

Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心



异质图神经网络

李新帅

2022年09月25日

- 背景简介
- 基础概念
 - 异质图
 - 元路径
- 算法原理
 - GATNE
 - MAGNN
- 应用总结
- 参考文献

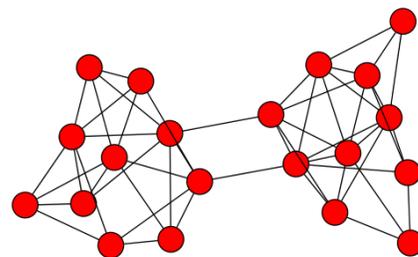
- 预期收获
 - 1. 了解图神经网络的基本原理
 - 2. 理解异质图神经网络的基本原理
 - 3. 理解异质图神经网络的应用
 - 4. 了解异质图神经网络的前沿发展

- 神经网络存在不足

- 神经网络算法不适用于**非欧式结构**数据，脱离现实应用场景（如社交网络、物理系统等）。



欧式结构数据



非欧式结构数据

- 图表示方法存在不足

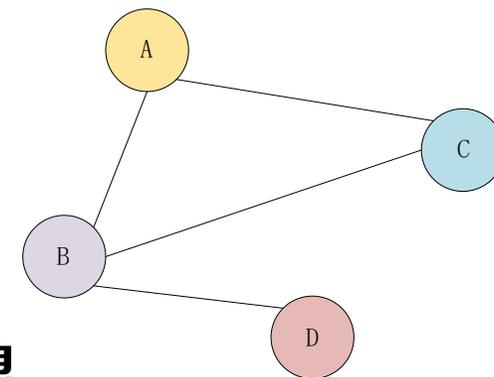
- 浅层的图表示方法，没有**参数共享**，每个节点都有自己独特的嵌入向量；
- 浅层的图表示方法，缺乏**泛化能力**；
- 浅层的图表示方法，未利用**节点特征**。

• 基本概念

- 图神经网络（Graph Neural Networks, GNN）是学习图结构数据的深度学习网络
- 提取和发掘图结构数据中的节点特征、边特征

• 研究对象

- 图是图神经网络研究的基本对象； $G = (V, E)$
- 是描述复杂事务的数据表示形式，由节点和边组成；
- 可描述不规则数据（非欧式数据），充分利用数据间关系信息

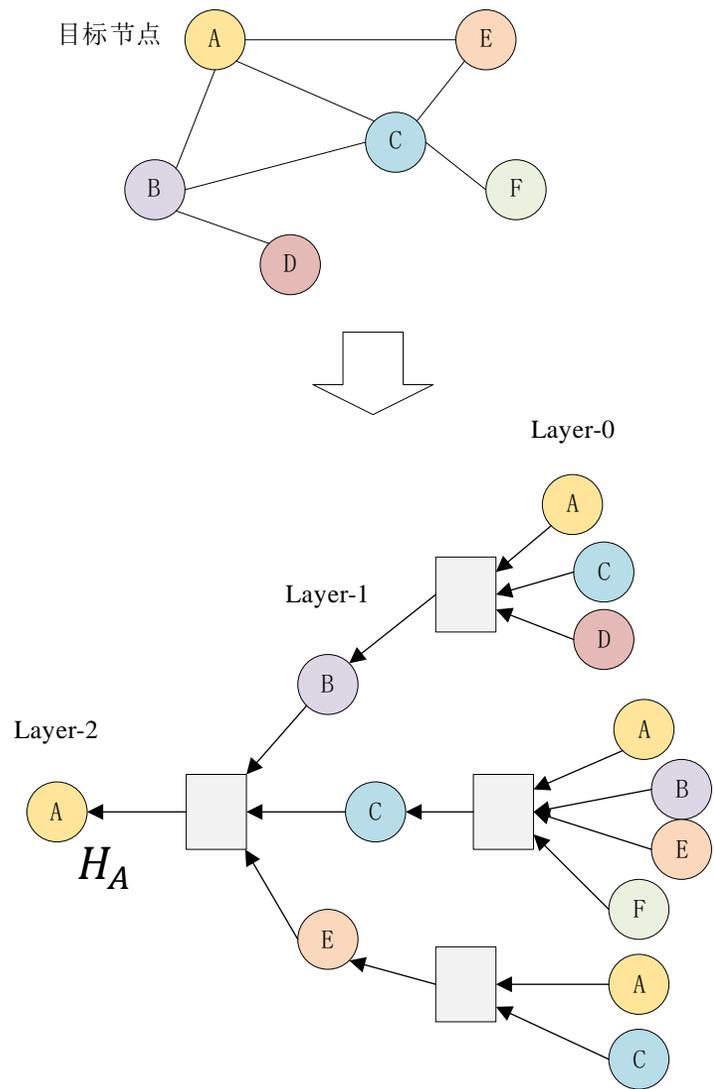


• 核心思想

- 使用神经网络聚合来自其邻居的信息，生成节点嵌入表示

• TIPO

T 目的	基于局部邻居节点特征，学习节点的状态表示 h_v
I 输入	图结构数据（节点、边）
P 过程	<ol style="list-style-type: none"> 1.基于邻居节点特征，迭代更新节点状态表示h_v，直到收敛时间步T，此时 $H^T \approx H$; 2.根据损失函数计算权重梯度; 3.依据梯度值更新权重参数。
O 输出	稳定的节点状态表示向量 h_v



邻居聚合

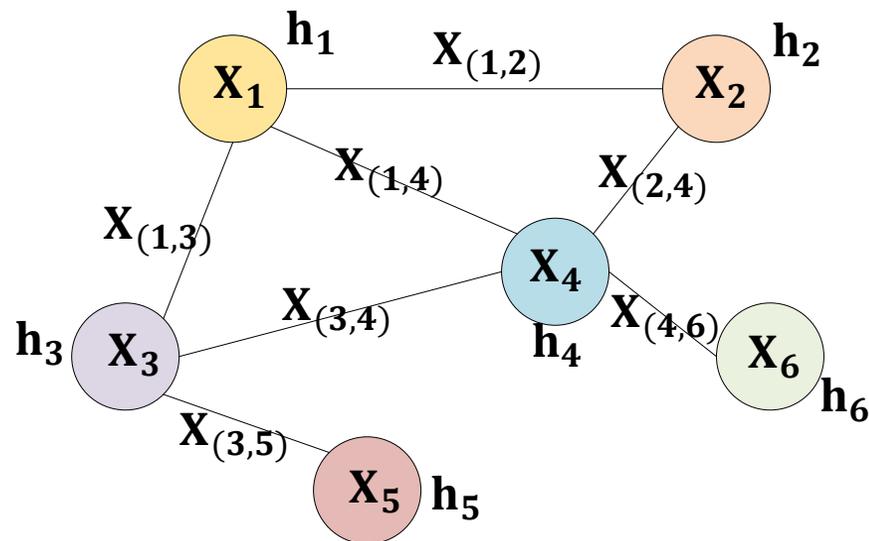


• 转移函数 (Transition function)

- 根据输入的节点邻居更新节点状态, 带参数的函数 f 全部节点**共享**

$$h_v = f(X_v, X_{co[v]}, h_{ne[v]}, X_{ne[v]})$$

- X 和 h 分别表示**输入特征**和**隐状态**;
- $co[v]$ 和 $ne[v]$ 分别表示和节点 v 相连的**边的集合**和**节点集合**。
- X_v 、 $X_{co[v]}$ 、 $h_{ne[v]}$ 、 $X_{ne[v]}$ 表示**节点特征**、**该节点的边的特征**、**该节点相邻节点的隐状态**、**以及该节点相邻节点的特征**。



$$h_3 = f(X_3, X_{(1,3)}, X_{(3,4)}, X_{(3,5)}, h_1, h_4, h_5, X_1, X_4, X_5)$$

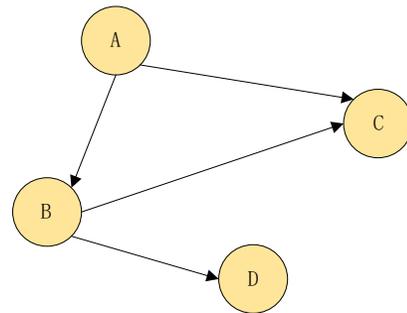
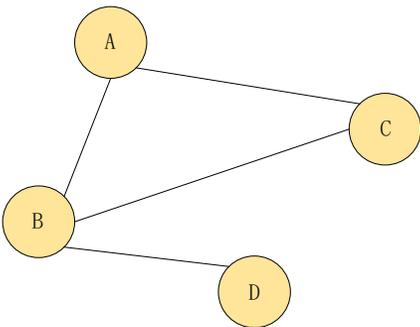
• 输出函数 (Output function)

