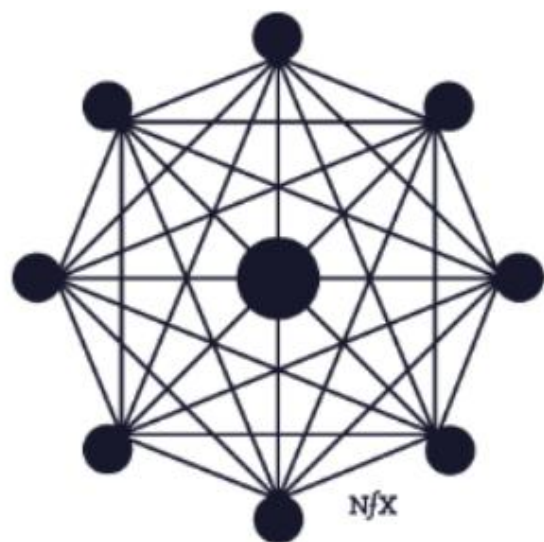


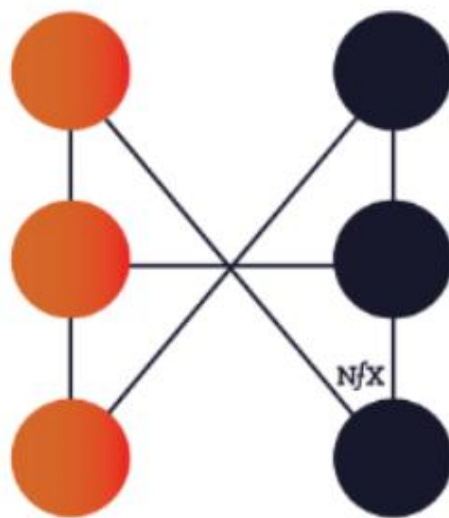
异构图

✓ 不是同构图的就是异构图了

✎ 点的类型多种，边的类型也多种(反正只要有一个多种的就算异构图了)



