



中国科学技术大学
University of Science and Technology of China

《人工智能数学原理与算法》

第4章：图神经网络

4.3 图神经网络

王翔

xiangwang@ustc.edu.cn



01 回顾

02 图神经网络：思想、建模

03 图神经网络：训练

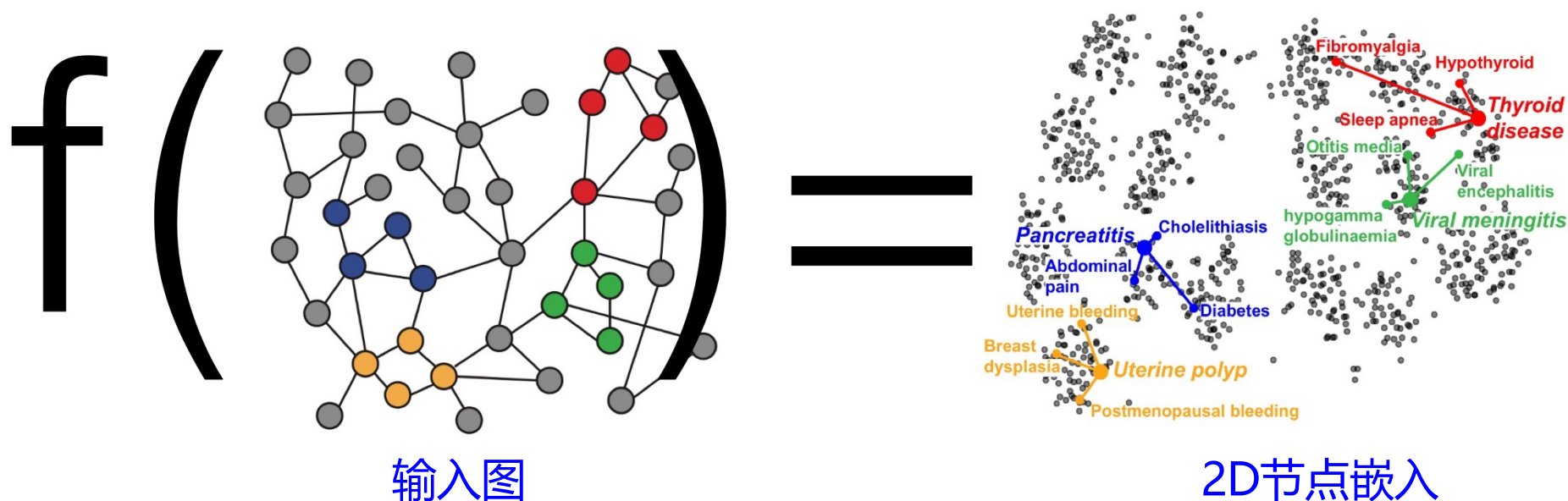
04 图神经网络：优势



目录

回顾：节点嵌入

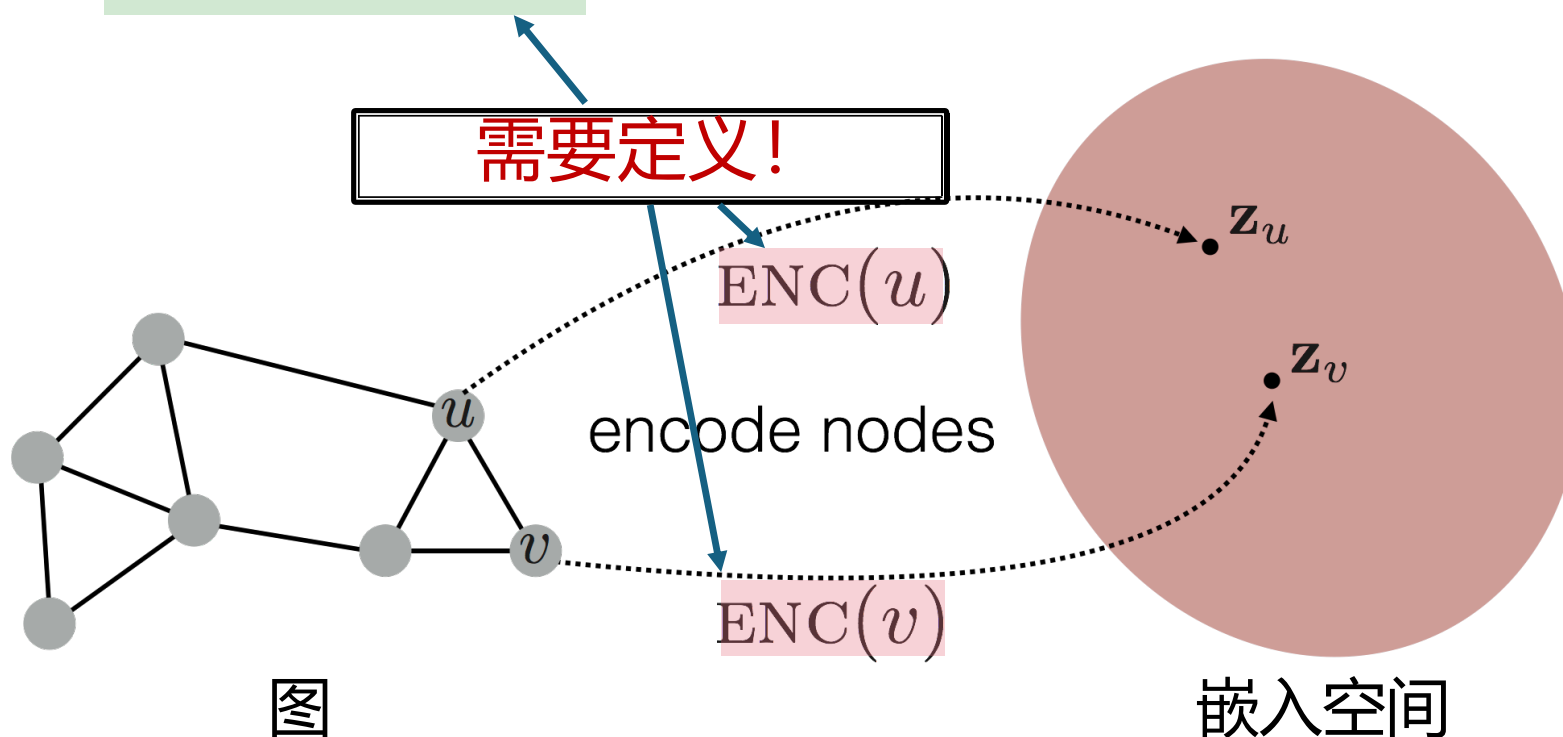
□ **目标**：目标是对节点进行编码，使得**嵌入空间中的相似性**（例如点积）能够近似反映**图中的相似性**



如何学习映射函数 f ?