$$O_v = \sigma(h_v, X_v)$$

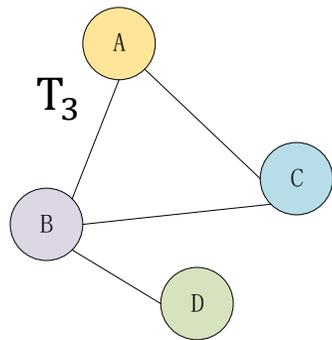
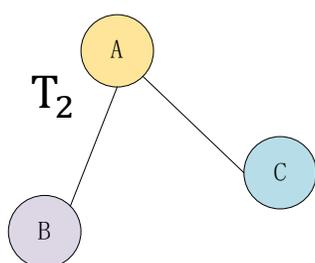
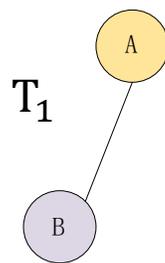
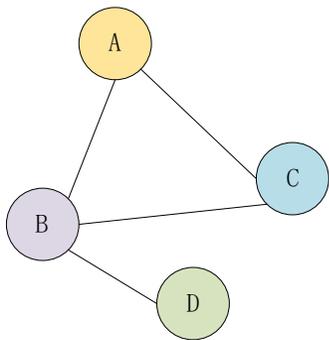
$$aggregate(h_i) = W \sum_{j \in N(i)} \frac{h_j}{|N(i)|} + B h_i$$

图论分类

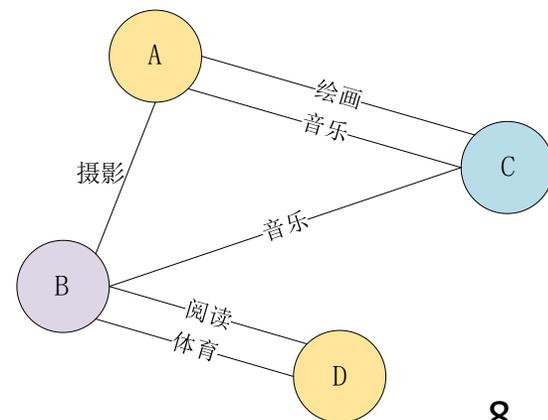
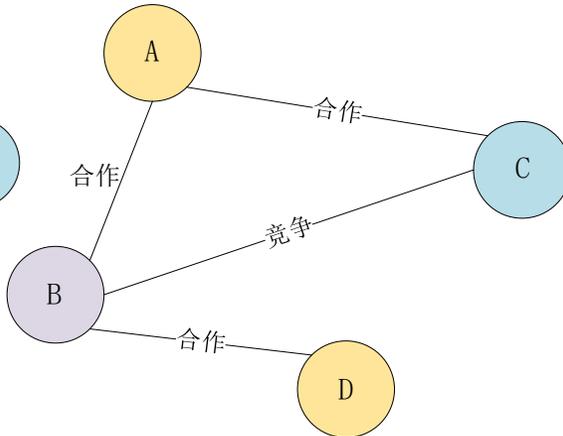
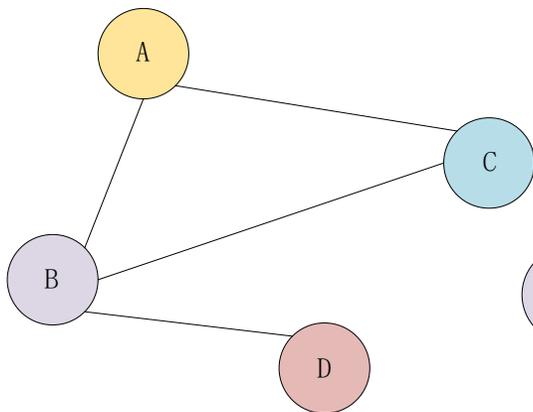
- 图的分类
 - 无向图、有向图



- 静态图、动态图



- 同质图、异质图



异质图神经网络 优势



- 同质图

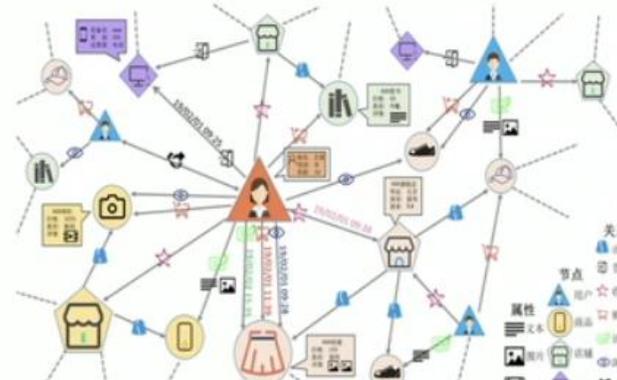
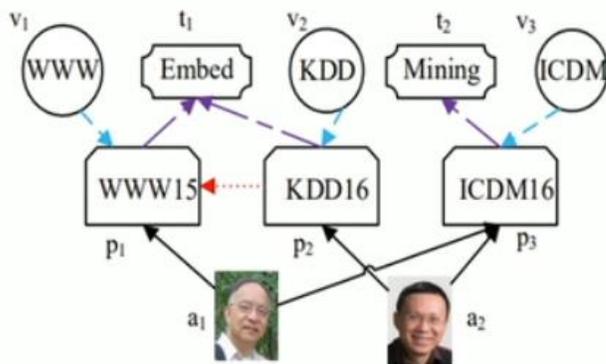
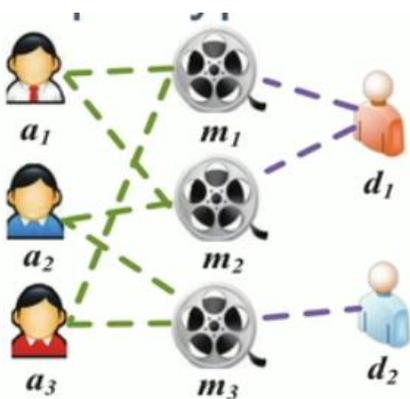
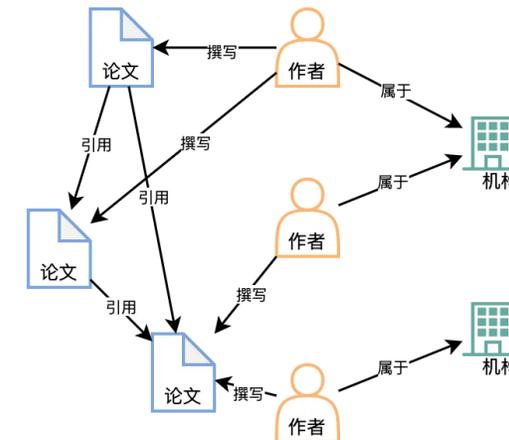
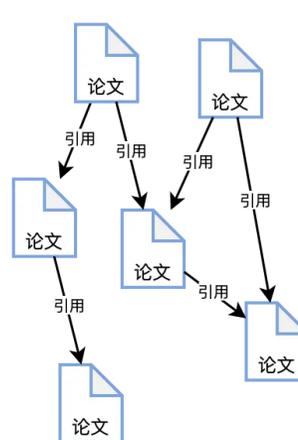
- 节点类型和边的类型只有一种的图