Homogeneous
Network

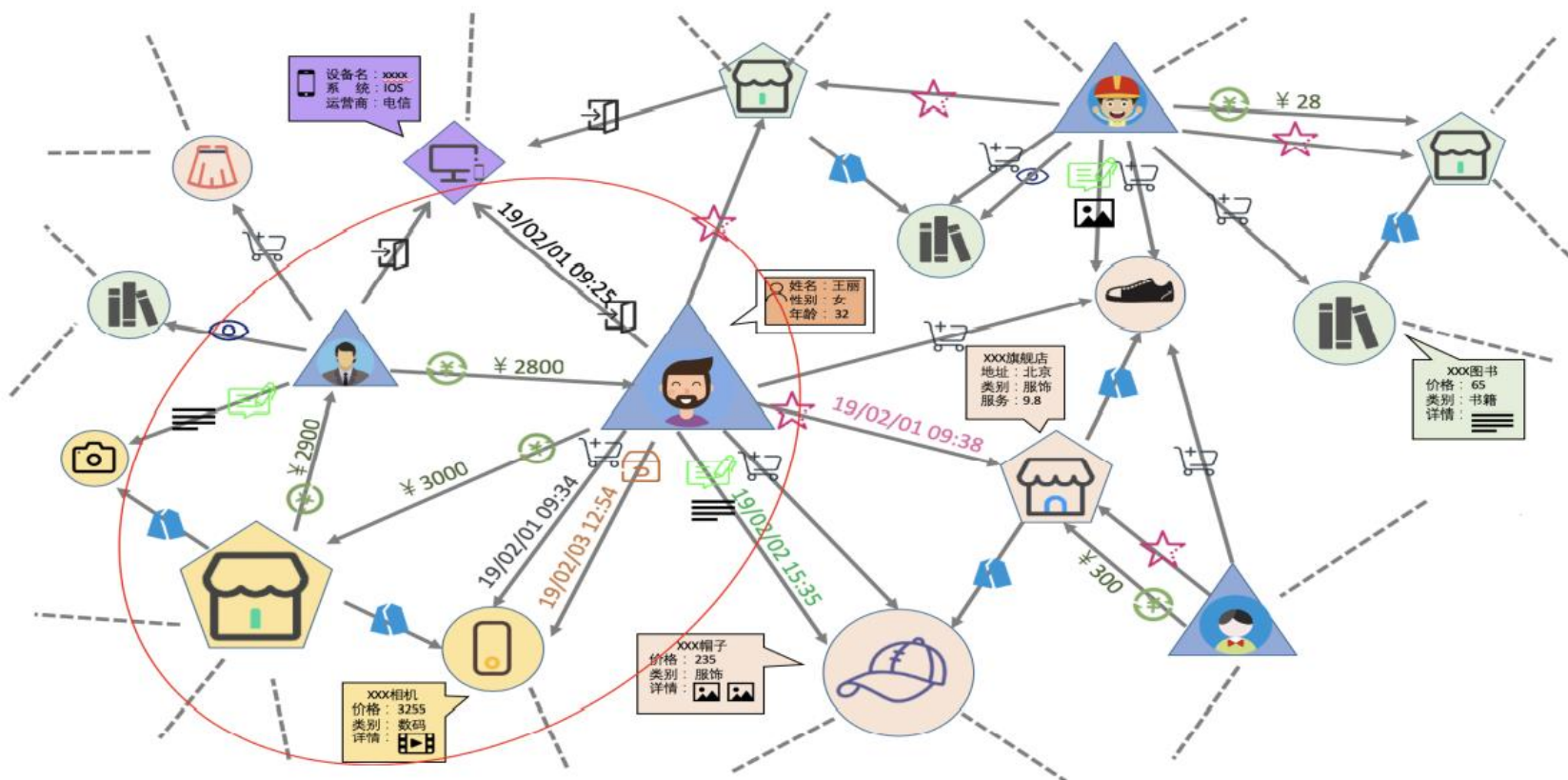


Heterogeneous
Networks

异构图

✓ 实际场景中

✎ 基本上但凡要应用基本都得是异构图，那么如何提取特征呢现在

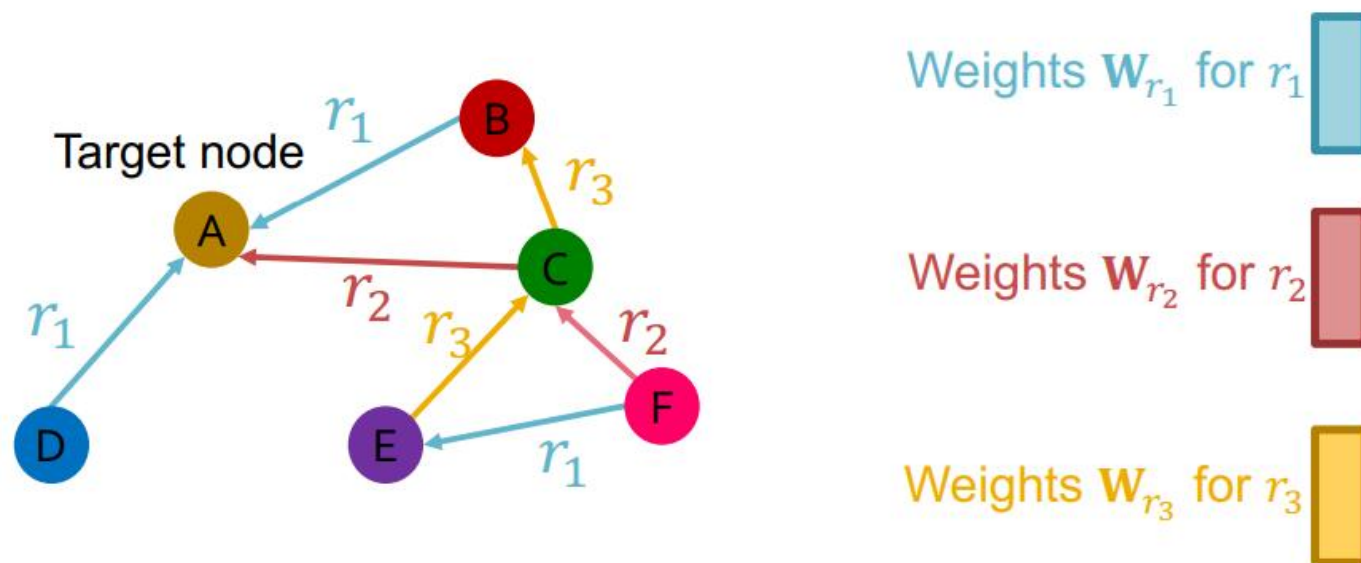


异构图

✓ 多种关系的时候该咋办呢

✎ 关系有点乱，但是每种关系处理好自己相关的节点就可以了

✎ 基本思想就是每种关系对应一组权重参数矩阵，是啥关系用啥矩阵

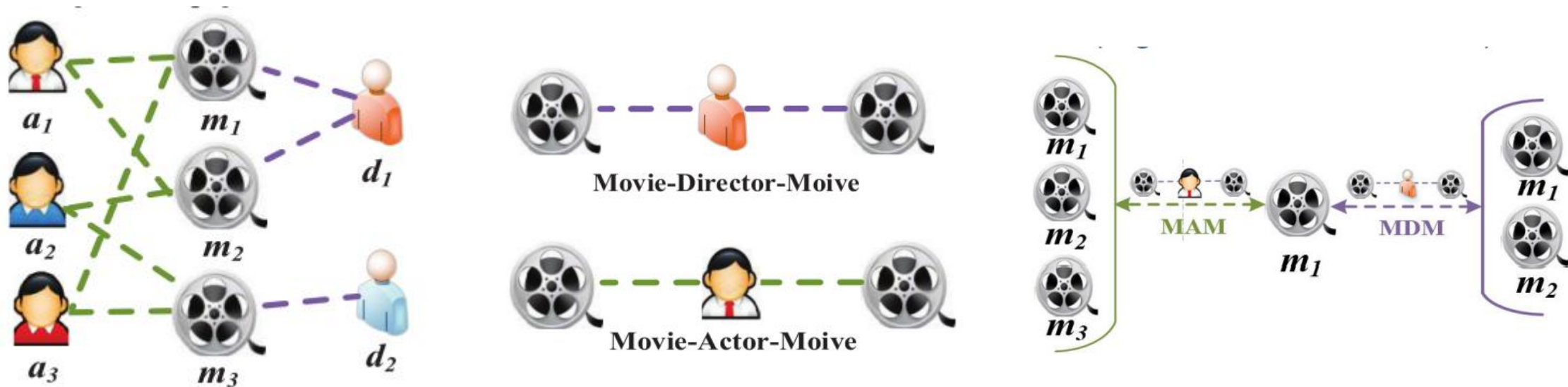


异构图

✓ Heterogeneous Graph Neural Networks

✎ 演员，电影，导演，这回咱们有三种节点以及他们之间关系也不同