回顾：节点嵌入

□ 目标： $\text{similarity}(u, v) \approx \mathbf{z}_v^T \mathbf{z}_u$
图上相似度 嵌入空间相似度



回顾：三个关键

1. 编码器Encoder将节点映射到嵌入表示

$$\text{ENC}(\mathbf{v}) = \mathbf{z}_v \in R^d$$

2. 定义一个节点相似性函数（即，度量图中的相似性）

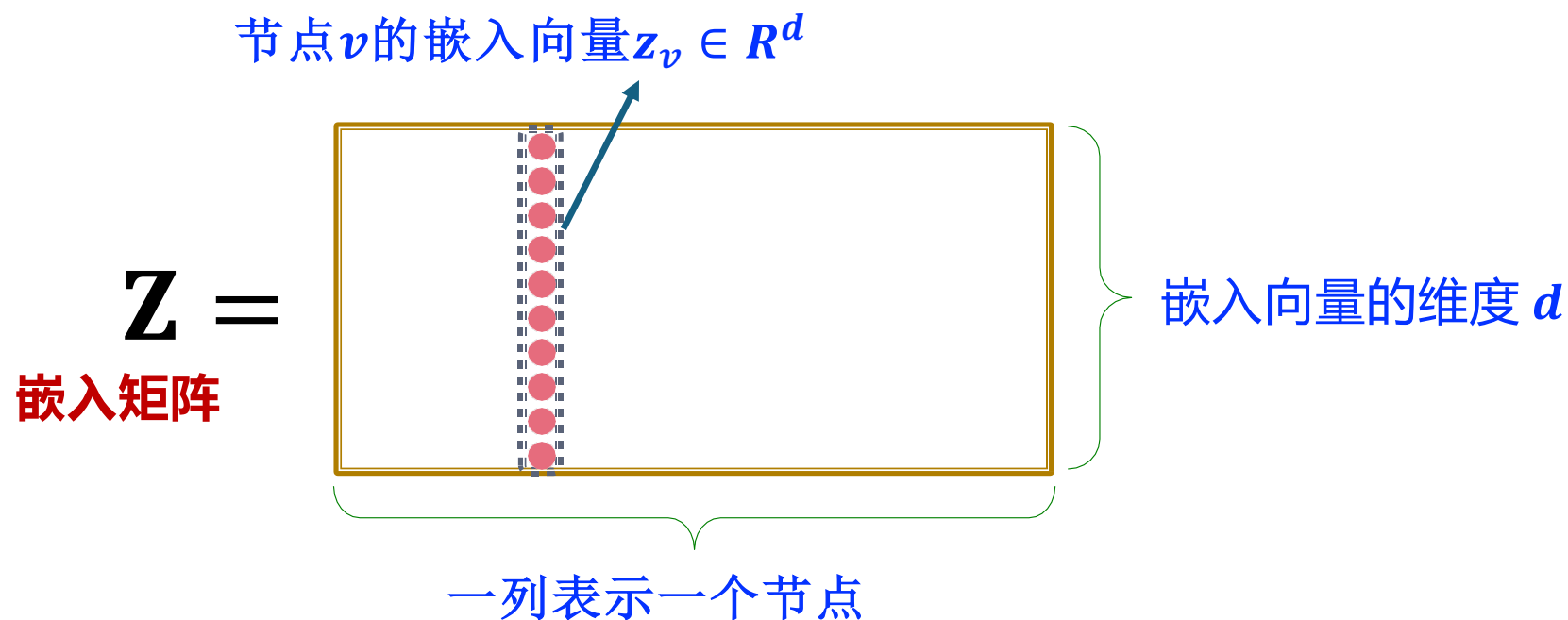
3. 解码器Decoder从嵌入中恢复相似性得分

4. 优化编码器的参数，使得：

$$\begin{array}{ccc} \text{similarity}(\mathbf{u}, \mathbf{v}) & \approx & \text{DEC}(\mathbf{z}_u, \mathbf{z}_v) = \mathbf{z}_v^T \mathbf{z}_u \\ \text{图上相似度} & & \text{嵌入空间相似度} \end{array}$$