- 异质图

- 节点类型+边的类型 > 2

- 优势

- 将不同属性的数据合到一起，加入更多的语义信息数据
 - 结合node和relation信息来提高表征能力



- 定义

- 首先明确异质图的定义，相比于以往的同质图 $G = (V, E)$ ，异质图多了两个属性 R 、 T ，其中 R 表示边的类型， T 表示节点的类型，可以表示为：

$$G = (V, E, R, T)$$

- 同质图的聚合公式：

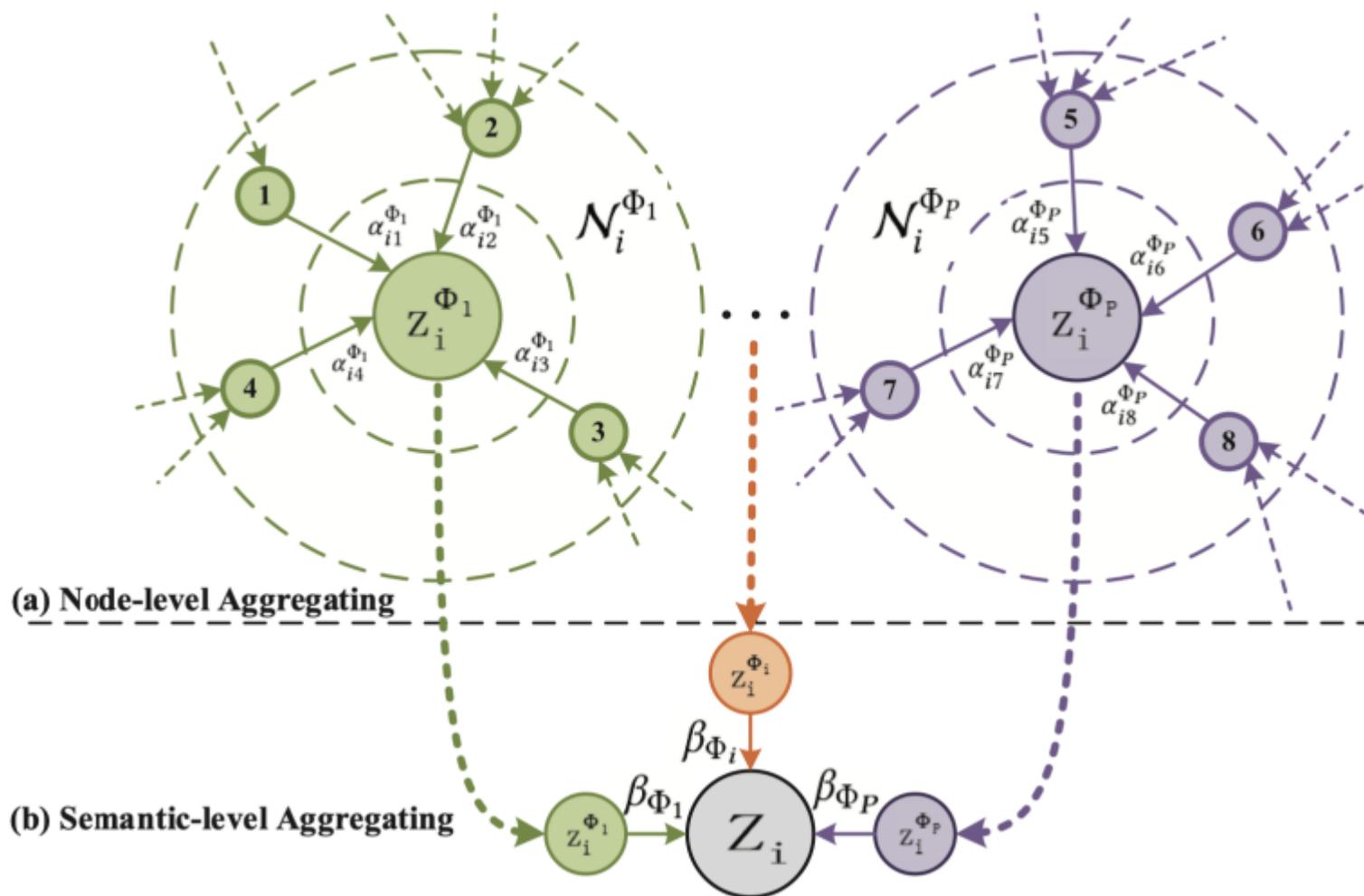
$$h_i^{l+1} = \sigma \left(\sum_{j \in N(i)} W^{l+1} \frac{h_j^l}{|N(i)|} \right)$$

- 异质图的聚合公式：

$$h_i^{l+1} = \sigma \left(\sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^l h_j^l + W_0^l h_i^l \right)$$

