior of a user. By transforming the original sequence into a user-aware adaptive graph and using the neural aggregation function of sequential CapsNet, the timing information is added to the graph in the training process. In the inference stage of the model, the max-pooling layer is used to obtain the final prediction score.

Method



3.2.1 Embedding Layer. In the embedding layer, we firstly form a user embedding table $U \in R^{N \times d}$ and an item embedding table $V \in R^{M \times d}$, where d denotes the dimension of the embedding vector. For the given user u and the associated behavior sequence b_u, we can perform the table lookup from U and V to obtain the corresponding user and item embedding representations $\mathbf{x}_u$ and $[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m]$ respectively. Hence, the user embeddings U are expected to encode the users' overall preference, while the item embeddings V reflect items' characteristics in this space instead.

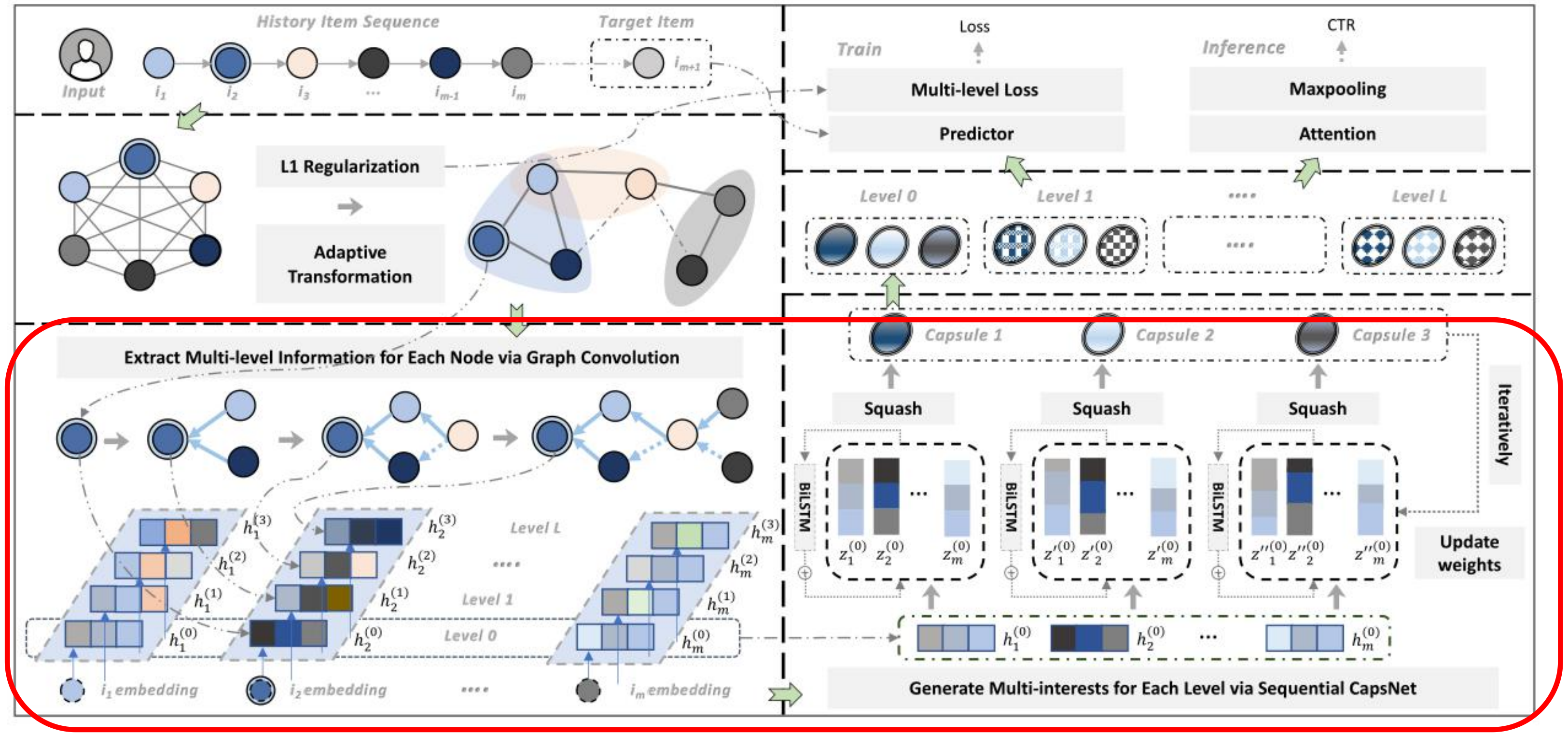
$$\mathrm{A}_{i,j} = sigmoid((\mathbf{x}_i \odot \mathbf{x}_j) \cdot \mathbf{x}_u), \tag{1}$$