回顾：“浅层”编码

- 最简单的编码方法：编码器仅是一个嵌入查找表



回顾：“浅层”编码

浅层嵌入方法的局限性：

- 需要 $O(|V|d)$ 个参数：

- 节点之间没有参数共享，每个节点都有其唯一的嵌入向量。

- 本质上是“传导式 (transductive)”：

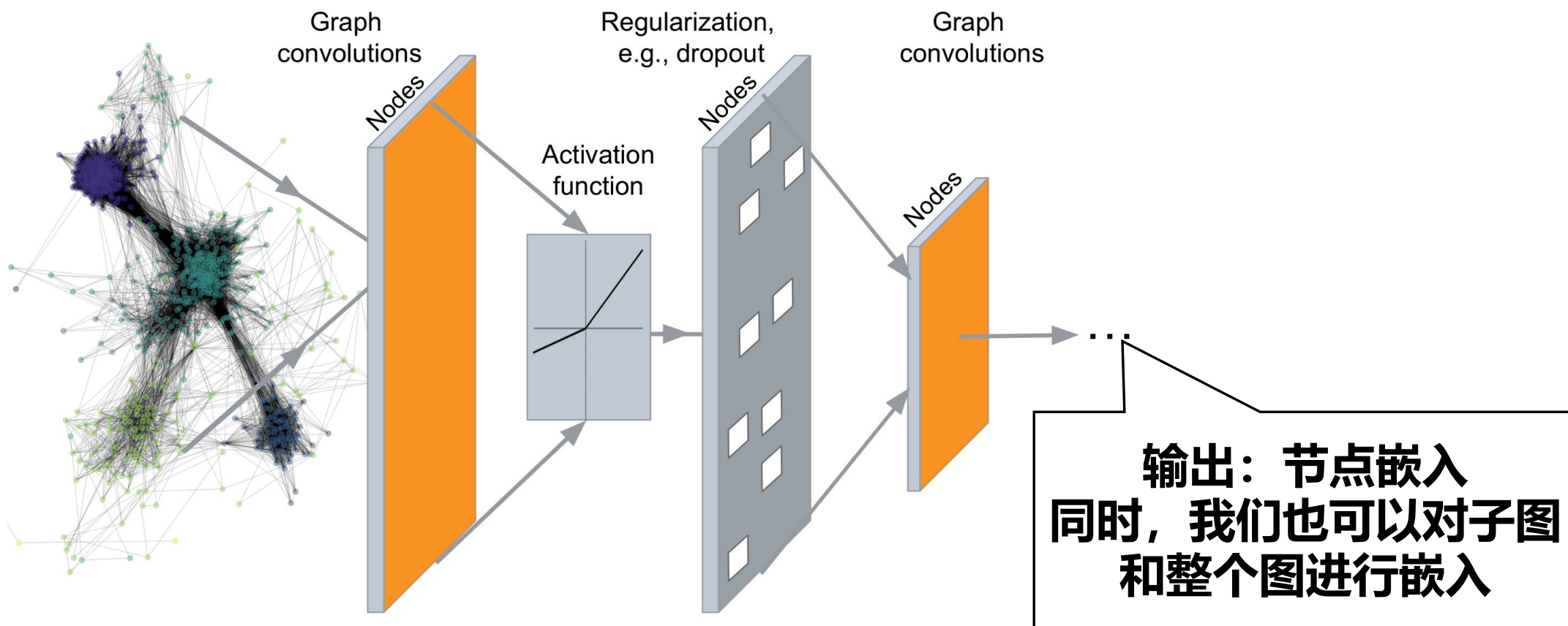
- 无法为训练期间未见的节点生成嵌入表示。

- 未利用节点特征：

- 许多图中的节点具有可用且应当利用的特征，但这些方法没有加以整合。

基于图神经网络的深度编码

$\text{ENC}(v) =$ 基于图结构的多层非线性变换



目录

01 回顾

02 图神经网络：思想、建模

03 图神经网络：训练

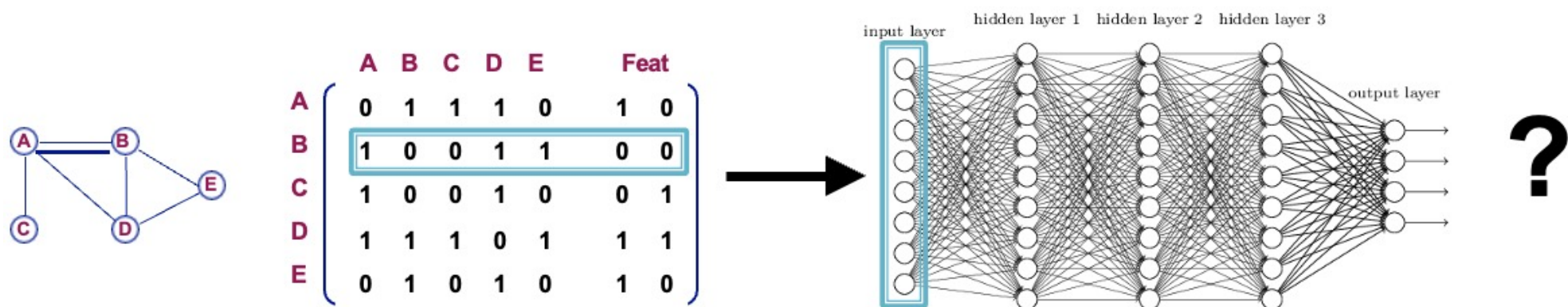
04 图神经网络：优势

□ **假设：**假设我们有一个图 G ：

- V 是节点集合
- $A \in R^{|V| \times |V|}$ 是链接矩阵（假设为二值矩阵）
- $X \in R^{|V| \times m}$ 是节点特征矩阵
- $N(v)$ 是节点 v 的邻居节点集合
- **节点特征：**
 - 社交网络：用户资料、用户图像
 - 生物网络：基因表达谱、基因功能信息

一个简单的方案

□ 将邻接矩阵与特征结合，输入到深度神经网络中：



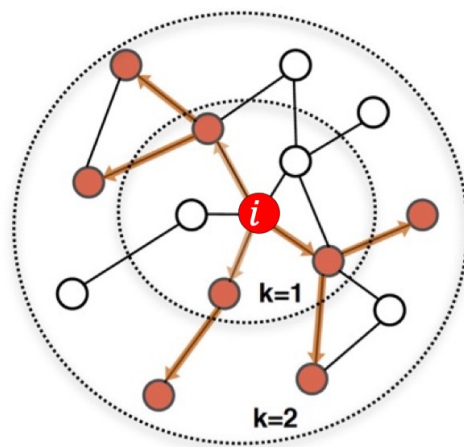
□ 局限：

- 参数量为 $O(|V|)$
- 不适用于不同大小的图
- 对节点顺序敏感

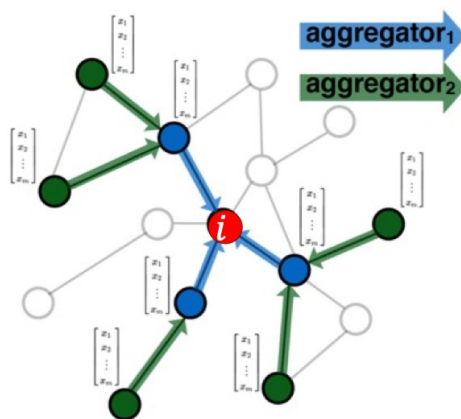
图神经网络 (Graph Neural Networks, GNNs)

□ 思路:

- 节点的邻居定义了一张**计算图 (computation graph)**
- 学习如何在计算图中**传播与变换信息 (propagate & transform information)** 以计算节点嵌入



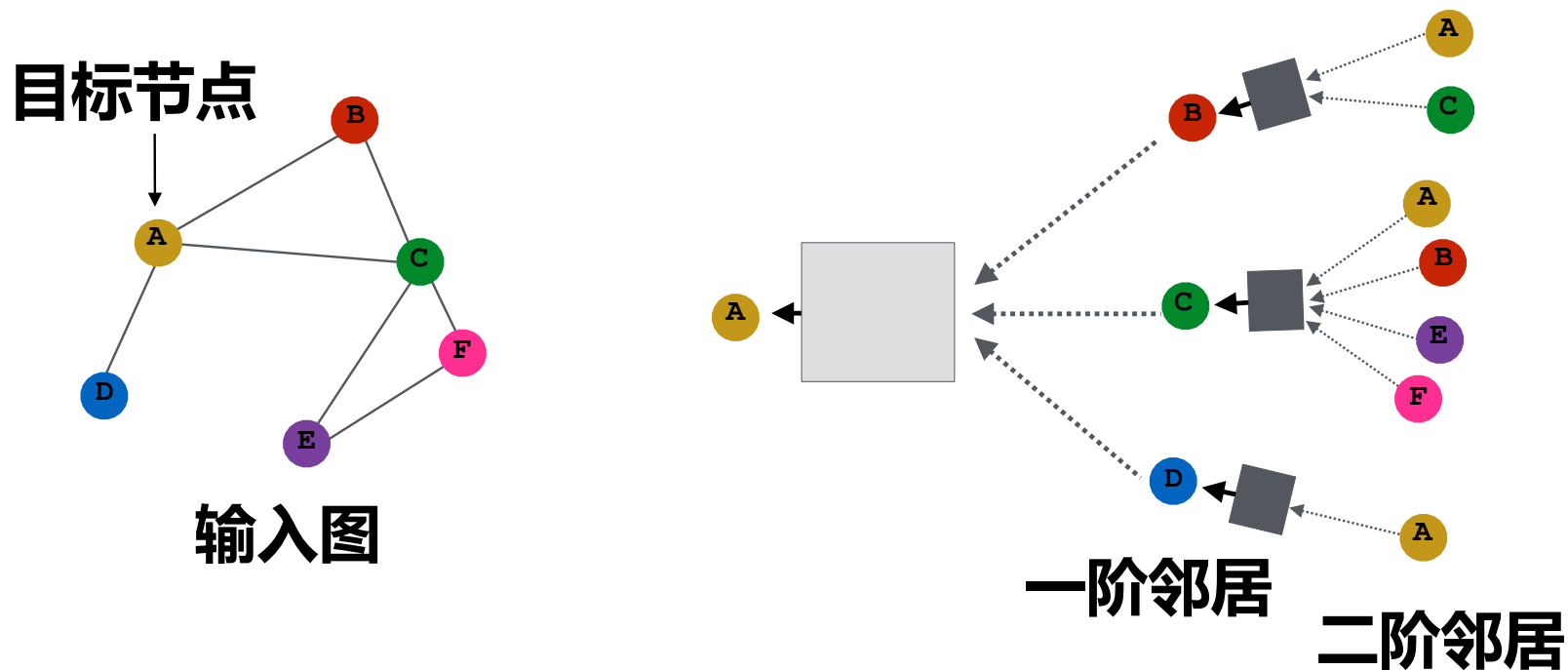
确定节点的计算图



传播并变换信息

思想：领域聚合 (Aggregate Neighbors)

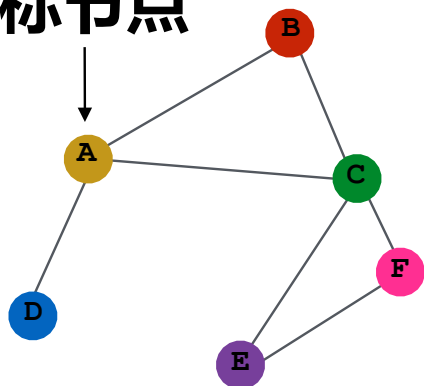
□ 核心思想：基于局部邻域 (local neighborhoods) 生成节点嵌入