✎ 首先第一个事就是meta-path（按照什么关系来寻找邻居）



异构图

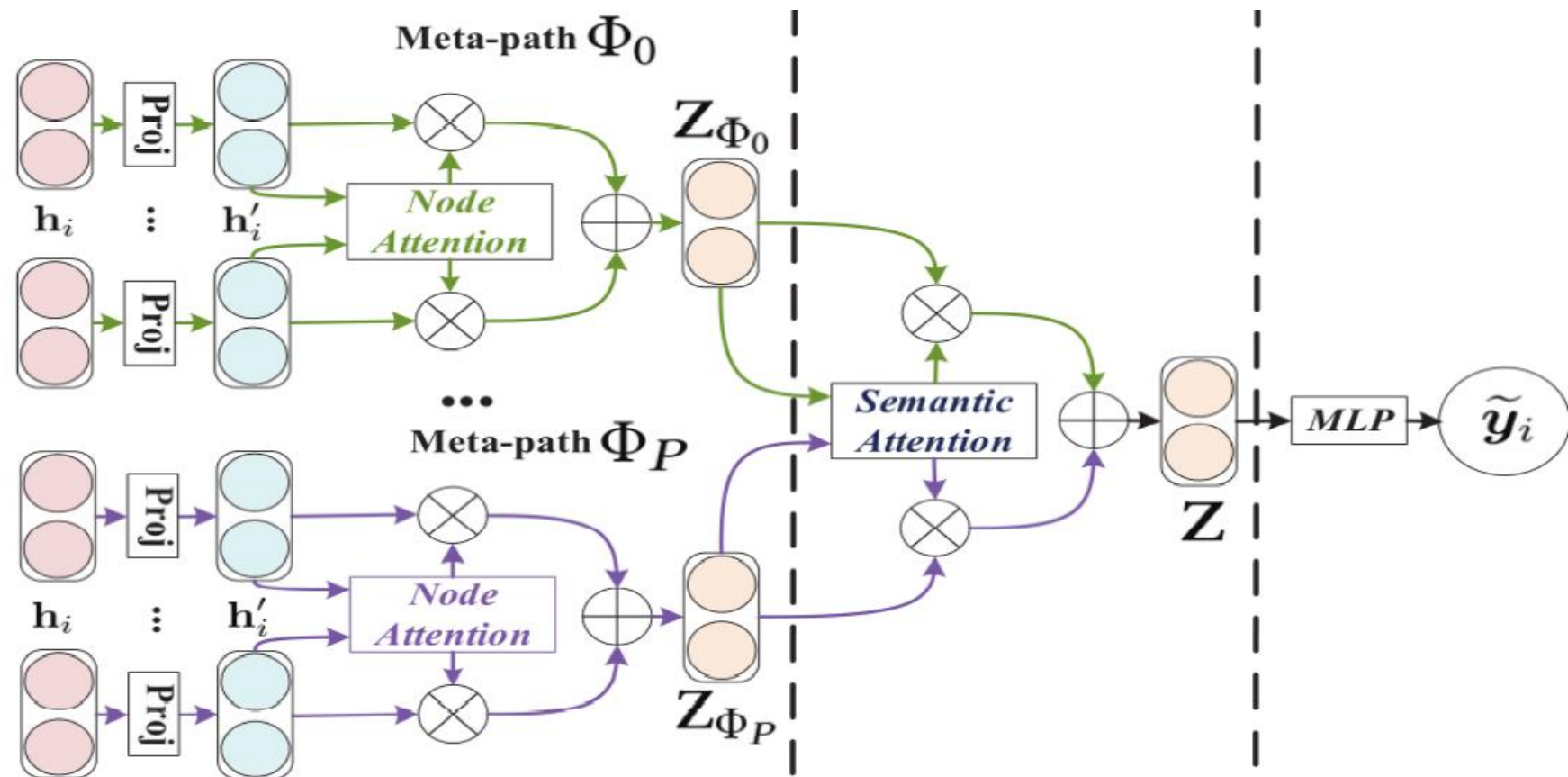
✓ 基本流程

📎 节点先attention

📎 不同路径聚合

📎 聚类也attention

📎 最后输出预测



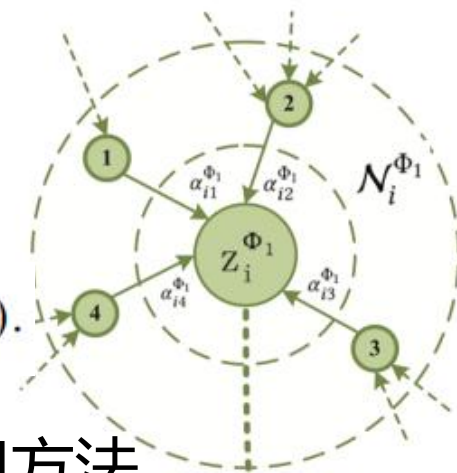
异构图

✓ 计算流程

✎ 对于每一个meta-path计算其上节点attention

$$\mathbf{h}'_i = \mathbf{M}_{\phi_i} \cdot \mathbf{h}_i,$$

$$e_{ij}^{\Phi} = \text{att}_{\text{node}}(\mathbf{h}'_i, \mathbf{h}'_j; \Phi).$$



✎ 首先特征映射(不管啥类型, 都映射到固定维度), 然后再用GAN方法

✎ softmax得到该meta-path下的权重分配, 也就能计算当前节点如何聚合特征

✎ 注意每个meta-path要学习的向量不同 $\alpha_{ij}^{\Phi} = \text{softmax}_j(e_{ij}^{\Phi}) = \frac{\exp(\sigma(\mathbf{a}_{\Phi}^T \cdot [\mathbf{h}'_i \| \mathbf{h}'_j]))}{\sum_{k \in \mathcal{N}_i^{\Phi}} \exp(\sigma(\mathbf{a}_{\Phi}^T \cdot [\mathbf{h}'_i \| \mathbf{h}'_k]))},$