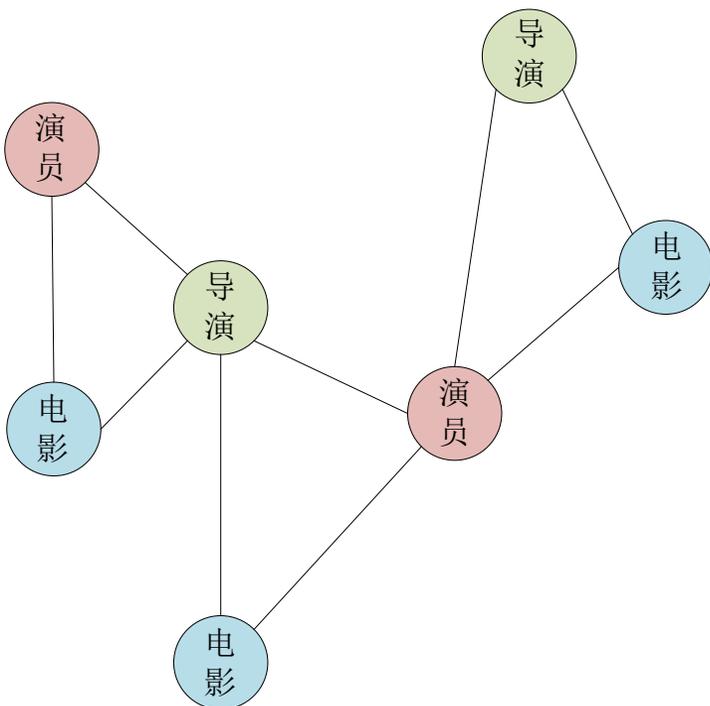
节点插播

- 一般步骤
 - 节点级别聚合
 - 语义级别聚合

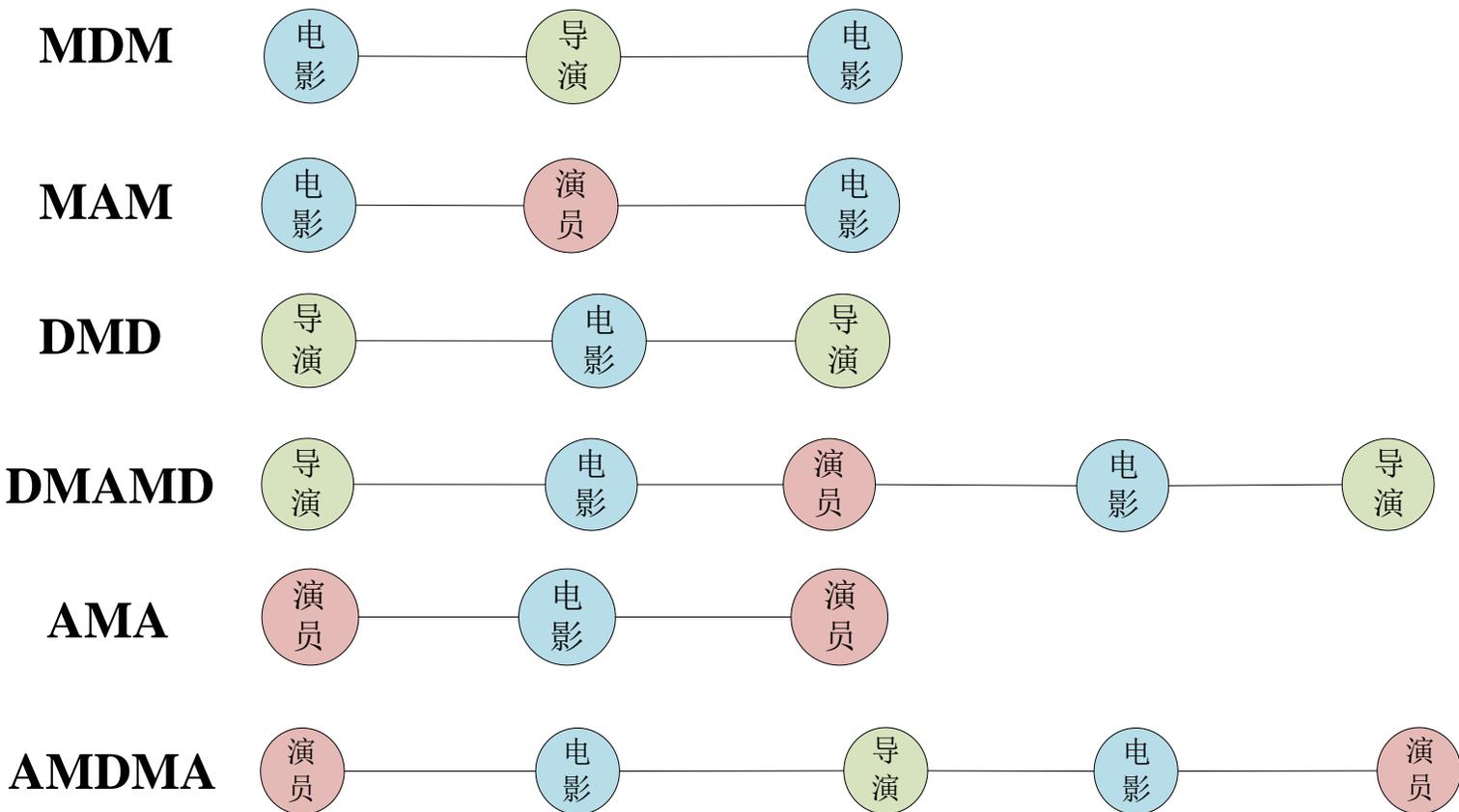


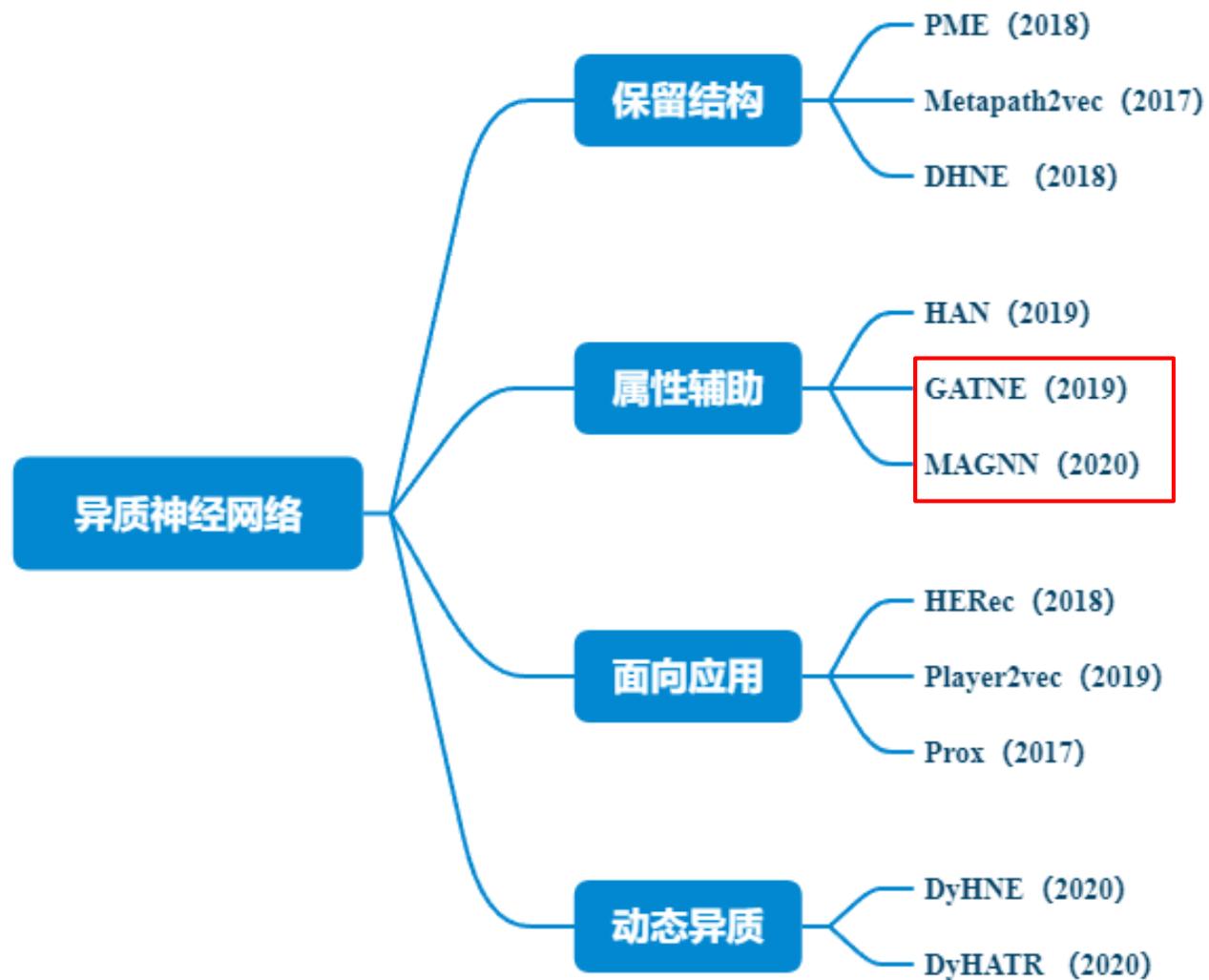
- 元路径

- 连接不同类型节点的一条路径
- 不同的元路径会有不同的路径类型，通常用节点类型路径来表示



电影 (M) 导演 (D) 演员 (A)





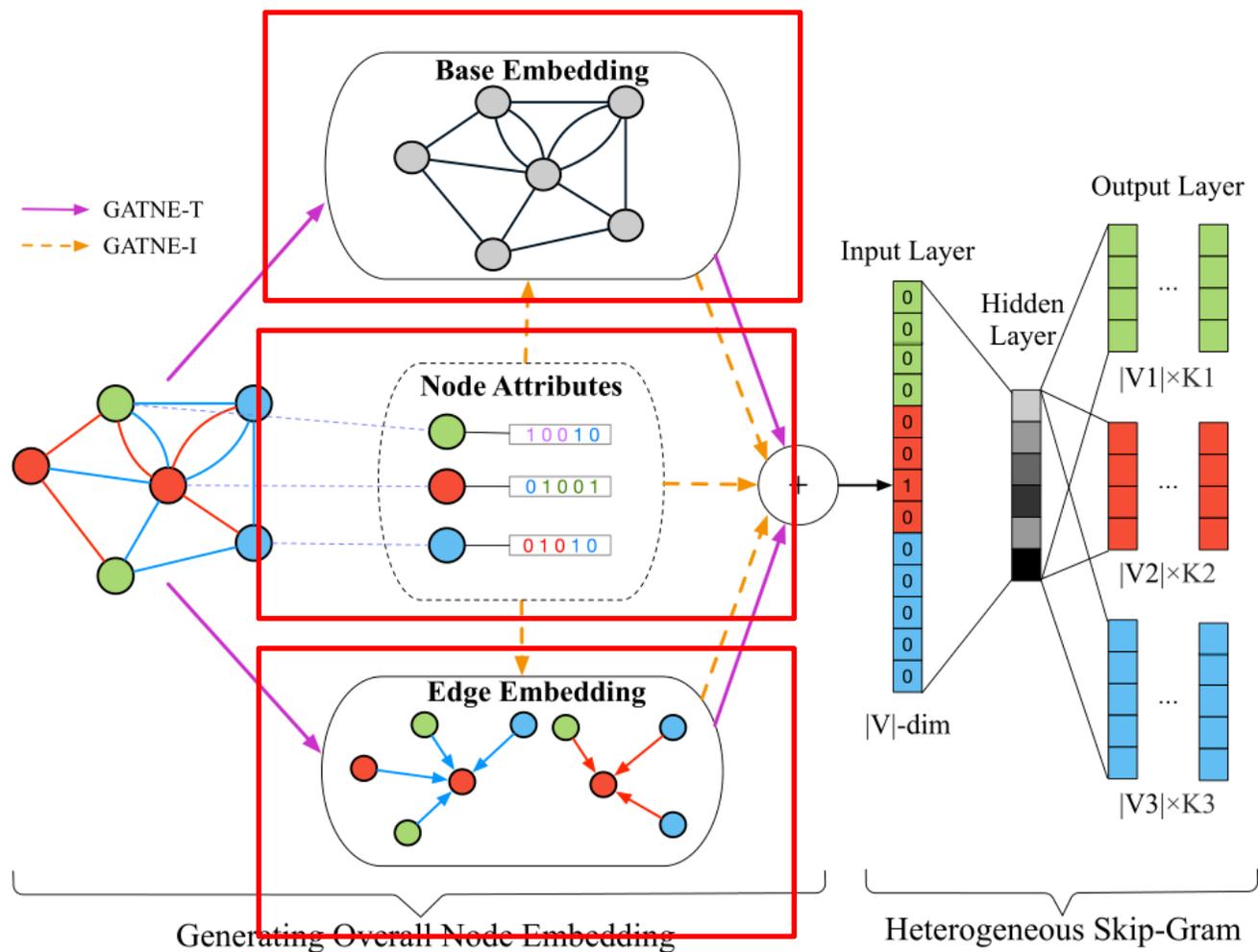


GATNE

T	基于邻域节点和边特征，生成节点嵌入向量
I	具有异质、属性特征的图结构数据
P	1.基本图结构嵌入学习； 2.节点属性嵌入学习； 3.异构边特征嵌入学习； 4.模型优化。
O	节点嵌入表示向量