$$\mathrm{H}^{(l+1)} = \delta(\tilde{\mathrm{D}}^{-\frac{1}{2}} \tilde{\mathrm{A}} \tilde{\mathrm{D}}^{-\frac{1}{2}} \mathrm{H}^{(l)} \mathrm{W}), \tag{2}$$

$$\tilde{\mathrm{D}}^{-\frac{1}{2}} = \mathrm{I} + \mathrm{D}^{-\frac{1}{2}} \mathrm{A} \mathrm{D}^{-\frac{1}{2}}, \tag{3}$$

$$\mathrm{H}^{(0)} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m], \tag{4}$$

Method



$$\mathbf{Z}_i = \mathbf{H}^{(l)}\mathbf{W}_i, \tag{5}$$

$$\mathbf{c} = softmax(\mathbf{g}). \tag{6}$$

$$\mathbf{o}_i = \frac{\|\mathbf{v}_i\|^2}{\|\mathbf{v}_i\|^2 + 1}\frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}, \tag{7}$$

$$\mathbf{v}_i = \sum_{j=1}^{m} c_j \mathbf{z}_j^{(l)}, \tag{8}$$

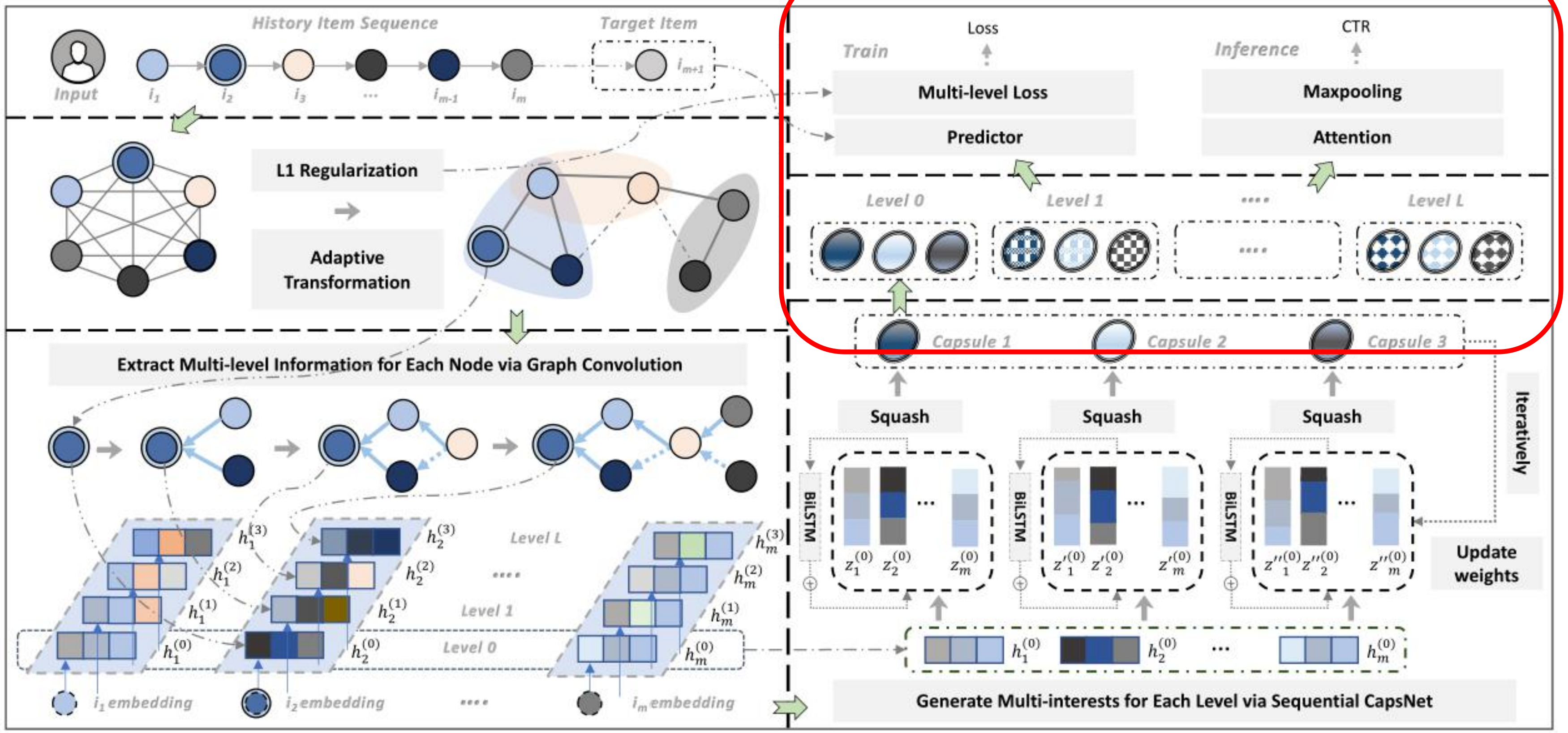
$$g_i = g_i + \mathbf{o}_i^{\top}\mathbf{z}_i. \tag{9}$$