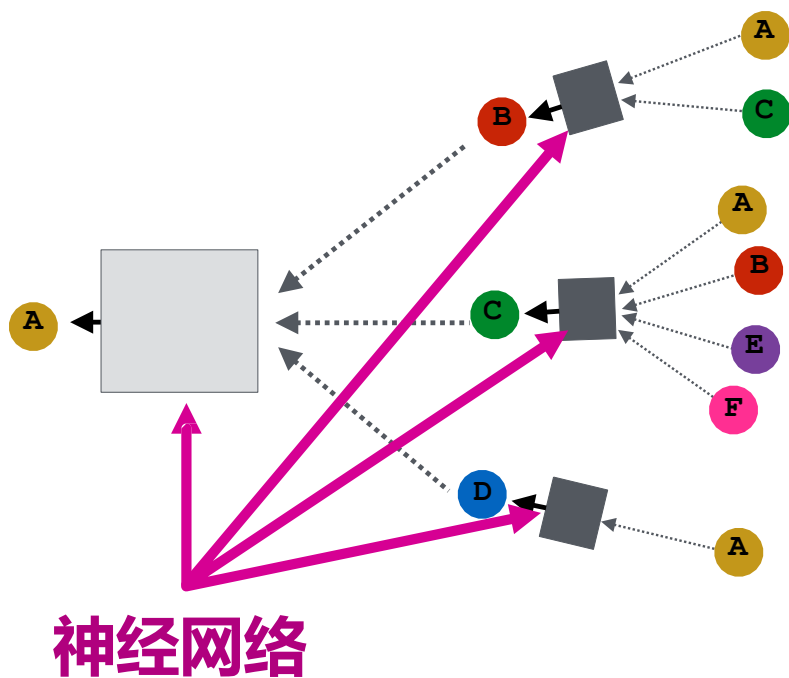
思想：领域聚合 (Aggregate Neighbors)

□ 直觉：节点通过神经网络从其邻居中聚合信息

目标节点

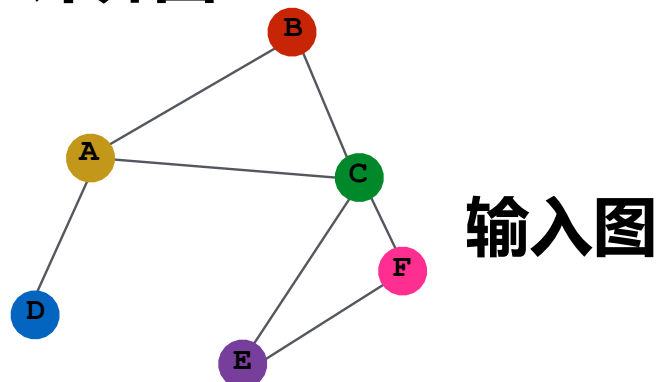


输入图

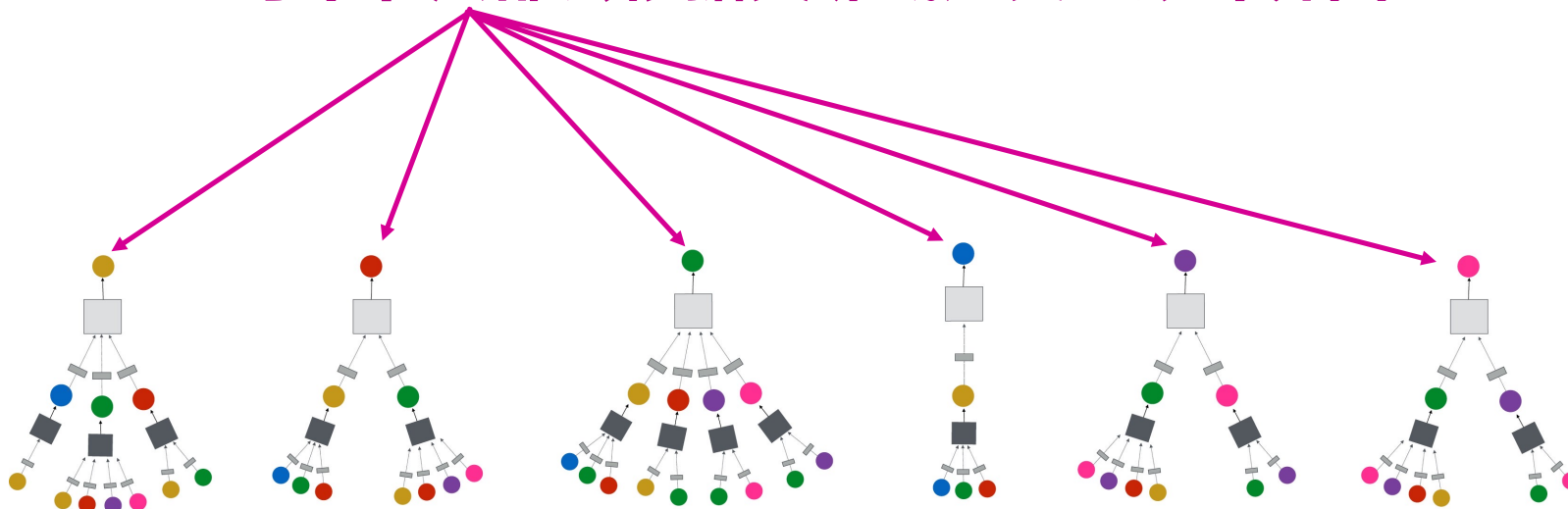


思想：邻域聚合 (Aggregate Neighbors)

□ 直觉：邻域定义了一张计算图



每个节点都会根据其邻域定义一张计算图！

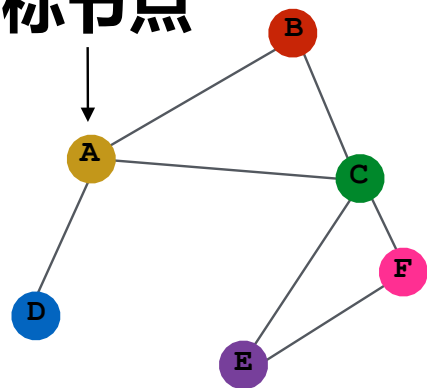


思想：领域聚合 (Aggregate Neighbors)