异构图

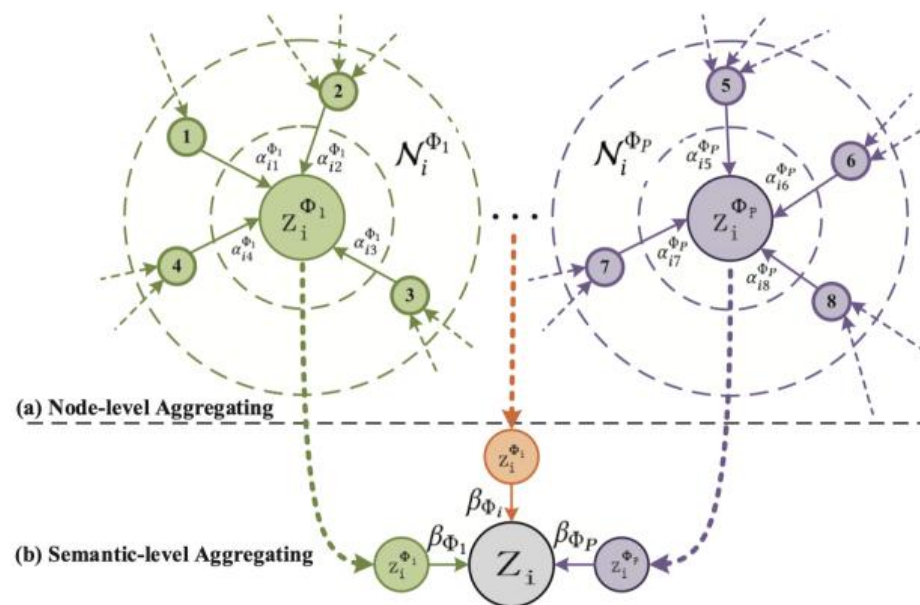
✓ 计算流程

✎ 聚合当前节点所有邻居特征（可拓展多头） $z_i^\Phi = \sigma\left(\sum_{j \in \mathcal{N}_i^\Phi} \alpha_{ij}^\Phi \cdot \mathbf{h}'_j\right)$. $z_i^\Phi = \parallel_{k=1}^K \sigma\left(\sum_{j \in \mathcal{N}_i^\Phi} \alpha_{ij}^\Phi \cdot \mathbf{h}'_j\right)$.

✎ meta-path权重: $w_{\Phi_p} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbf{q}^T \cdot \tanh(\mathbf{W} \cdot \mathbf{z}_i^{\Phi_p} + \mathbf{b})$,

✎ softmax: $\beta_{\Phi_p} = \frac{\exp(w_{\Phi_p})}{\sum_{p=1}^P \exp(w_{\Phi_p})}$,

✎ 最终节点embedding: $\mathbf{Z} = \sum_{p=1}^P \beta_{\Phi_p} \cdot \mathbf{Z}_{\Phi_p}$.