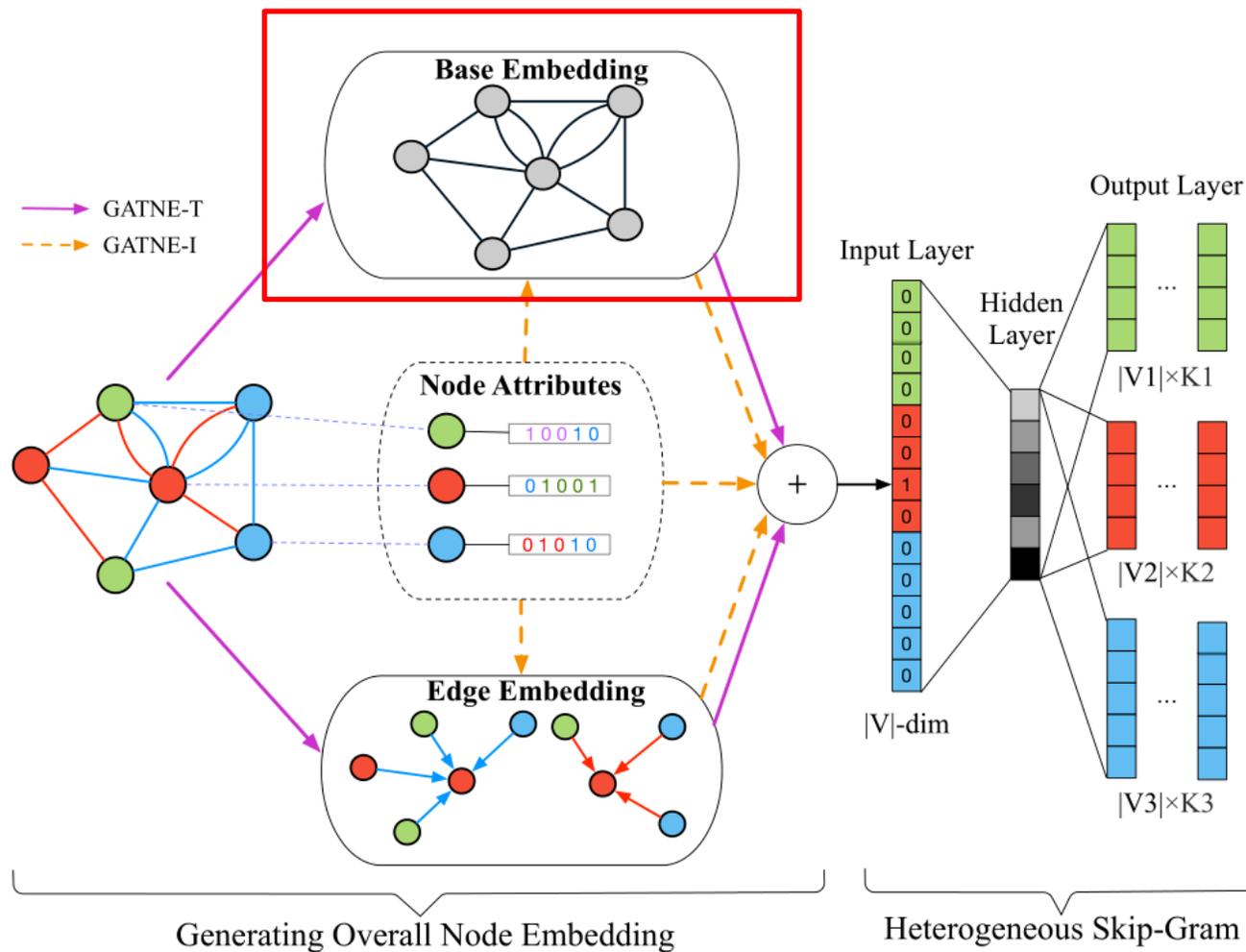
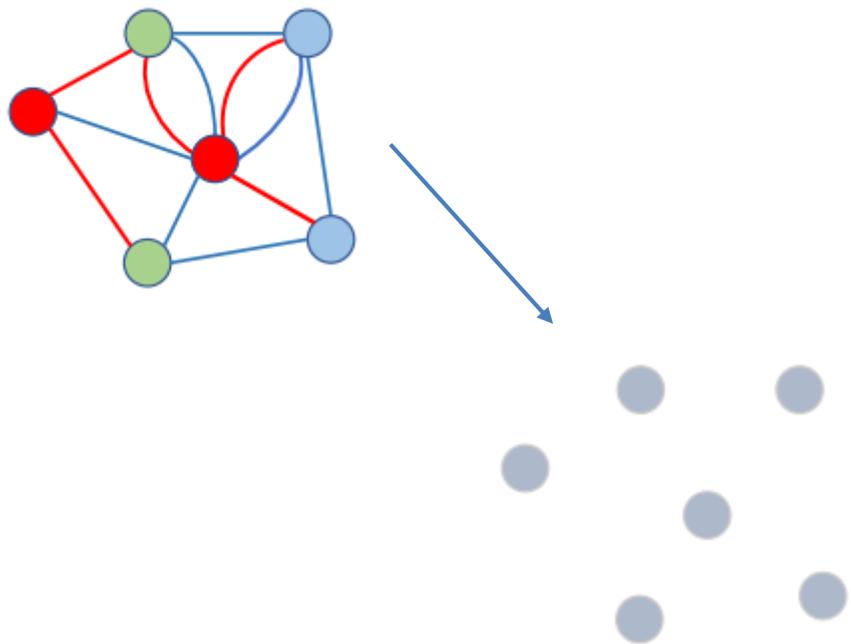
P	节点属性和边属性的特征表示
C	已知网络的异构属性信息
D	如何利用节点属性和边属性学习节点的特征表示
L	KDD 2019 (CCF A)

• 算法原理图



- 初始特征嵌入向量
 - 节点自身特征 x_i 的函数

$$b_i = h_z(x_i)$$



聚合计算方法

- 节点边特征嵌入向量，节点 v_i 在边类别 r 下的初始边嵌入向量 $\mathbf{u}_{i,r}^{(0)}$ 同样为 x_i 的函数

$$\mathbf{u}_{i,r}^{(0)} = g_{z,r}(x_i)$$

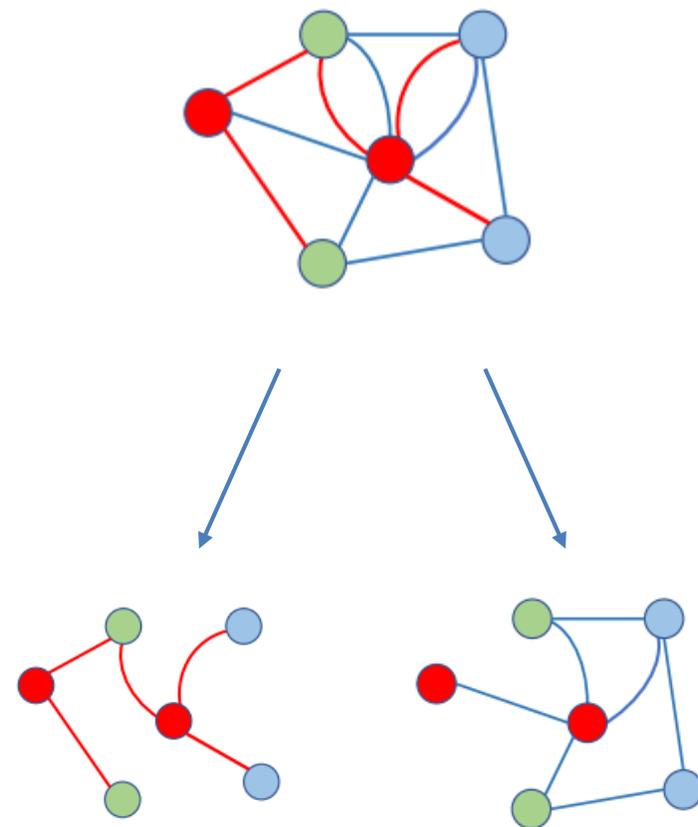
- 信息传递：相邻节点，**相同类型边**的聚合

$$\mathbf{u}_{i,r}^{(k)} = \text{aggregator} \left(\mathbf{u}_{j,r}^{(k-1)}, \forall v_j \in N_{i,r} \right)$$