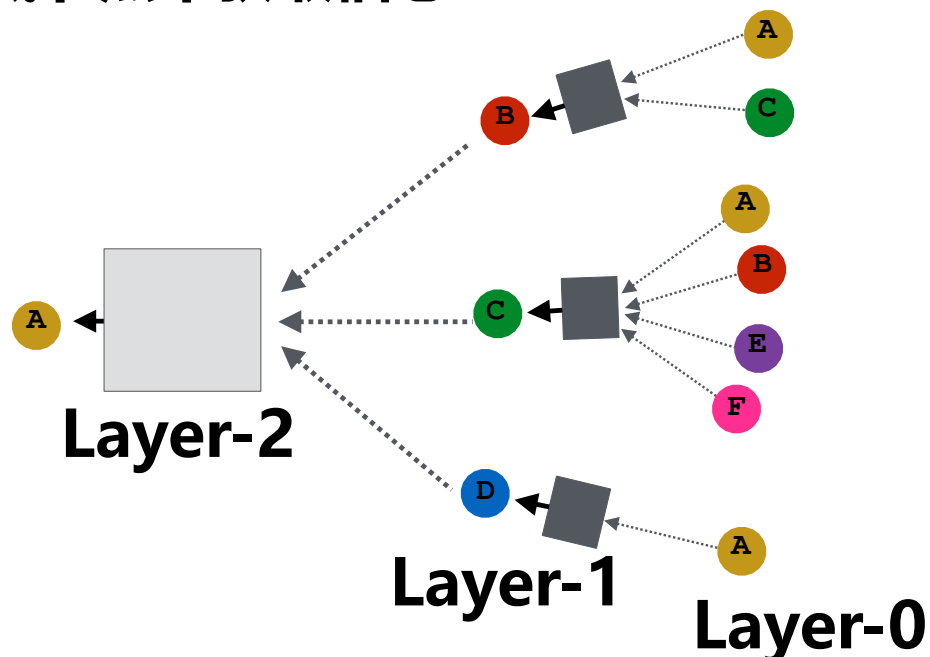
□ 模型可以具有任意深度：

- 每个节点在每一层都有嵌入表征
- 第0层的嵌入是节点 v 的输入特征 X_v
- 第 k 层的嵌入是从 k 阶邻域节点中获取信息

目标节点



输入图

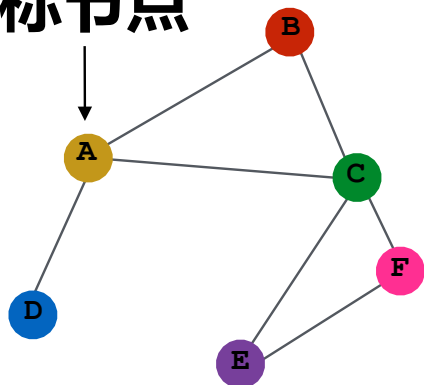


思想：邻域聚合 (Aggregate Neighbors)

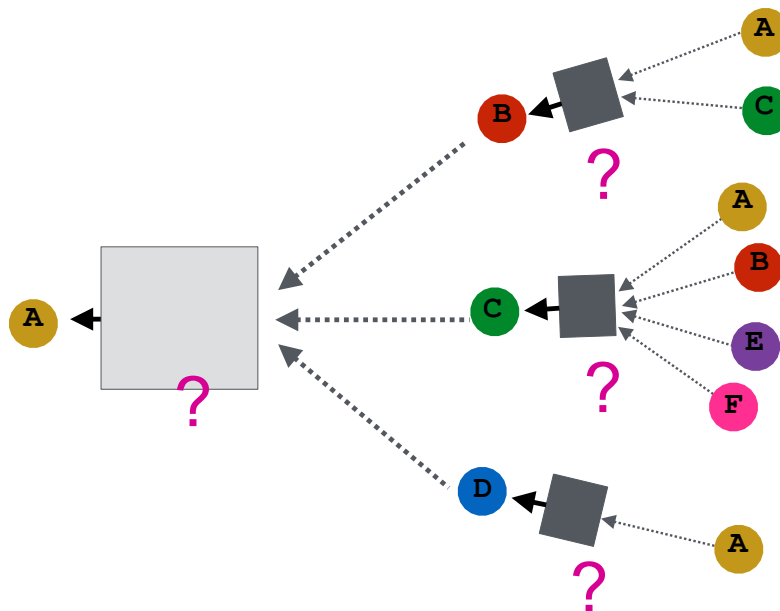
□ 邻域聚合：

- 不同方法的关键区别在于它们在各层中聚合信息的方式

目标节点



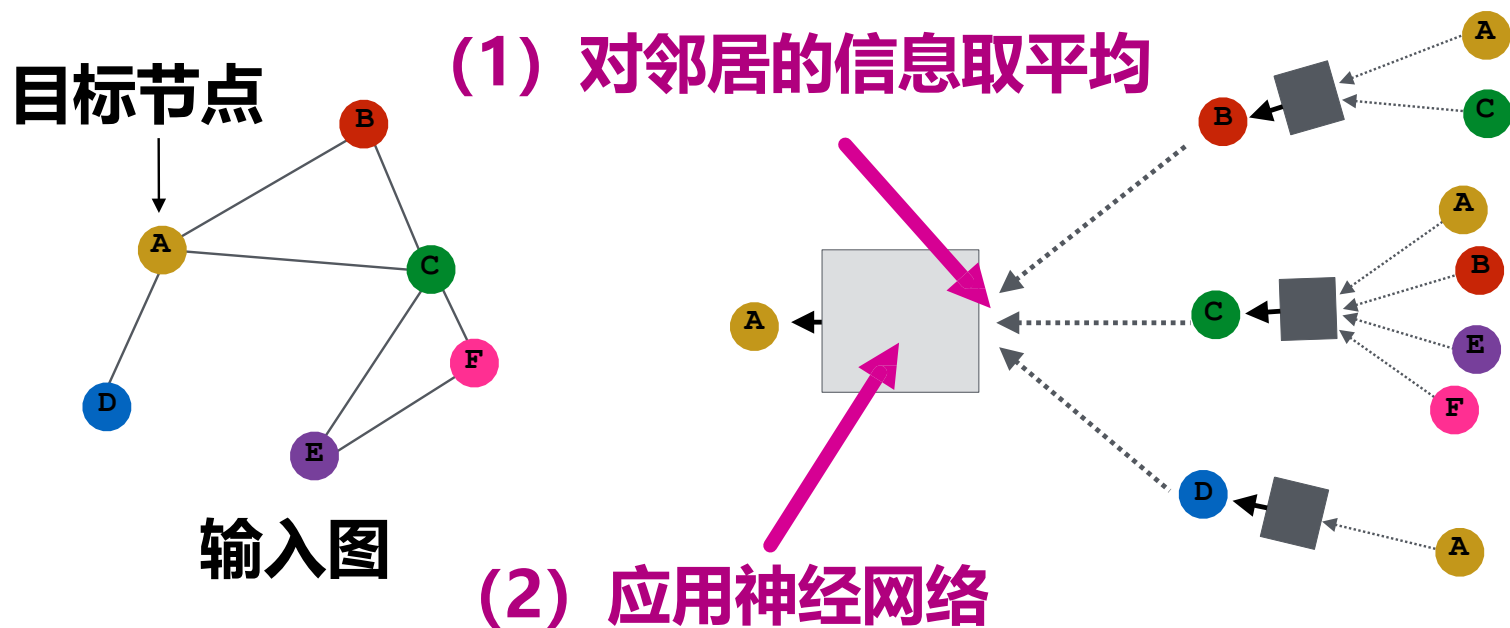
输入图



思想：领域聚合 (Aggregate Neighbors)

□ 基本方法：

- 对邻居的信息取平均后，应用神经网络进行处理



数学：深度编码器

□ 基本方法：

- 对邻居的信息取平均后，应用神经网络进行处理

$h_v^0 = x_v$ 初始的第0层嵌入等于节点的特征。

对邻居节点的上一层嵌入取平均

$$h_v^{(k+1)} = \sigma \left(W_k \sum_{u \in N(v)} \frac{h_u^{(k)}}{|N(v)|} + B_k h_v^{(k)} \right), \forall k \in \{0, \dots, K-1\}$$

非线性激活函数 第k层嵌入 神经网络总层数

$z_v = h_v^{(K)}$ 经过K层邻域聚合后的嵌入表示 → 最终的节点嵌入

数学：深度编码器

□ 模型参数

- W_k : 第 k 层用于邻域聚合的权重矩阵
- B_k : 第 k 层用于变换节点自身嵌入的权重矩阵

可训练的权重矩阵（即，我们需要学习的）

$$h_v^{(k+1)} = \sigma \left(\boxed{W_k} \sum_{u \in N(v)} \frac{h_u^{(k)}}{|N(v)|} + \boxed{B_k} h_v^{(k)} \right), \forall k \in \{0, \dots, K-1\}$$