异构图

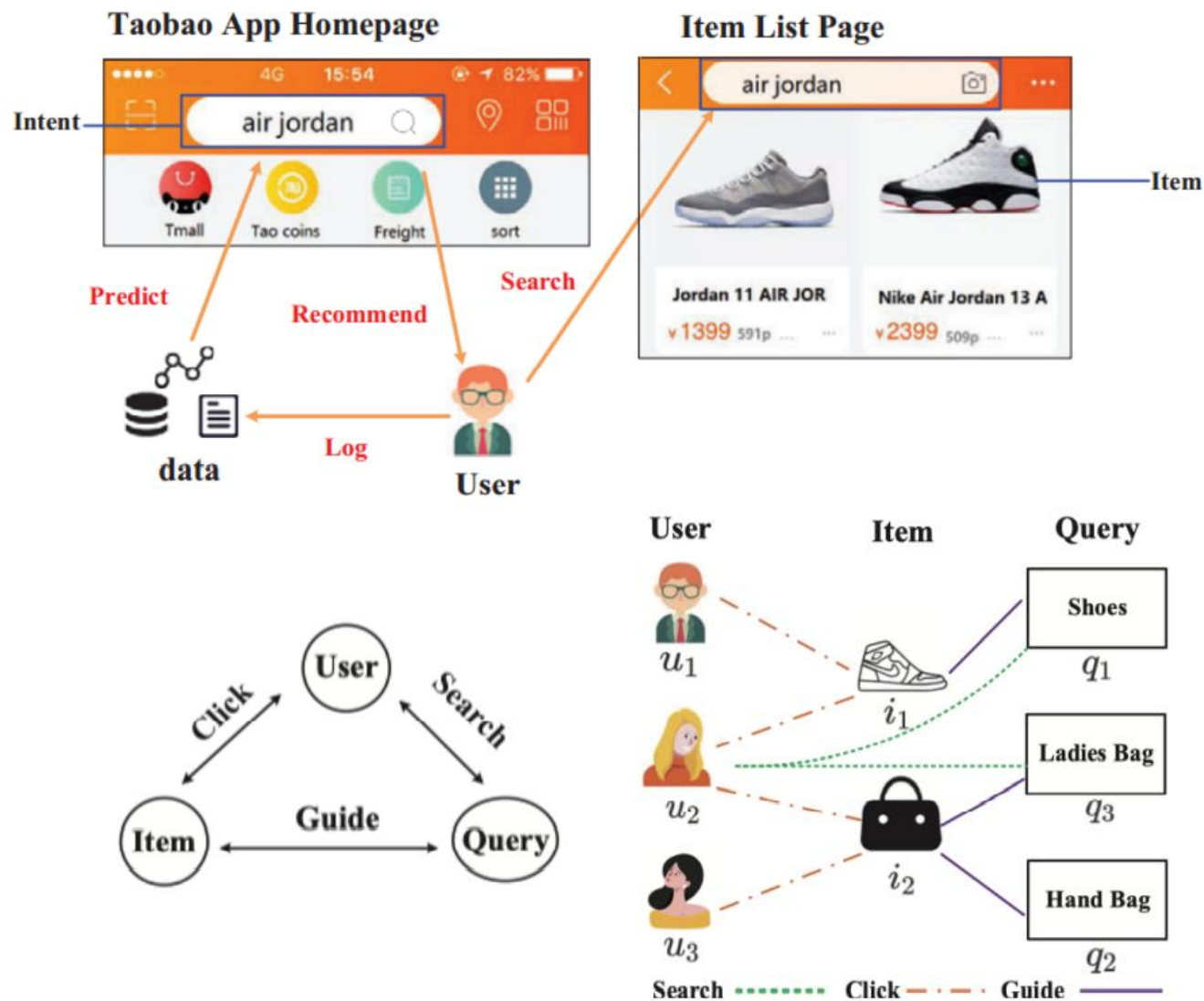
✓ 应用实例

✎ 用户根据关键词点击商品

✎ 那么现在就有三种节点了

✎ 任务：预测用户会不会买

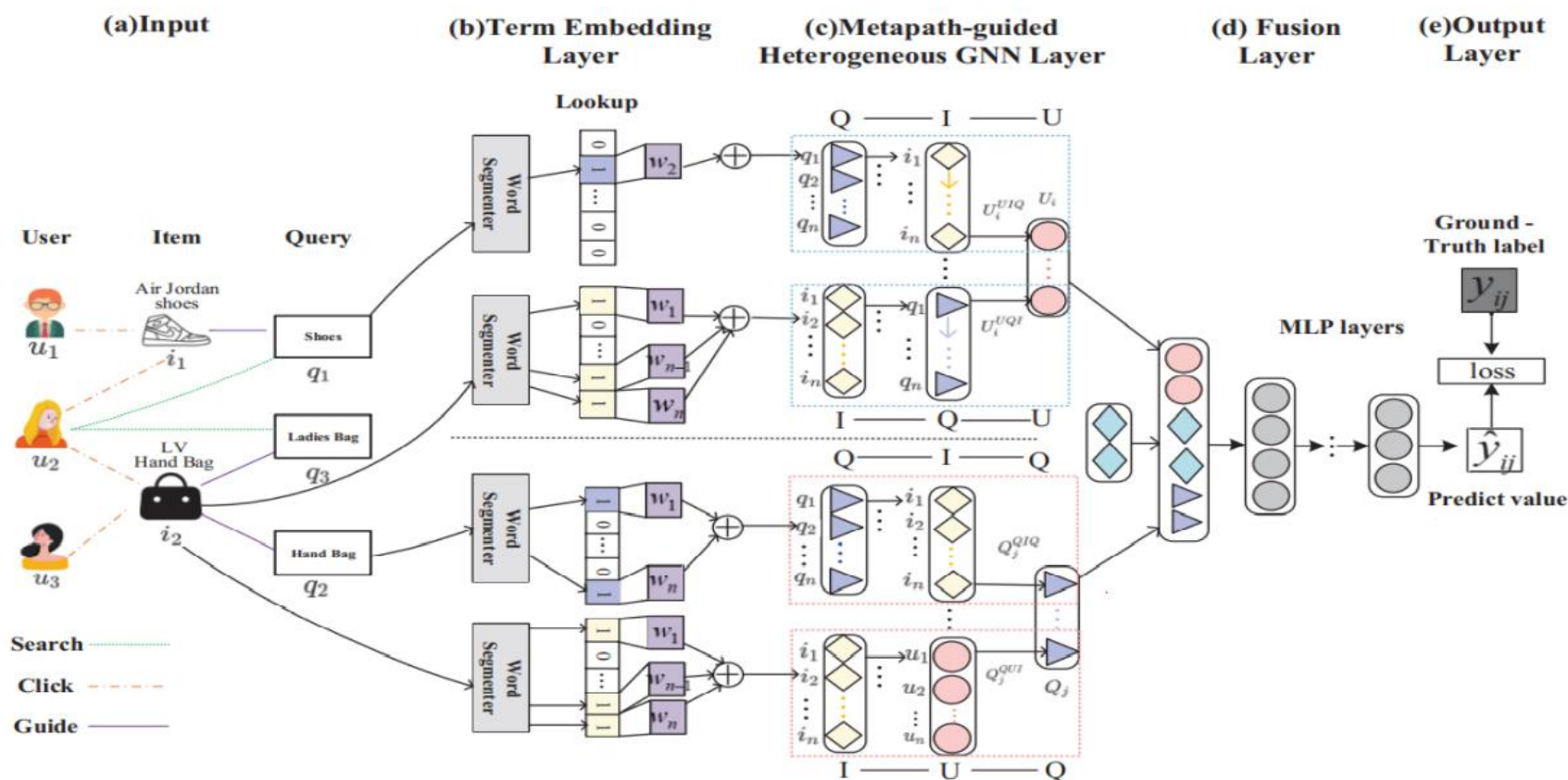
✎ 其实还是异构图提特征



异构图

✓ 应用实例

✎ 基本流程还是分别得到U和I以及其他特征的表达，然后拼接预测



异构图

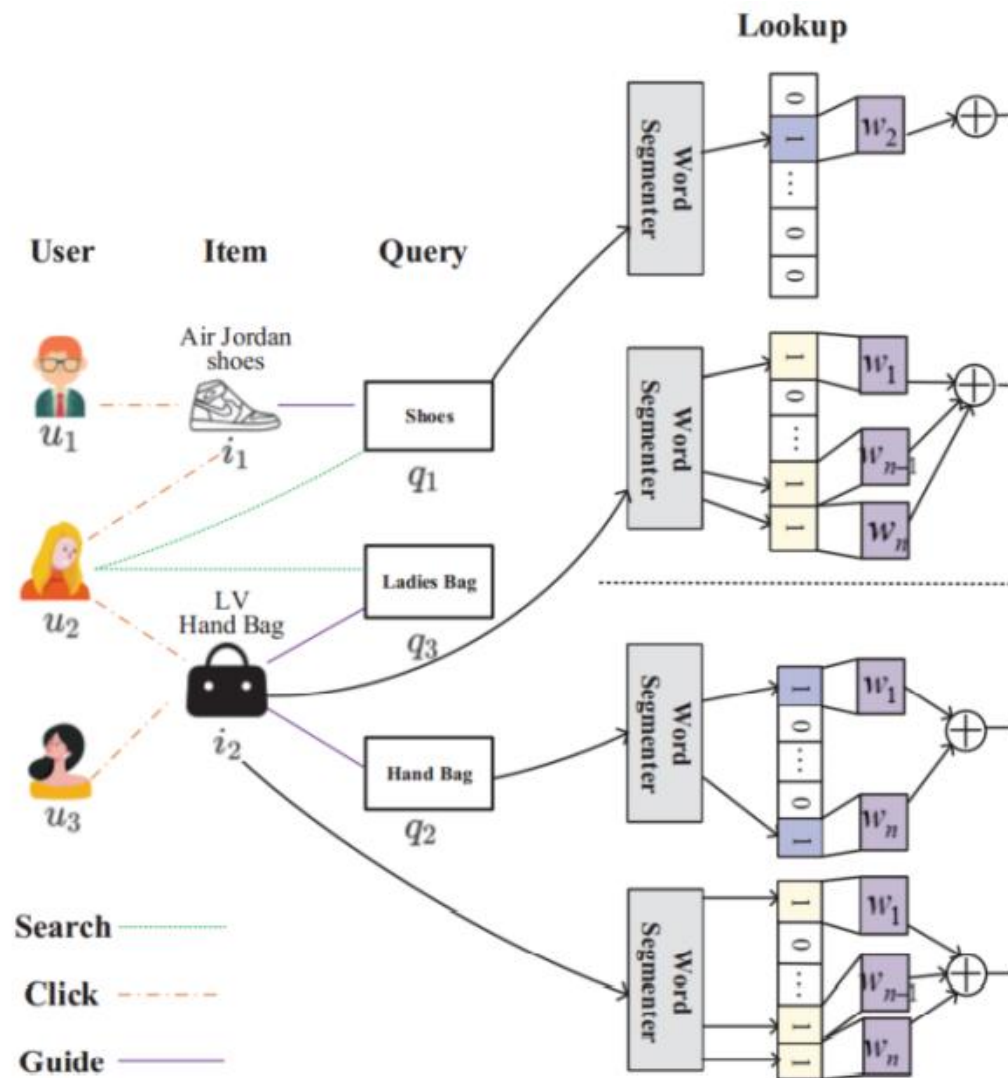
✓ 编码处理

✎ 针对I和Q，直接编码会导致矩阵太大了

✎ 创建固定词表，这样one-hot就可以了

✎ 简化Embedding表示后的结果

$$\{w_1, w_2, \dots, w_{n-1}, w_n\}$$
$$q_2 = (1, 0, \dots, 0, 1)$$
$$i_2 = (1, 0, \dots, 1, 1).$$



异构图

✓ 编码聚合预测