capsule

$$\mathbf{Z}_i = \mathbf{Z}_i + BiLSTM(\mathbf{Z}_i). \tag{10}$$

$$\mathbf{q}_i^{(l)} = ReLU(\mathbf{o}_i\mathbf{W}_i'), \tag{11}$$

Method



$$\mathbf{p}_u^{(l)} = \sum_{j=1}^{K} a_j \mathbf{q}_j^{(l)}, \tag{12}$$

$$a_j = \frac{\exp(\mathbf{q}_j^{(l)\top}\mathbf{x}_t)}{\sum_{k=1}^{K}\exp(\mathbf{q}_k^{(l)\top}\mathbf{x}_t)}, \tag{13}$$

$$\hat{y}_{u,i}^{(l)} = \mathbf{p}_u^{(l)\top}\mathbf{x}_t, \tag{14}$$

$$\hat{y}_{u,i} = \max(\hat{y}_{u,i}^{(0)}, \cdots, \hat{y}_{u,i}^{(L)}). \tag{15}$$

$$\mathcal{L}_{all} = \sum_{l=0}^{L}\mathcal{L}_l + \theta_1\mathcal{L}_1 + \theta_2\mathcal{L}_2, \tag{16}$$

$$\mathcal{L}_l = -\sum_{u,i}[y_{u,i}ln(\hat{y}_{u,i}^{(l)}) + (1 - y_{u,i})ln(1 - \hat{u}_{u,i}^{(l)})], \tag{17}$$

Method

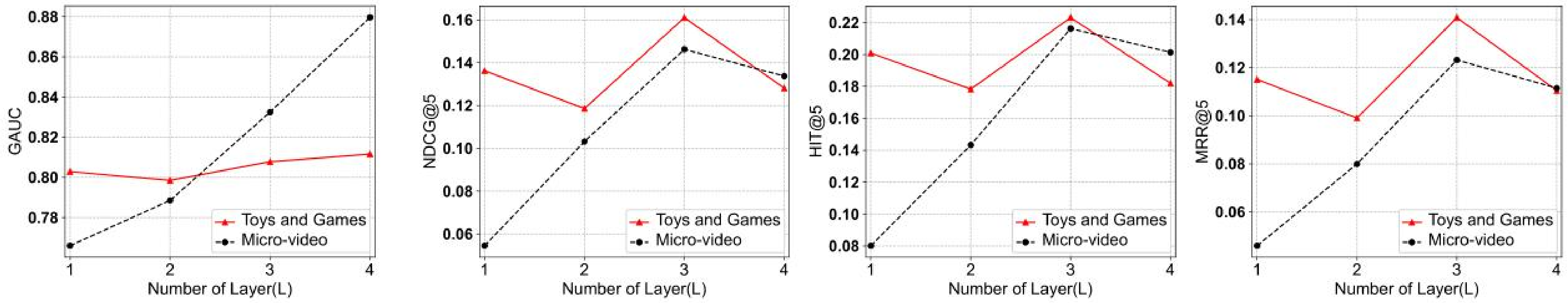
Table 1: Statistics of the three datasets.

Datasets	#Users	#Items	#Interactions
Micro-video	60,813	292,286	14,952,659
Musical Instruments	60,739	56,301	946,627
Toys and Games	313,557	241,657	6,212,901

Table 2: Performance comparison of different methods across the three datasets. The best and second-best results are highlighted in boldface and underlined respectively. $*$ indicates that the performance difference against the best result is statistically significant at 0.05 level. Note that TGSRec took too long to train hence has no results on the large Micro-video dataset See context for details.