矩阵形式

□ 许多聚合操作可以通过（稀疏）矩阵运算高效地完成。

➤ 让 $H^{(k)} = [h_1^{(k)}, \dots, h_{|V|}^{(k)}] \in R^{|V| \times d_k}$ 为第 k 层所有节点的嵌入矩阵

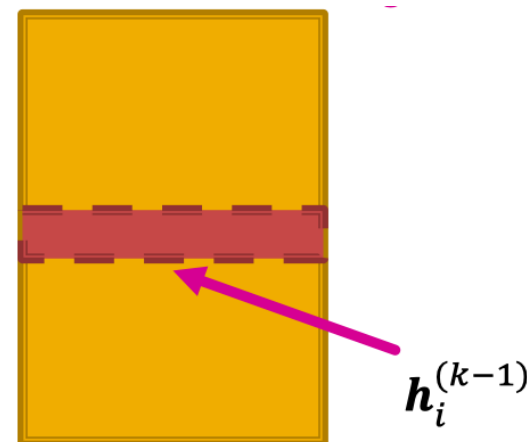
则有: $\sum_{u \in N(v)} h_u^{(k)} = A_v H^{(k)}$

➤ 让 $D \in R^{|V| \times |V|}$ 为对角矩阵, 其中 $D_{v,v} = |N(v)|$ 节点 v 的度数

则有: $D_{v,v}^{-1} = \frac{1}{|N(v)|}$

因此:

则有: $\sum_{u \in N(v)} \frac{h_u^{(k)}}{|N(v)|} = D_{v,v}^{-1} A_v H^{(k)}$



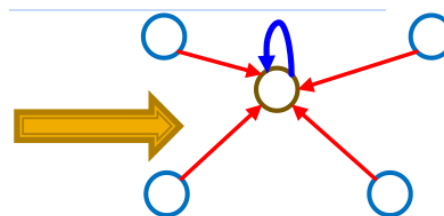
矩阵形式

□ 将更新函数重写为矩阵形式：

$$H^{(k+1)} = \sigma(\tilde{A}H^{(k)}W_k^T + H^{(k)}B_k^T), \quad \tilde{A} = D^{-1}A$$

➤ $\tilde{A}H^{(k)}W_k^T$ ：领域聚合

➤ $H^{(k)}B_k^T$ ：自身变换



- 在实际中，这意味着可以使用**高效的稀疏矩阵乘法**（因为 A 是稀疏的）。
- 注意：当聚合函数较复杂时，并非所有的图神经网络都能用简单的矩阵形式表示。

目录

01 回顾

02 图神经网络：思想、建模

03 图神经网络：训练

04 图神经网络：优势

如何训练图神经网络?

□ 节点嵌入 z_v 是输入图的函数

□ **有监督设定**: 我们的目标是最小化损失函数 \mathcal{L} :

$$\min_{\Theta} \mathcal{L}(y, f_{\Theta}(z_v))$$

➤ y 是节点标签

➤ 如果 y 是实数, \mathcal{L} 可以是L2损失; 如果 y 是类别, \mathcal{L} 可以是交叉熵损失;

□ **无监督设定**:

➤ 没有节点标签可用

➤ 利用图结构本身作为监督信号!

无监督训练

□ 解决方案1：图上“相似”的节点具有相似的嵌入

$$\min_{\Theta} \mathcal{L}(y, f_{\Theta}(z_v)) = \sum_{z_v, z_u} \text{CE}(y_{u,v}, \text{DEC}(z_u, z_v))$$

- 如果节点 u 和 v 是相似的, $y_{u,v} = 1$; 否则的话, $y_{u,v} = 0$
- $z_u = f_{\Theta}(u)$, $\text{DEC}(z_u, z_v) = z_v^T z_u$
- CE是交叉熵损失函数

$$\text{CE}(y, f(x)) = - \sum_{i=1}^c (y_i \log(f_{\Theta}(x)_i))$$

无监督训练

□ 解决方案2: 图上“相似”的节点具有相似的嵌入

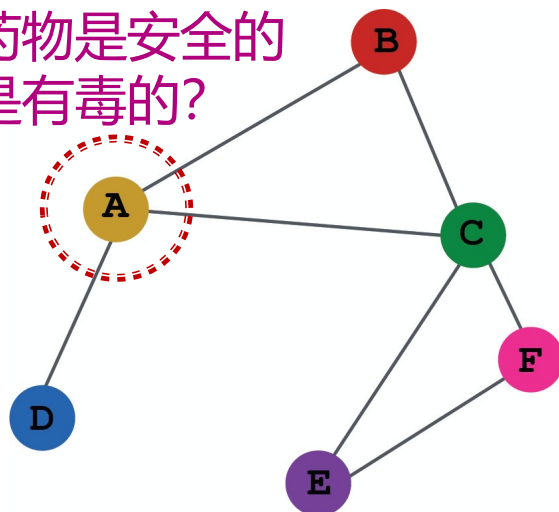
$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log\left(\frac{\exp(\mathbf{z}_u^T \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^T \mathbf{z}_n)}\right)$$

对所有节点 u 求和 与 u 相邻的节点 v 求和 计算节点 u 和 v 在相似的预测概率

有监督训练

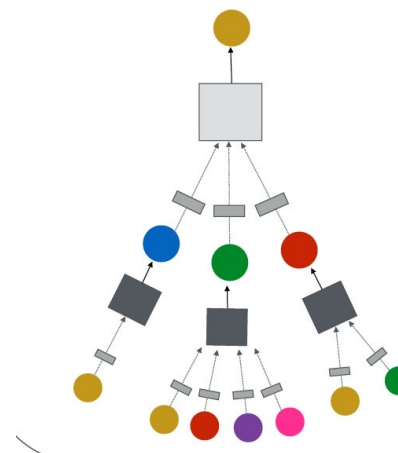
□ 直接针对监督任务训练模型（例如，节点分类）

这个药物是安全的
还是有毒的？



例如：一个药物-
药物相互作用网络。

这个药物是安全的还是有毒的？



有监督训练

□ 直接针对监督任务训练模型（例如，节点分类）

➤ 使用交叉熵损失函数

$$\mathcal{L} = - \sum_{v \in V} y_v \log \left(\sigma(z_v^T \theta) \right) + (1 - y_v) \log \left(1 - \sigma(z_v^T \theta) \right)$$

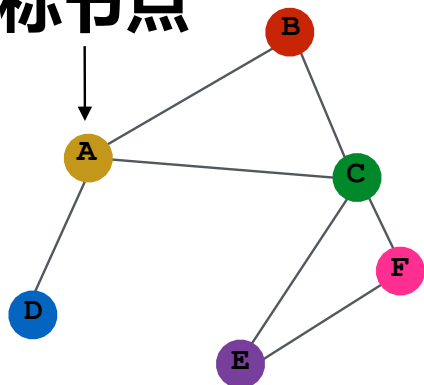
编码器输出：节点嵌入

节点类别标签
(有毒、无毒)

