



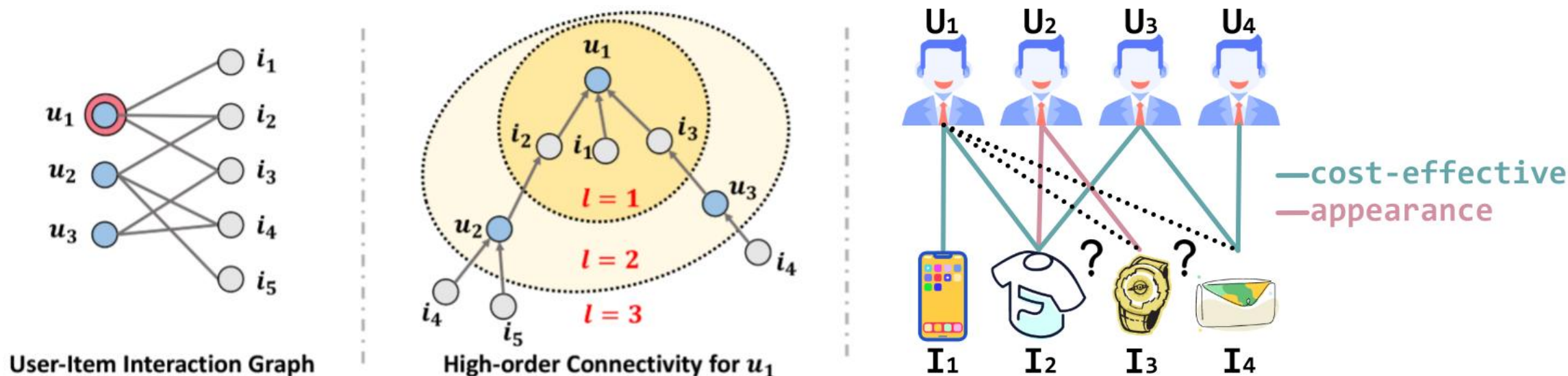
# Multi-Component Graph Convolutional Collaborative Filtering

**Xiao Wang<sup>1</sup>, Ruijia Wang<sup>1</sup>, Chuan Shi<sup>1\*</sup>, Guojie Song<sup>2</sup>, Qingyong Li<sup>3</sup>**

<sup>1</sup>Beijing University of Posts and Telecommunications, <sup>2</sup>Peking University, <sup>3</sup>Beijing Jiaotong University  
{xiaowang, wangruijia, shichuan}@bupt.edu.cn, gjsong@pku.edu.cn, qingyongli@gmail.com

AAAI2020

- 之前的方法关注高阶连接性：**过去有相似购买行为的用户在未来也会有相似的选择**
- 然而现在**用户与物品之间的交互通常是由多种复杂的潜在购买动机驱动的**