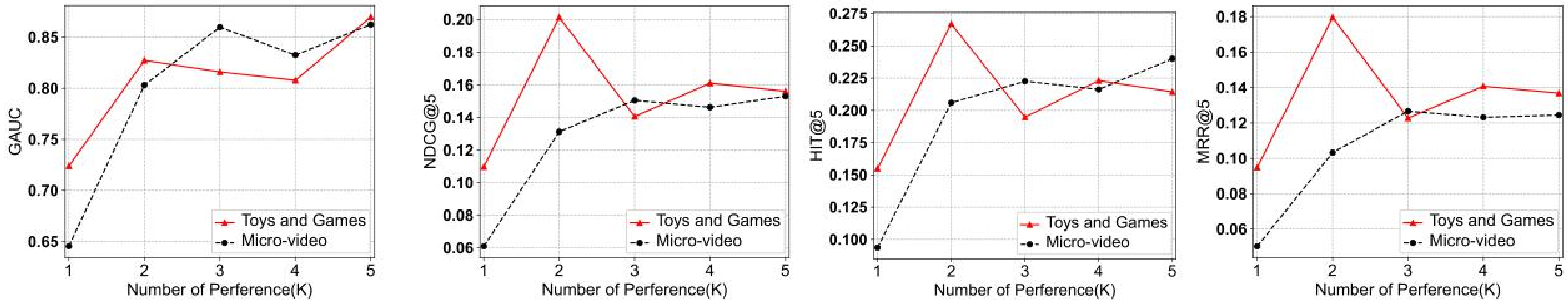
Method	Micro-video				Toys and Games				Music Instruments			
	GAUC	NDCG@5	HIT@5	MRR@5	GAUC	NDCG@5	HIT@5	MRR@5	GAUC	NDCG@5	HIT@5	MRR@5
Caser	0.6917*	0.0964*	0.1417*	0.0815*	0.6234*	0.0679*	0.1012*	0.0569*	0.6763*	0.0955*	0.1178*	0.0883*
A2svd	0.6808*	0.0443*	0.0686*	0.0364*	0.6846*	0.0507*	0.0739*	0.0430*	0.6652*	0.0956*	0.1368*	0.0820*
GRU4Rec	0.6944*	0.0702*	0.1050*	0.0589*	0.6624*	0.0840*	0.1278*	0.0697*	0.6498*	0.0619*	0.1049*	0.0478*
SLi_rec	0.6903*	0.0948*	0.1390*	0.0802*	0.7847*	0.0932*	0.1327*	0.0803*	0.6912*	**0.1078**	0.1507*	0.0937*
TGSRec	–	–	–	–	0.7915*	0.1410*	0.2027*	0.1164*	**0.7759**	0.0946*	0.1653	0.0729*
MIMN	0.7387*	0.1151*	0.1683*	0.0977*	0.7224*	0.1158*	0.1676*	0.0988*	0.6787*	0.0955*	0.1509*	0.0750*
MIND	0.6778*	0.08582*	0.1367*	0.0700*	0.6611*	0.1015*	0.1510*	0.0824*	0.6588*	0.1040*	0.1422*	0.0898*
ComiRec-DR	0.7028*	0.0863*	0.1307*	0.0718*	0.6681*	0.1131*	0.1597*	0.0978*	0.6647*	0.1091*	0.1541*	0.0943*
ComiRec-SA	0.6249*	0.0354*	0.0577*	0.0281*	0.6486*	0.0665*	0.0977*	0.0563*	0.6559*	0.0820*	0.1204*	0.0694*
SURGE	0.8116*	0.1091*	0.1728*	0.0883*	0.7863*	0.0930*	0.1353*	0.0791*	0.6902*	0.1056*	0.1494*	0.0913*
MGNM	**0.8325**	**0.1463**	**0.2163**	**0.1232**	**0.8078**	**0.1611**	**0.2231**	**0.1408**	0.7480*	0.1057	**0.1658**	**0.1021**

Experiment



(a) GAUC (b) NDCG@5 (c) HIT@5 (d) MRR@5

Figure 3: The performance of different L values on Toys and Games and Micro-video Datasets.



(a) GAUC (b) NDCG@5 (c) HIT@5 (d) MRR@5

Figure 4: The performance of different K values on Toys and Games and Micro-video Datasets.

Experiment

Table 3: The ablation study of MGNM on Toys and Games Dataset. The best results are highlighted in boldface.