- *aggregator* 的方式可采用**平均聚合器** (mean) 或**最大池化聚合器** (pooling)

$$\mathbf{u}_{i,r}^{(k)} = \sigma \left(\widehat{\mathbf{W}}^k \cdot \text{mean} \left(\left\{ \mathbf{u}_{j,r}^{(k-1)}, \forall v_j \in N_{i,r} \right\} \right) \right)$$

$$\mathbf{u}_{i,r}^{(k)} = \max \left(\left\{ \widehat{\mathbf{W}}_{pool}^k \mathbf{u}_{j,r}^{(k-1)} + \hat{b}_{pool}^k, \forall v_j \in N_{i,r} \right\} \right)$$



不同边类型聚合计算

- 不同边类型聚合计算

- 模型定义第K层 $u_{i,r}^{(K)}$ 为边嵌入 $u_{i,r}$ 向量，拼接 v_i 节点所有边嵌入向量 $u_{i,r}$ 生成 U_i

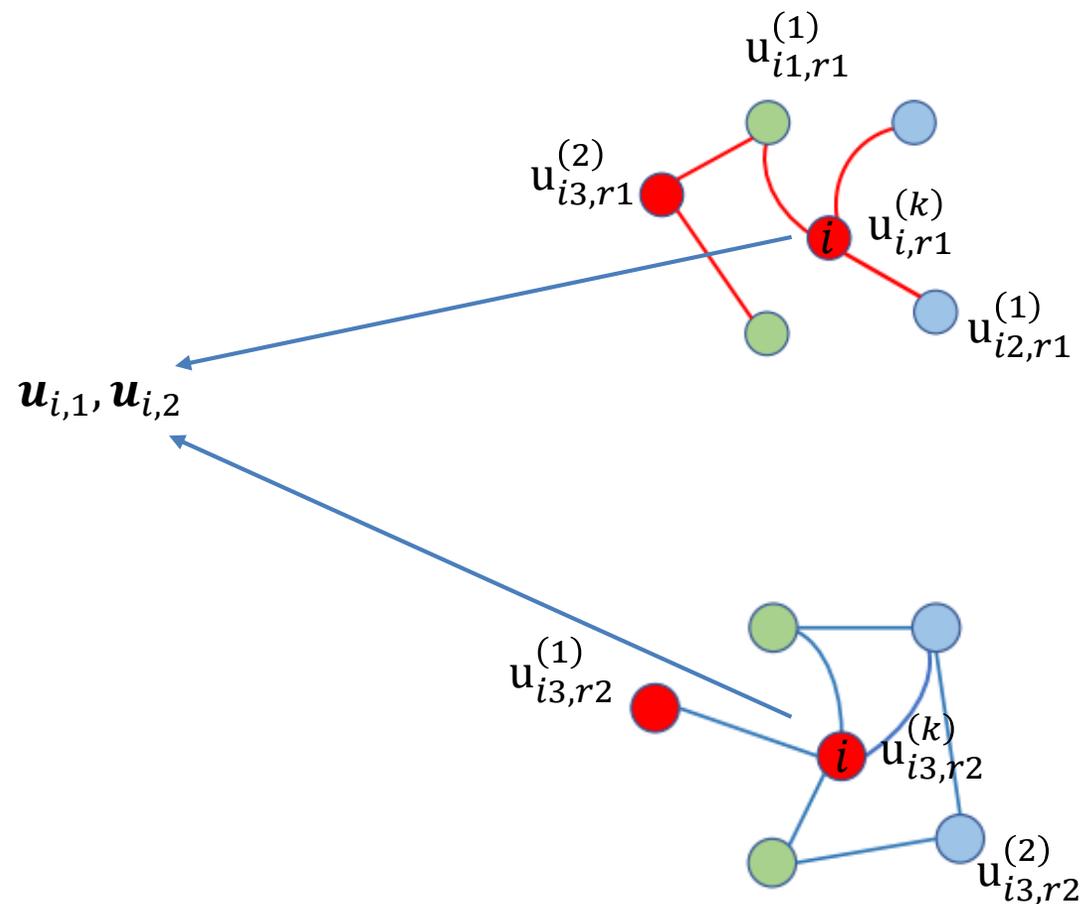
$$U_i = (u_{i,1}, u_{i,2}, \dots, u_{i,m})$$

- 使用自注意力机制计算各边嵌入的权重系数

$$\alpha_{i,r} = \text{softmax}(\mathbf{w}_r^T \tanh(\mathbf{W}_r U_i))^T$$

- 聚合所有边嵌入向量，生成节点 v_i 的最终嵌入向量

$$v_i = b_i + M^T U_i \alpha_{i,r}$$



实验设计：节点异构的影响

• 实验设计

- 数据集：3个公开数据集 Amazon、YouTube、Twitter 和阿里数据集（user-item访问、购买、点击、加购关系）进行实验

Dataset	# nodes	# edges	# n-types	# e-types
Amazon	10,166	148,865	1	2
YouTube	2,000	1,310,617	1	5
Twitter	10,000	331,899	1	4
Alibaba (small)	6,184	18,120	2	4
Alibaba	41,991,048	571,892,183	2	4

	正例	负例
预测正	真正例 (TP)	假正例 (FP)
预测负	假负例 (FN)	真负例 (TN)

– 评价指标：

- ROC-AUC: ROC 曲线下的面积
- PR-AUC: PR 曲线下的面积
- F1: 调和平均值

$$TPR = \frac{TP}{TP+FN} \quad \text{真正例率}$$

$$FPR = \frac{FP}{FP+TN} \quad \text{假正例率}$$

$$P = \frac{TP}{TP+FP} \quad \text{精确率}$$

$$R = \frac{TP}{TP+FN} \quad \text{召回率}$$

$$F1 = \frac{2PR}{P+R}$$



实验结果（链接预测）

	Amazon			YouTube			Twitter			Alibaba-S		
	ROC-AUC	PR-AUC	F1									
DeepWalk	94.20	94.03	87.38	71.11	70.04	65.52	69.42	72.58	62.68	59.39	60.62	56.10
node2vec	94.47	94.30	87.88	71.21	70.32	65.36	69.90	73.04	63.12	62.26	63.40	58.49
LINE	81.45	74.97	76.35	64.24	63.25	62.35	62.29	60.88	58.18	53.97	54.65	52.85
metapath2vec	94.15	94.01	87.48	70.98	70.02	65.34	69.35	72.61	62.70	60.94	61.40	58.25
ANRL	71.68	70.30	67.72	75.93	73.21	70.65	70.04	67.16	64.69	58.17	55.94	56.22
PMNE(n)	95.59	95.48	89.37	65.06	63.59	60.85	69.48	72.66	62.88	62.23	63.35	58.74
PMNE(r)	88.38	88.56	79.67	70.61	69.82	65.39	62.91	67.85	56.13	55.29	57.49	53.65
PMNE(c)	93.55	93.46	86.42	68.63	68.22	63.54	67.04	70.23	60.84	51.57	51.78	51.44
MVE	92.98	93.05	87.80	70.39	70.10	65.10	72.62	73.47	67.04	60.24	60.51	57.08
MNE	90.28	91.74	83.25	82.30	82.18	75.03	91.37	91.65	84.32	62.79	63.82	58.74
GATNE-T	97.44	97.05	92.87	84.61	81.93	76.83	92.30	91.77	84.96	66.71	67.55	62.48
GATNE-I	96.25	94.77	91.36	84.47	82.32	76.83	92.04	91.95	84.38	70.87	71.65	65.54