分类器权重

模型设计：全局概览

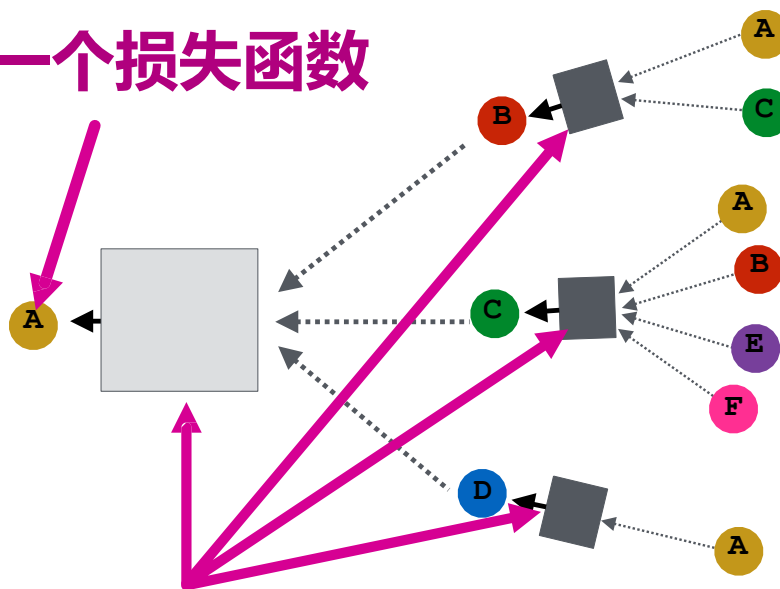
目标节点



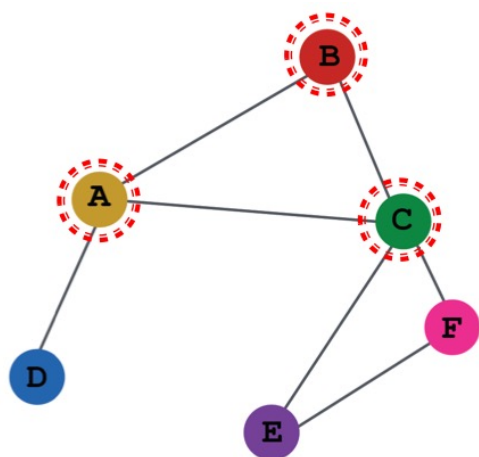
输入图

(2) 在嵌入表示上
定义一个损失函数

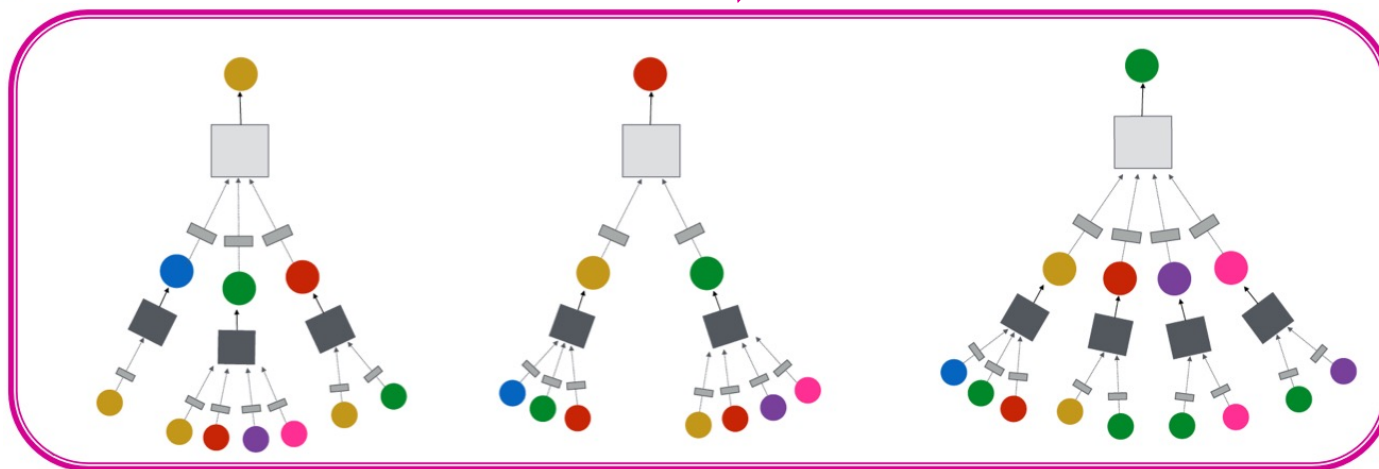
(1) 定义一个邻域聚合函数



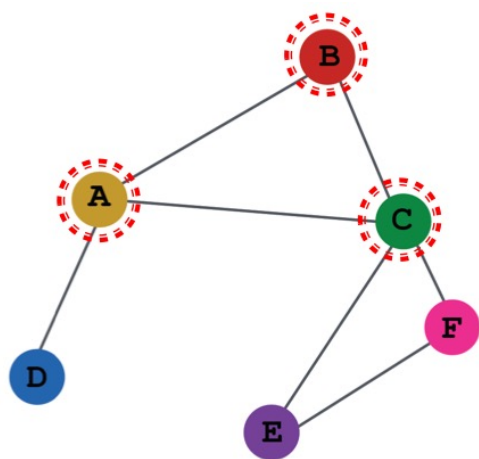
模型设计：全局概览



(3) 在一组节点上进行训练，
即一批计算图。

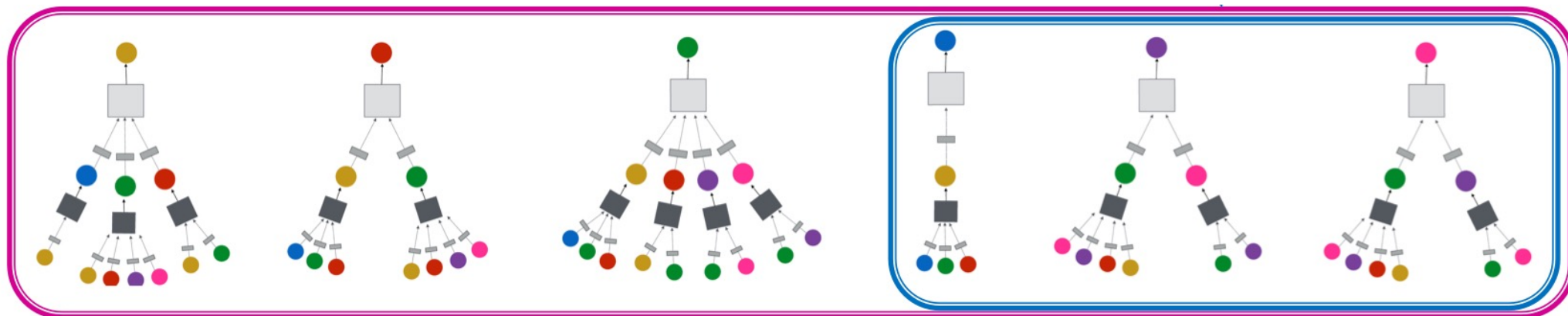


模型设计：全局概览



(4) 根据需要生成节点的嵌入表示。

即使是从未训练过的节点也可以生成



目录

01 回顾

02 图神经网络：思想、建模

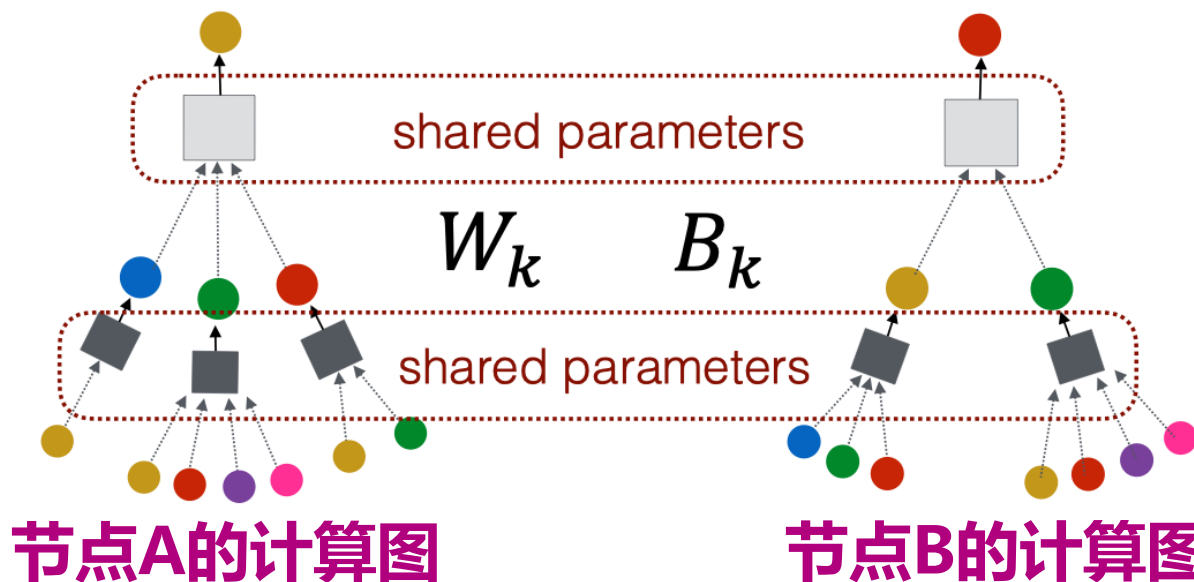
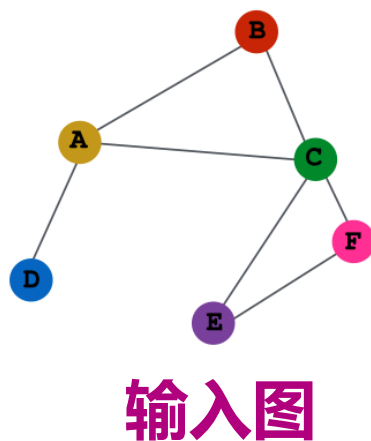