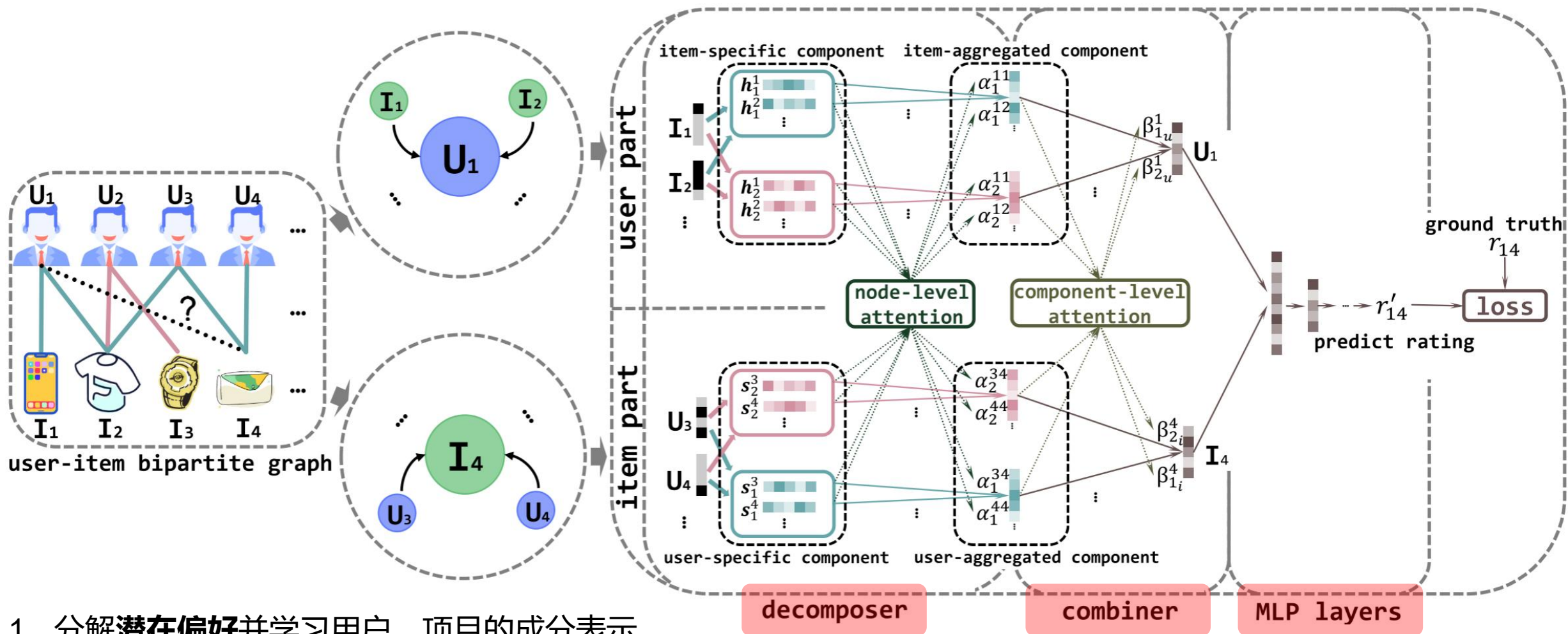
协同过滤中传统矩阵分解和基于深度学习、图的方法仍存在**无法区分不同的购买动机**的问题。图神经网络也需要进一步探索如何捕捉用户和物品之间复杂关系，**捕捉到细粒度的用户偏好**。

# 多组件图卷积协同过滤方法 (MCCF)



南京航空航天大学

Nanjing University of Aeronautics and Astronautics



1. 分解**潜在偏好**并学习用户、项目的成分表示
2. 将学习到的内容利用多组件神经网络**加权融合**
3. MLP+Predict：进行**评分预测**

# 分解器 二部图&多成分提取



$\mathcal{G} = \{U, I, R, E\}$  用户; 物品; 评分; 边

$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{N_u}] \in \mathbb{R}^{L_u \times N_u}$  特征数\*用户数

$\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{N_i}] \in \mathbb{R}^{L_i \times N_i}$

假设M个潜在组件捕获用户-项目交互动机

对于用户而言  $\mathbf{W} = \{\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_M\}$

对于物品而言  $\mathbf{Q} = \{\mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_M\}$