✎ 不同类别节点特征编码: $E_{q_2} = g(\mathbf{e}_{w_1}, \mathbf{e}_{w_n}), E_{i_2} = g(\mathbf{e}_{w_1}, \mathbf{e}_{w_{n-1}}, \mathbf{e}_{w_n}),$

✎ 按照不同meta-path聚合节点特征: $I_j^{\text{UIQ}} = g(E_{q_1}, E_{q_2}, \dots),$
 $U_i = g(U_i^{\rho_1}, U_i^{\rho_2}, \dots, U_i^{\rho_k}),$
 $U_i^{\text{UIQ}} = g(I_1^{\text{UIQ}}, I_2^{\text{UIQ}}, \dots),$

✎ 通过异构图分别得到U和I以及其他特征（如果有的话，都可以考虑进来）

✎ 最后进行特征拼接，完成预测即可: $\hat{y}_{ij} = \text{sigmoid}(f(U_i \oplus Q_j \oplus S_{ij})),$

异构图

✓ Link prediction

✎ 首先第一件事就是数据集切分，跟之前有些不同

✎ 为了防止透题，训练集中要区别消息传递的和输出预测的(计算损失的)

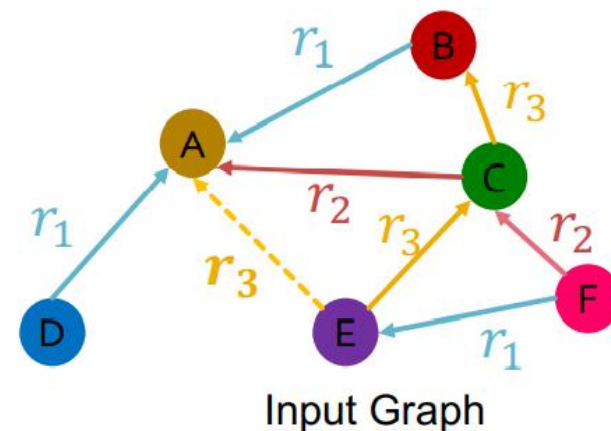


异构图

✓ Link prediction

✎ 右图中 (E, r_3, A) 是 training supervision edge

✎ 其他的边都是 training supervision edge



✎ 虚线的我们可以当作正样本，但是负样本如何定义呢？

✎ 可以生成一些负样本，个数都可以自己定，例如： (E, r_3, B) , (E, r_3, D)

异构图

✓ Link prediction

✎ 任何图神经网络模型都可以，聚合得到各点特征

✎ 通过生成的负样本与正样本进行有监督训练

✎ 损失函数如下：（预测链路是否存在）

$$\ell = -\log \sigma \left(f_{r_3}(h_E, h_A) \right) - \log(1 - \sigma(f_{r_3}(h_E, h_B)))$$