03 图神经网络：训练

04 图神经网络：优势

模型优势：归纳能力 (Inductive Capability)

□ 所有节点共享相同的聚合参数：

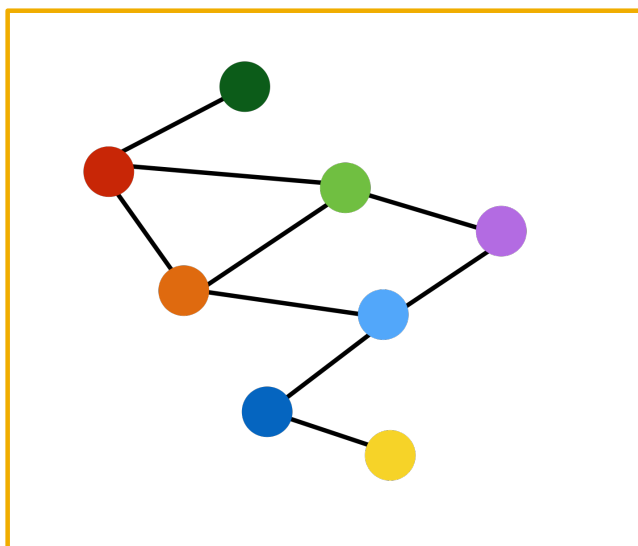
- 模型参数数量相对于 $|V|$ 是次线性的
- 并且可以推广到**未见过的节点**！



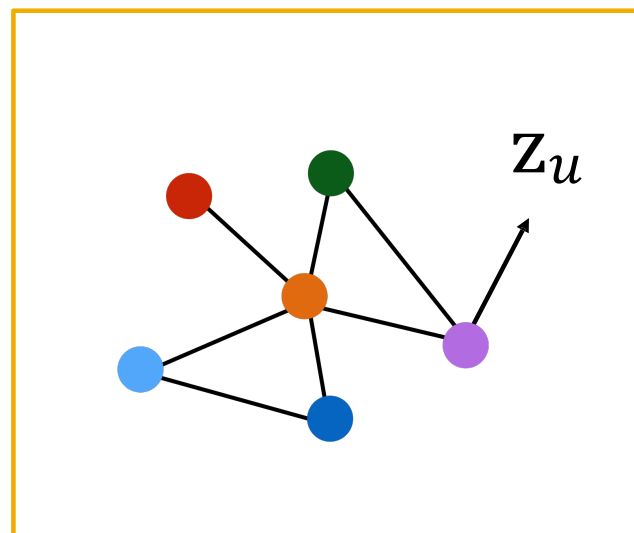
模型优势：归纳能力 (Inductive Capability)

□ 归纳式节点嵌入：可以泛化到完全未见过的图：

- 例如：在模式生物A的蛋白质相互作用图上进行训练，并在新采集的生物B的数据上生成嵌入表示。

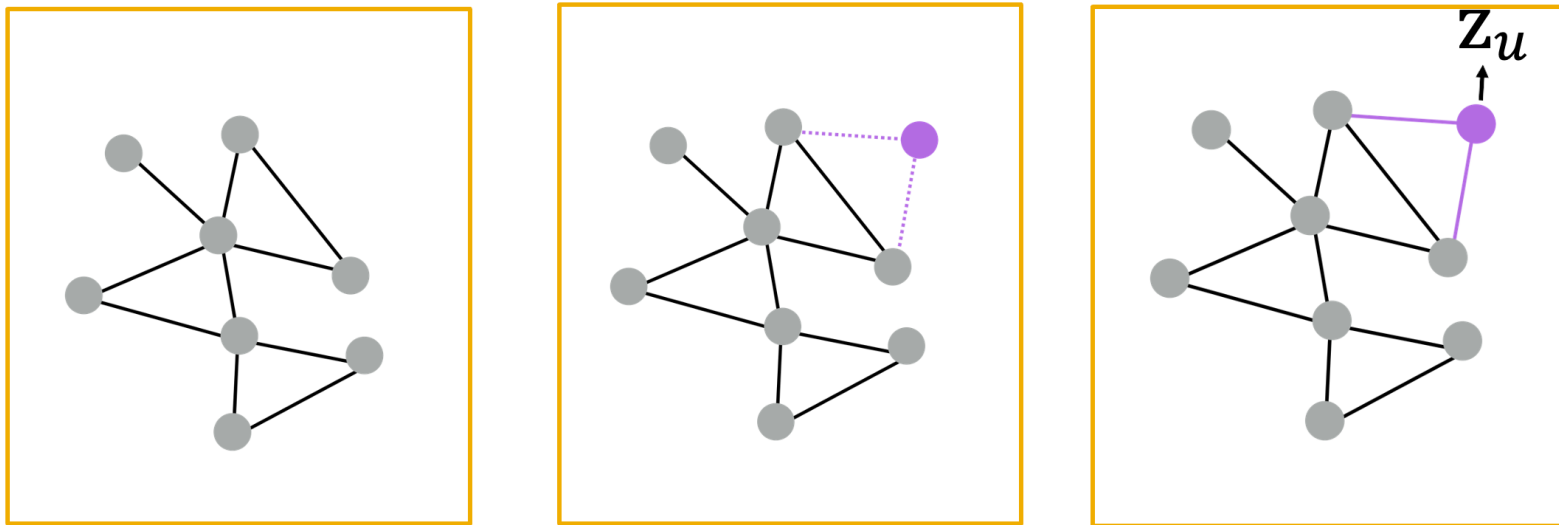


图A上训练



可泛化到图B

模型优势：归纳能力 (Inductive Capability)



□ 许多应用场景会持续遇到以前未见过的节点：

- 例如：Reddit、YouTube、Google Scholar
- 需要“即时”生成新的嵌入表示