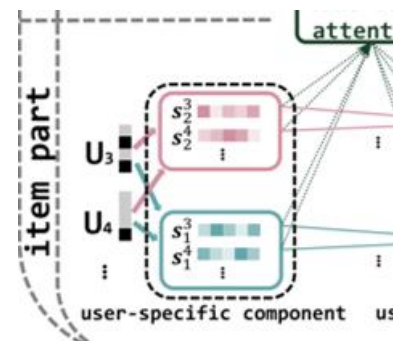
对于物品i, 第m个特定组件表示为

$$\mathbf{h}_m^i = \mathbf{Q}_m \mathbf{p}_i.$$

对于用户u, 第m个特定组件表示为

$$\mathbf{s}_m^u = \mathbf{W}_m \mathbf{u}_u.$$

$\mathbf{h}_m^i$  可以理解为物品i的  
第m种潜在成分表示



# 分解器 节点级注意



$$\{\mathbf{s}_m^u\}_{m=1}^M$$

$$\{\mathbf{h}_m^i\}_{m=1}^M \quad i \in \mathcal{P}_u$$

$$e_m^{ui} = \text{att}_{\text{node}}(\mathbf{s}_m^u, \mathbf{h}_m^i; m)$$

基于第m个组件的情况下用户U与物品i交互的注意力权重  $e_m^{ui}$

$$\alpha_m^{ui} = \text{softmax}(e_m^{ui}) = \frac{\exp(\sigma(\mathbf{a}_m^T \cdot [\mathbf{s}_m^u \parallel \mathbf{h}_m^i]))}{\sum_{i \in \mathcal{P}_u} \exp(\sigma(\mathbf{a}_m^T \cdot [\mathbf{s}_m^u \parallel \mathbf{h}_m^i]))}$$

对得到的可能性进行softmax归一化;  
沿用了graph attention network (GAT) 的做法

$$\mathbf{z}_m^u = \sigma \left( \sum_{i \in \mathcal{P}_u} \alpha_m^{ui} \cdot \mathbf{h}_m^i \right)$$

$$\{\mathbf{z}_m^u\}_{m=1}^M$$

第m个组件下用户u的聚合表示  $\mathbf{z}_m^u$   
得到了用户在特定组件下的交互偏好

$z_1^u$  表示用户  $u$  对价格的敏感程度  
 $z_2^u$  表示用户  $u$  对某些品牌的偏好  
 $z_3^u$  表示用户  $u$  对产品功能的需求

由分解器生成的用户的  
最终物品聚合组件

每个组件对于用户u的最终嵌入的贡献程度

上面得到的每个组件对于用户u的最终嵌入

$$(\beta_1^u, \beta_2^u, \dots, \beta_M^u) = att_{com}(\mathbf{z}_1^u, \mathbf{z}_2^u, \dots, \mathbf{z}_M^u)$$

执行组件级的深度神经网络

$$\mathbf{d}_m^u = \sigma(\mathbf{C}_m \cdot [\mathbf{z}_m^u \parallel \mathbf{s}_m^u] + \mathbf{b}_m)$$

考虑每个用户的特定信息，将用户最终聚合组件和用户特定组件进行级联拼接，经过一个全连接层计算用户u在第m个特定组件下的（综合行为模式）统一嵌入表示  $\mathbf{d}_m^u$

$$w_m = \sigma(\mathbf{q}^T \cdot \mathbf{d}_m^u + b)$$

将上面统一嵌入表示与组件级注意向量q点积计算后加上偏置向量q，通过激活函数得到**组件m对用户u的重要性得分**  $w_m$   
例如用户在意性价比，那么性价比对应得分就会更高