- GATNE-T: 直推式 (Transductive) 学习
- GATNE-I: 归纳式 (Inductive) 学习

- 优势
 - GATNE 算法考虑利用额外的**属性**信息，学习节点表示
 - 解决**冷启动**问题
- 劣势
 - 在**缺乏节点属性**的数据集中GATNE-I模型效果不佳
 - 划分子图，过于稀疏，更适合一对多



MAGNN

TIPO

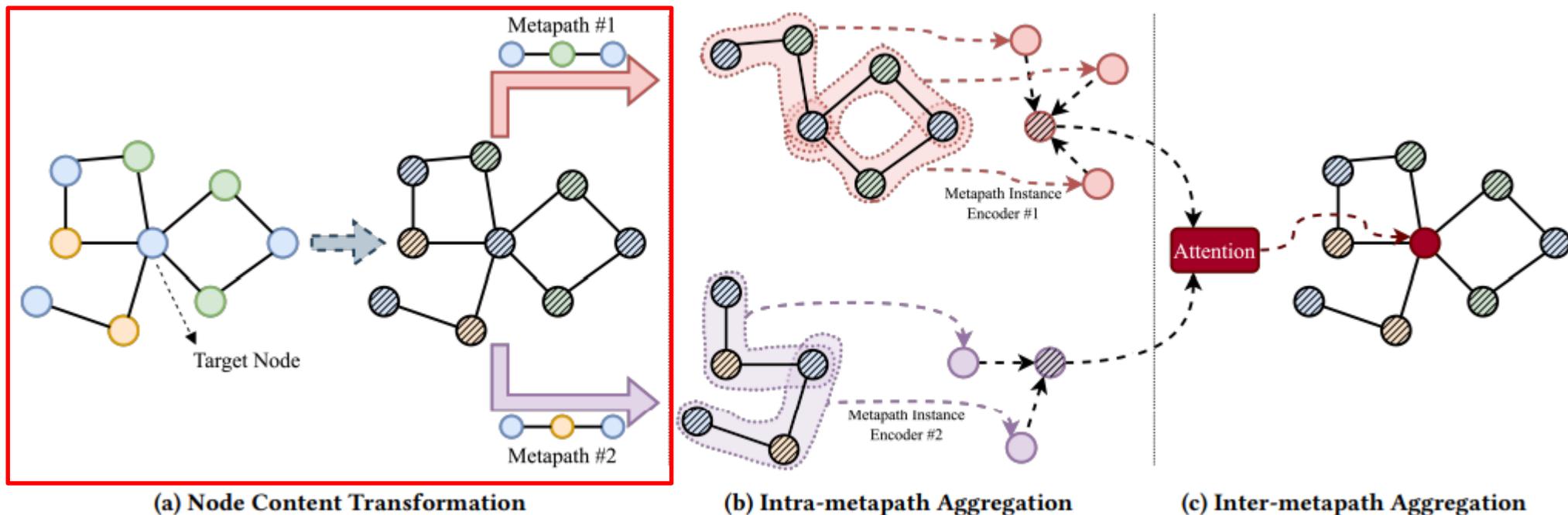
T	对异构图网络学习嵌入向量表示
I	含有节点属性的异构图神经网络
P	1.节点属性转换 2.元路径内聚合 3.元路径间聚合
O	节点嵌入向量表示

P	如何有效利用元路径学习节点表示
C	节点属性、节点类型、预先定义好的元路径
D	元路径的选取，嵌入的准确性
L	WWW 2020 (CCF A)

- 节点属性转换

- 原始特征向量为 x_v^A ，映射后向量表示为 h'_v ，在类型A下的节点参数权重矩阵表示为 W_A ，具体过程如下

$$h'_v = W_A \cdot x_v^A$$



- 中间节点表示学习

$$h_{p(v,u)} = f_{\theta}(P(v,u)) = f_{\theta}\left(h'_v, h'_u, \{h'_t, \forall t \in \{m^{P(v,u)}\}\}\right)$$

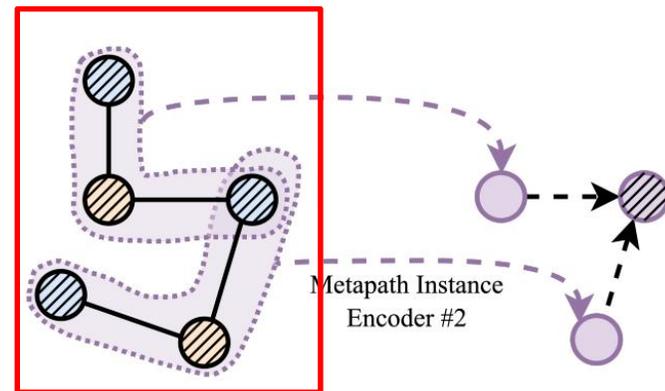
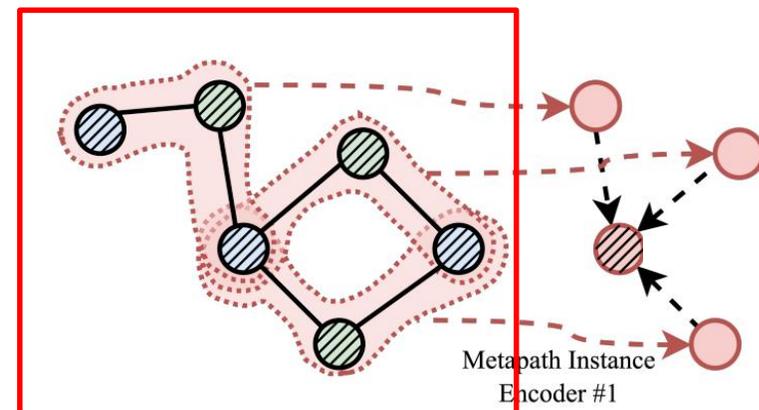
- 其中, $\{m^{P(v,u)}\} = P(v,u) \setminus \{u,v\}$, 用来表示 u, v 的中间节点

- 平均编码

$$h_{p(v,u)} = MEAN(\{h'_t, \forall t \in P(v,u)\})$$

- 线性编码

$$h_{p(v,u)} = W_p \cdot MEAN(\{h'_t, \forall t \in P(v,u)\})$$



- 图注意力网络 (Graph Attention Networks, GAT)

- 计算元路径实例 $p(v, u)$ 对节点 v 的重要性

$$e_{vu}^P = \text{LeakyRelu}(a_P^T \cdot [h'_v \parallel h_{p(v,u)}])$$

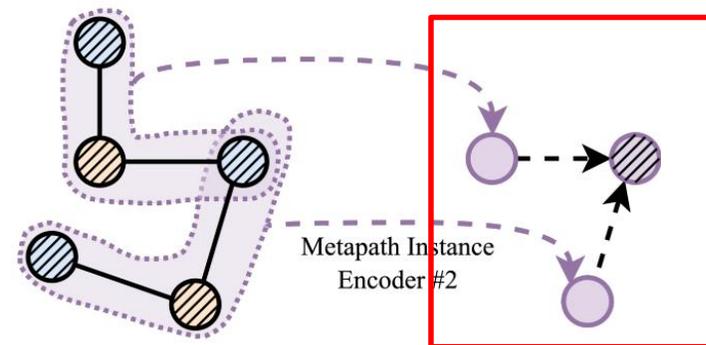
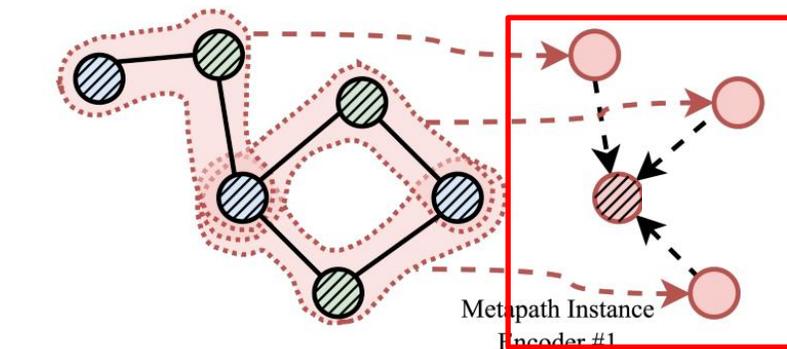
- 其中, P 表示某一元路径实例