Model	Toys and Games			
	GAUC	NDCG@5	HIT@5	MRR@5
w/o UGCN	0.7499	0.0929	0.1325	0.0799
w/o $L1$Norm	0.7757	0.1306	0.1848	0.1128
w/o BiLSTM	0.6743	0.1205	0.1689	0.1046
w/o MaxPool	0.8491	0.0980	0.1430	0.0832
SCN→ BiLSTM	0.6589	0.0838	0.1223	0.0712
SCN→ SumPool	0.6651	0.0846	0.1232	0.0720
SCN→ SelfAtt	0.6724	0.0791	0.1148	0.0674
SCN (Transformer)	0.6663	0.0923	0.1321	0.0792
MGNM	**0.8078**	**0.1611**	**0.2231**	**0.1408**

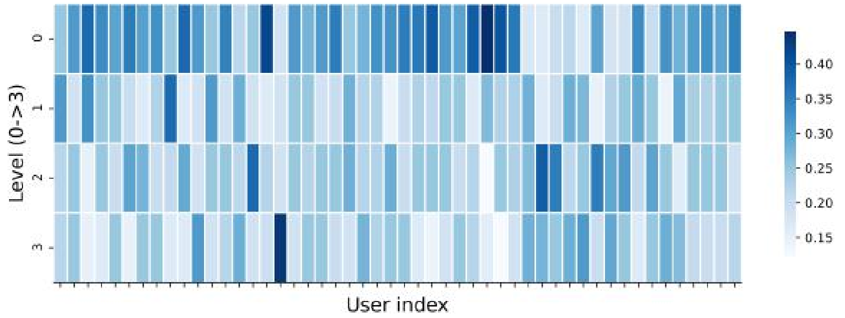


Figure 5: Visualization of multi-level user interest distribution on Micro-video dataset (*Best viewed in color*).

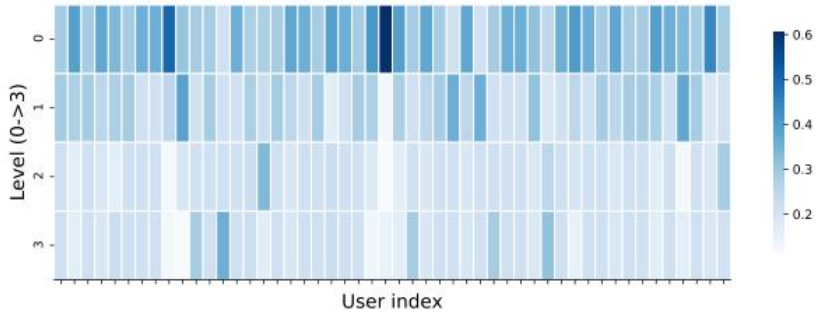


Figure 6: Visualization of multi-level user interest distribution on Toys and Games dataset (*Best viewed in color*).

Experiment

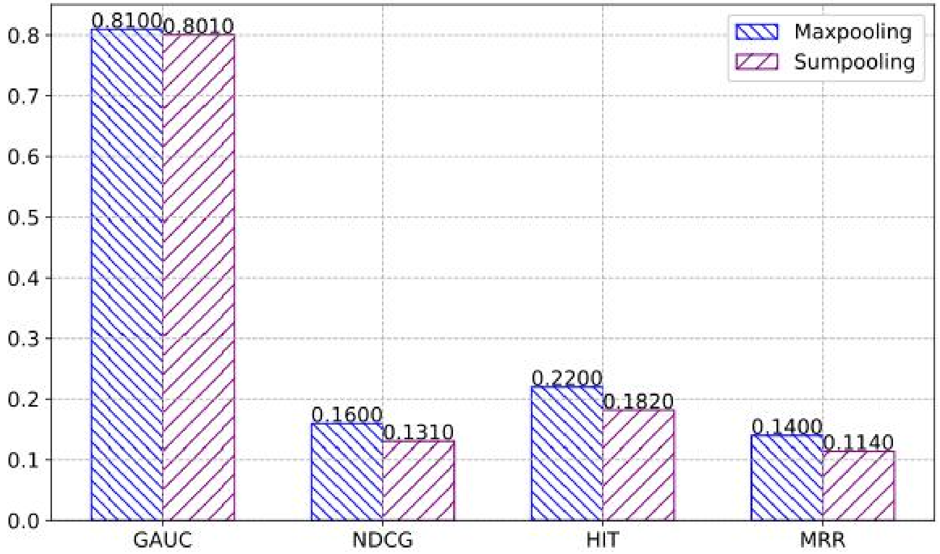


Figure 7: Max-pooling vs. sum-pooling for MGNM in the inference stage.

Table 4: Runtime comparisons for different datasets.

Datasets	Per Iteration (s)	Iterations	Total Time (m)
Micro-video	0.3825	15,311	97.60
Toys and Games	0.1843	13,202	40.55
Music Instruments	0.0598	2,373	2.37

Time Complexity Analysis. Table 4 reports the runtime of MGNM training procedure for a single user on different datasets by using a single GPU. Although the MGNM adopt the graph convolution, we can see that the model training with 15M interactinos takes about 1.5H for one epoch, which is computationally efficient.

Thanks