$$\beta_m^u = \frac{\exp(w_m)}{\sum_{k=1}^M \exp(w_k)}$$

根据之前计算的重要性得分学习不同组件的注意力权重

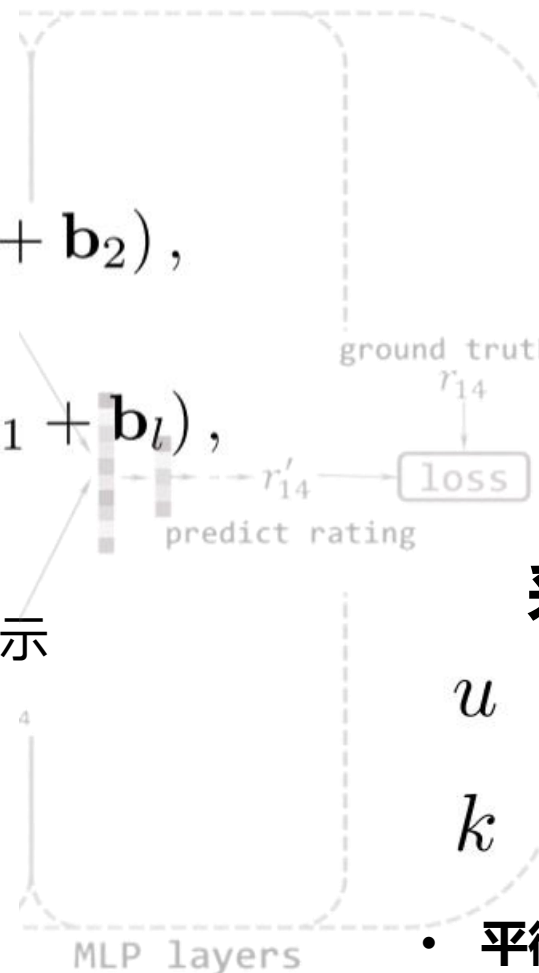
$$\mathbf{z}_u = \sum_{m=1}^M \beta_m^u \cdot \mathbf{z}_m^u$$

结合用户u在多个组件下的最终的偏好表示

## MLP预测

$$\begin{aligned}
 \mathbf{g}_1 &= [\mathbf{z}_u \parallel \mathbf{v}_i], \\
 \mathbf{g}_2 &= \sigma(\mathbf{W}_2 \cdot \mathbf{g}_1 + \mathbf{b}_2), \\
 &\dots \\
 \mathbf{g}_{l-1} &= \sigma(\mathbf{W}_l \cdot \mathbf{g}_{l-1} + \mathbf{b}_l), \\
 r'_{ui} &= \mathbf{w}^T \cdot \mathbf{g}_{l-1},
 \end{aligned}$$

通过用户和物品的最终嵌入表示  
用户u对物品i的交互预测



## 损失函数

$$\mathcal{L}_r = \frac{1}{2|\mathcal{O}|} \sum_{(u,i) \in \mathcal{O}} (r'_{ui} - r_{ui})^2$$

$$\min_{\Theta} \mathcal{L} = \mathcal{L}_r + \lambda \|\theta\|_0$$

稀疏正则化, 减少不必要的参数, 防止过拟合, 提升泛化能力

## 采样策略

$$u \sim \mathcal{U}(0, 1),$$

$$k = u^{\frac{1}{r}},$$

- u代表0-1之间的随机数
- r代表节点间的交互数
- ➔ 高交互r大, r 倒数减小, k 增大

- **平衡邻居数量**: 缓解数据量大导致计算量大
- **关注高交互邻居**: 高交互的用户-物品更能反映真实喜好

## 数据集:

1. **MovieLens-100k**: 包含 943 名用户对 1,682 部电影的 **100,000** 条评分。
2. **Amazon**: 包含 1,000 名用户对 1,000 种商品的 **65,170** 条评分。
3. **Yelp**: 包含 1,286 名用户对 2,614 家本地商家的 **30,838** 条评分。
4. **合成数据集**: 用于验证多组件机制的有效性。

## 基线方法:

1. **矩阵分解**方法: PMF、BiasedMF、LLORMA-Local。
2. **自编码器**方法: AutoRec、CF-NADE。
3. **图卷积网络**方法: GC-MC。
4. **MCCF 变体**: MCCF-nd (去掉节点级注意力)、MCCF-cmp (去掉组件级注意力)。

## 评估指标:

1. **RMSE** (均方根误差): 衡量预测评分与真实评分之间的差异。
2. **MAE** (平均绝对误差): 衡量预测评分与真实评分之间的平均绝对差异。

Table 1: Performance comparison of rating prediction. The smaller values, the better performance.