- 归一化的重要性权重 α_{vu}^P

$$\alpha_{vu}^P = \frac{\exp(e_{vu}^P)}{\sum_{s \in N_v^P} \exp(e_{vs}^P)}$$

- 状态更新公式

$$h_v^P = \sigma \left(\sum_{u \in N_v^P} \alpha_{vu}^P \cdot h_{p(v,u)} \right)$$



- 元路径间聚合

- 对特定类型下的节点进行转换，并取平均

$$s_{P_i} = \frac{1}{|\mathcal{V}_A|} \sum_{v \in \mathcal{V}_A} \tanh(M_A \cdot h_v^{P_i} + b_A)$$

- 其中， $\mathcal{V}_A = \{h_v^{P_1}, h_v^{P_2}, \dots, h_v^{P_M}\}$

- 使用注意力计算特定元路径下的节点 v 的特征向量

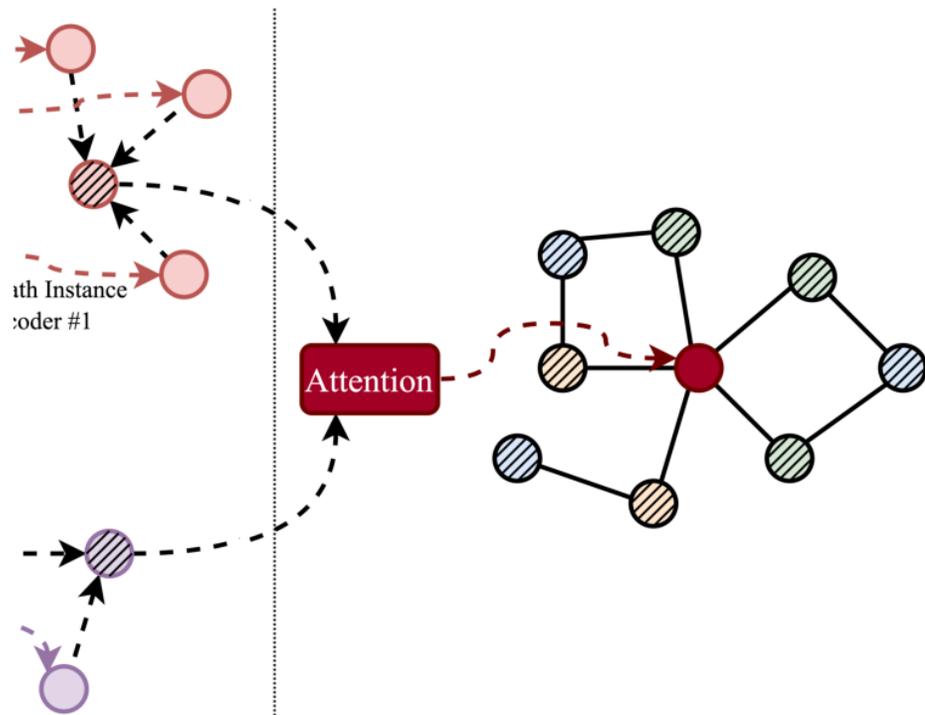
$$e_{P_i} = q_A^T \cdot s_{P_i}$$

$$\beta_{P_i} = \frac{\exp(e_{P_i})}{\sum_{P \in \mathcal{P}_A} \exp(e_P)}$$

$$h_v^{\mathcal{P}_A} = \sum_{P \in \mathcal{P}_A} \beta_P \cdot h_v^P$$

- 输出函数计算

$$h_v = \sigma(W_o \cdot h_v^{\mathcal{P}_A})$$



- 实验数据

- IMDb

- 这是一个电影分级网络，记录用户对不同电影的偏好
 - 每部电影都包含导演、演员和类型

- DBLP

- 这是一个反映作者和论文之间关系的网络
 - 节点有四种类型：作者、论文、术语和地点

- Last.fm

- 这是一个音乐网站，跟踪用户从不同来源获取的收听信息
 - 节点有三种类型：用户、艺术家、艺术家标签

Dataset	Node	Edge	Metapath
IMDb	# movie (M): 4,278 # director (D): 2,081 # actor (A): 5,257	# M-D: 4,278 # M-A: 12,828	MDM MAM DMD DMAMD AMA AMDMA
DBLP	# author (A): 4,057 # paper (P): 14,328 # term (T): 7,723 # venue (V): 20	# A-P: 19,645 # P-T: 85,810 # P-V: 14,328	APA APTPA APVPA
Last.fm	# user (U): 1,892 # artist (A): 17,632 # tag (T): 1,088	# U-U: 12,717 # U-A: 92,834 # A-T: 23,253	UU UAU UATAU AUA AUUA ATA

• 节点分类

Dataset	Metrics	Train %	Unsupervised					Semi-supervised			
			LINE	node2vec	ESim	metapath2vec	HERec	GCN	GAT	HAN	MAGNN
IMDb	Macro-F1	20%	44.04	49.00	48.37	46.05	45.61	52.73	53.64	56.19	59.35
		40%	45.45	50.63	50.09	47.57	46.80	53.67	55.50	56.15	60.27
		60%	47.09	51.65	51.45	48.17	46.84	54.24	56.46	57.29	60.66
		80%	47.49	51.49	51.37	49.99	47.73	54.77	57.43	58.51	61.44
	Micro-F1	20%	45.21	49.94	49.32	47.22	46.23	52.80	53.64	56.32	59.60
		40%	46.92	51.77	51.21	48.17	47.89	53.76	55.56	57.32	60.50
		60%	48.35	52.79	52.53	49.87	48.19	54.23	56.47	58.42	60.88
		80%	48.98	52.72	52.54	50.50	49.11	54.63	57.40	59.24	61.53
DBLP	Macro-F1	20%	87.16	86.70	90.68	88.47	90.82	88.00	91.05	91.69	93.13
		40%	88.85	88.07	91.61	89.91	91.44	89.00	91.24	91.96	93.23
		60%	88.93	88.69	91.84	90.50	92.08	89.43	91.42	92.14	93.57
		80%	89.51	88.93	92.27	90.86	92.25	89.98	91.73	92.50	94.10
	Micro-F1	20%	87.68	87.21	91.21	89.02	91.49	88.51	91.61	92.33	93.61
		40%	89.25	88.51	92.05	90.36	92.05	89.22	91.77	92.57	93.68
		60%	89.34	89.09	92.28	90.94	92.66	89.57	91.97	92.72	93.99
		80%	89.96	89.37	92.68	91.31	92.78	90.33	92.24	93.23	94.47

– 元路径实例比基于元路径的邻居包含更丰富的信息，证明元路径内聚合的有效性

• 链接预测

Dataset	Metrics	LINE	node2vec	ESim	metapath2vec	HERec	GCN	GAT	GATNE	HAN	MAGNN
Last.fm	AUC	85.76	67.14	82.00	92.20	91.52	90.97	92.36	89.21	93.40	98.91
	AP	88.07	64.11	82.19	90.11	89.47	91.65	91.55	88.86	92.44	98.93

– 较单一元路径有更多的语义信息被考虑



总结

- 优势
 - MAGNN和GATNE算法均考虑利用额外的属性信息
 - 考虑了元路径上的所有中间节点，丰富属性信息
 - 考虑了多种元路径对节点的影响
- 劣势
 - 元路径**预定义**，无法自主学习元路径
- 发展前沿
 - 推荐任务
 - 自主学习元路径、不利用元路径：GTNs、Simple-Han

- [1] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation Learning for Attributed Multiplex Heterogeneous Network. In SIGKDD. 1358–1368.
- [2] Xinyu Fu, Jiani Zhang, Ziqiao Meng, Irwin King. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. Proceedings of the WWW, 2020.
- [3] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, Hyunwoo J.Kim. 2019. Graph Transformer Networks. In NeurIPS.
- [4] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Yizhou Sun. 2020. Heterogeneous Graph Transformer. 2020. In WWW.

谢谢!

大成若缺，其用不弊。大盈
若冲，其用不穷。大直若屈。
大巧若拙。大辩若讷。静胜
躁，寒胜热。清静为天下正。