Models		PMF	BiasMF	LLORMA-Local	I-AUTOREC	I-CF-NADE	GC-MC	MCCF- <i>nd</i>	MCCF- <i>cmp</i>	MCCF
Yelp	RMSE	0.3967	0.3902	0.3890	0.3817	0.3857	0.3850	0.3836	<b>0.3806</b>	<b>0.3806</b>
	MAE	0.1571	0.1616	0.1547	0.1201	0.1427	0.1354	0.1286	<b>0.1029</b>	<b>0.1029</b>
Amazon	RMSE	0.9339	0.9028	0.9019	0.9213	0.8987	0.8946	0.8942	0.8919	<b>0.8876</b>
	MAE	0.7113	0.6759	0.6725	0.7064	0.6565	0.6619	0.6595	0.6483	<b>0.6428</b>
Movielens	RMSE	0.9638	0.9257	0.9313	0.9435	0.9229	0.9145	0.9203	0.9142	<b>0.9070</b>
	MAE	0.7559	0.7258	0.7286	0.7370	0.7168	0.7165	0.7160	0.7081	<b>0.7050</b>

多组件机制性  
能在组件数量  
为3的时候达  
到最佳

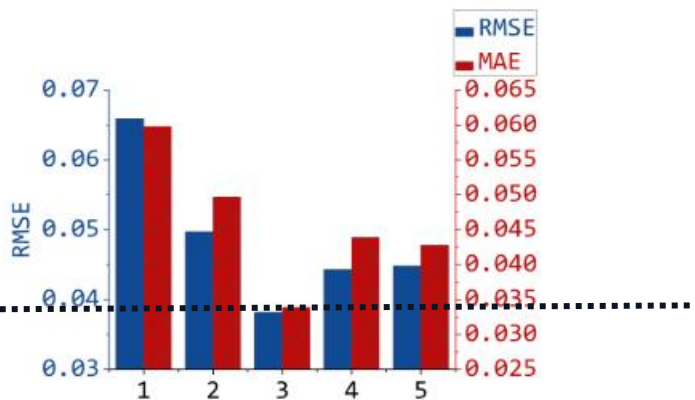
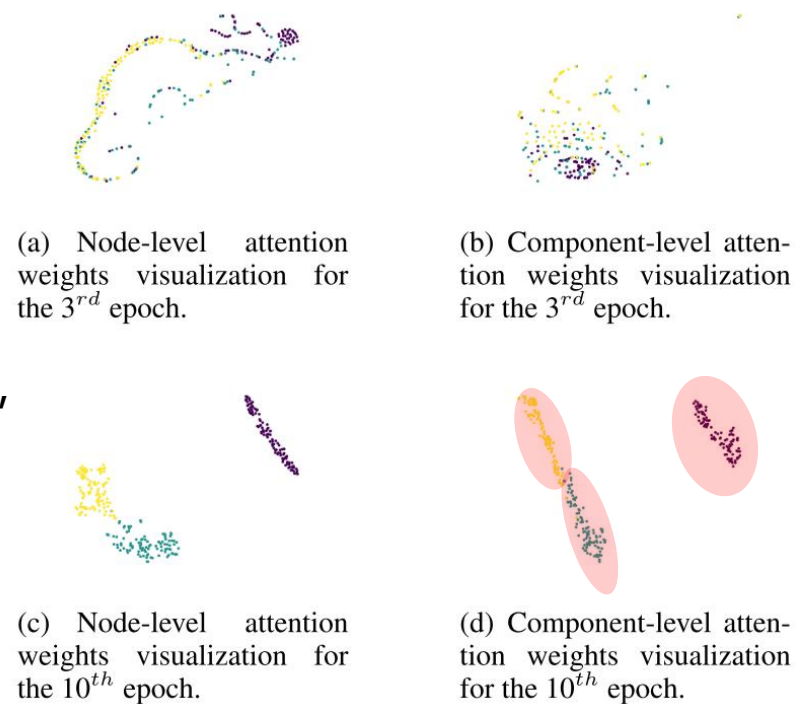


Figure 3: RMSE and MAE on synthetic user-item bipartite graph generated with three latent components.

训练中模型可以有效捕捉用户和物品之间细粒度的关系, 识别用户的潜在购买动机



组件数量K

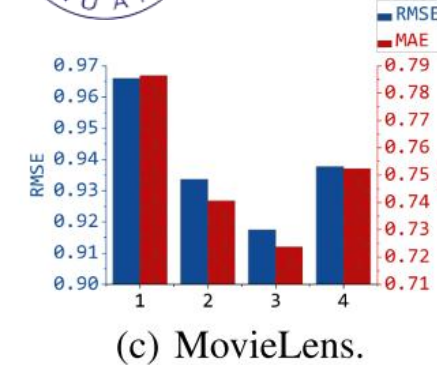
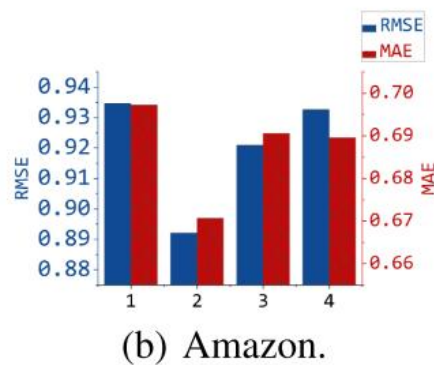
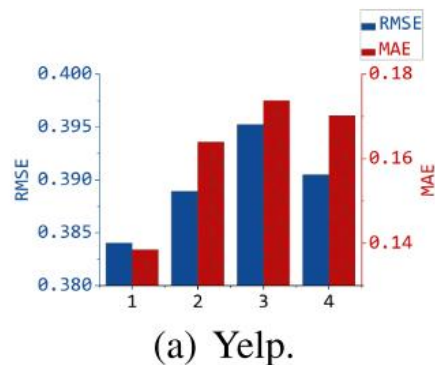


Figure 5: Impact of latent components numbers on three real datasets.

嵌入维度 d

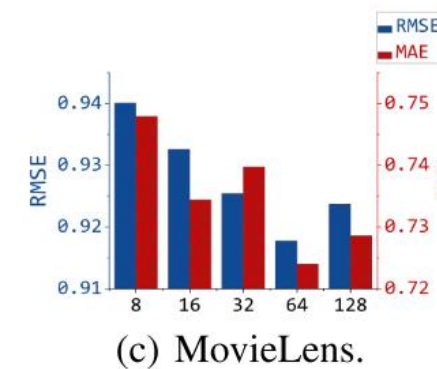
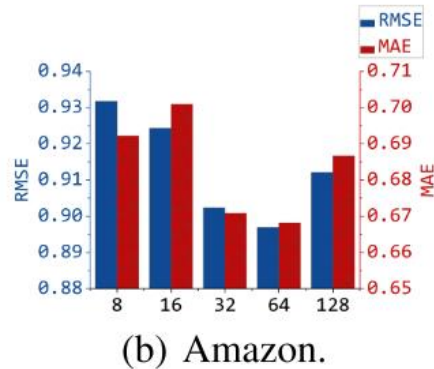
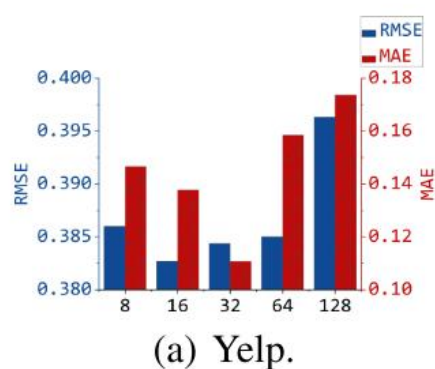
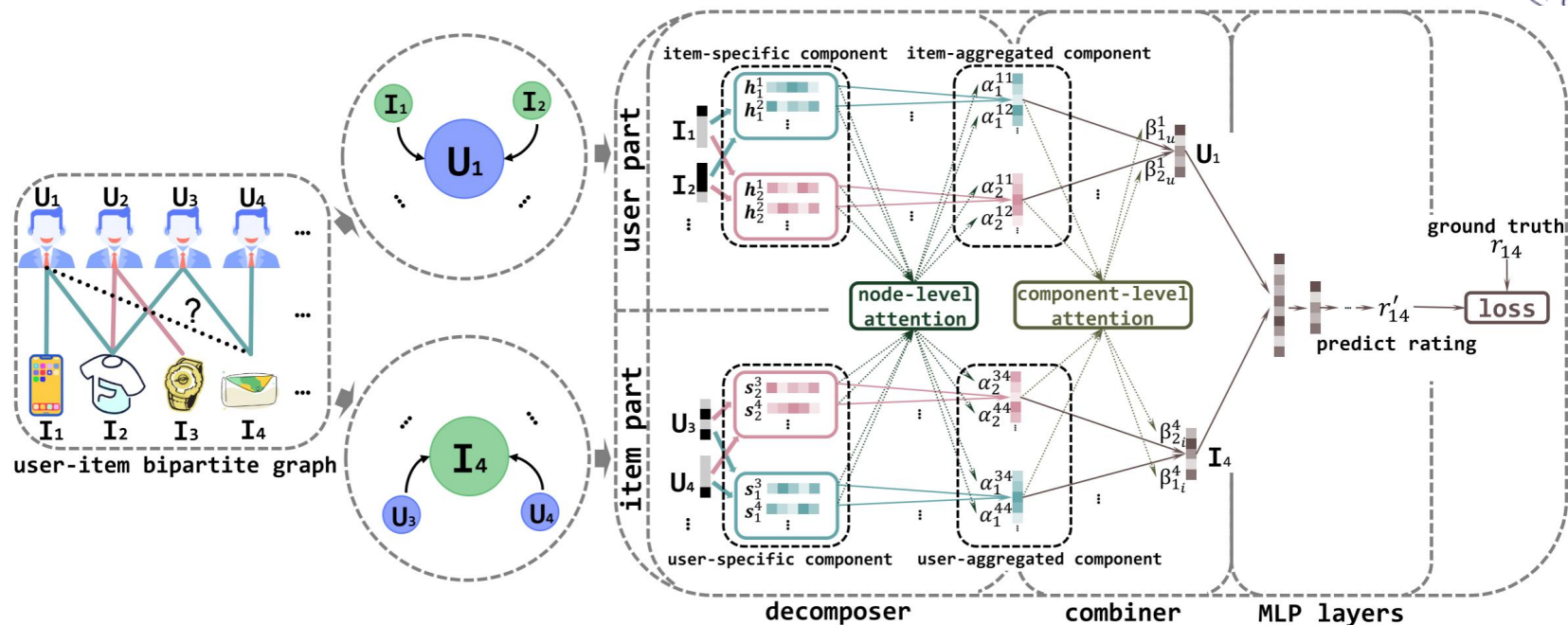


Figure 6: Impact of embedding dimensions on three real datasets.

超参数K、d 根据数据集的特点进行调优



- 论文主要关注用户和物品之间的高交互部分（即热门物品和活跃用户），而对低交互部分（即长尾物品和不活跃用户）的关注较少，这可能导致长尾物品和不活跃用户得不到足够的推荐机会，从而产生**长尾效应**。
- 借鉴论文中的**多组件机制**，进一步研究如何通过调整组件权重或引入公平性约束，来平衡热门物品和长尾物品的推荐机会。



南京航空航天大学  
Nanjing University of Aeronautics and Astronautics

**Thanks**

---