



# 分层对比学习增强异构图神经网络



讲者：王培宇





## 论文介绍

- Liu N, Wang X, Han H, et al. Hierarchical Contrastive Learning Enhanced Heterogeneous Graph Neural Network[J]. IEEE Transactions on Knowledge and Data Engineering, 2023.



**Nian Liu** received the B.E. degree in 2020 from Beijing University of Posts and Telecommunications. He is a second-year M.S. student in the Department of Computer Science of Beijing University of Posts and Telecommunications. His main research interests including graph mining and contrastive learning.



**Xiao Wang** is an Associate Professor in the School of Computer Science, Beijing University of Posts and Telecommunications. He received his Ph.D. degree from the School of Computer Science and Technology, Tianjin University, Tianjin, China, in 2016. He was a post-doctoral researcher in Department of Computer Science and Technology, Tsinghua University, Beijing, China. His current research interests include data mining, social network analysis, and machine learning. Until now, he has published more than 70 papers in refereed journals and conferences.



# Motivation

- 异质信息网络(HIN)在真实世界中非常普遍，近年来，异质图神经网络方法(HGNNs)在处理HIN数据方面获得了巨大成功。
- 大多数的HGNN采用半监督的范式，而真实数据的标签在许多场景下是难以获得的。
- 对比学习方法能够较好的解决上述问题。

将对比学习引入异质图神经网络需要解决以下三个问题：

1. 如何设计异质的对比机制？单一视图的对比学习不能完善捕捉HIN的复杂信息，需要跨视图的对比学习。
2. 如何选择合适的视图？视图需要同时包含局部结构信息和高阶结构信息——网络模式视图(Network Schema View)和元路径视图(Metapath View)。
3. 如何设置一个困难的对比任务？增强两个视图的信息多样性，或者生成高质量的困难负样本。

Heco

# Method

## ➤ Preliminary

- Heterogeneous Information Network:

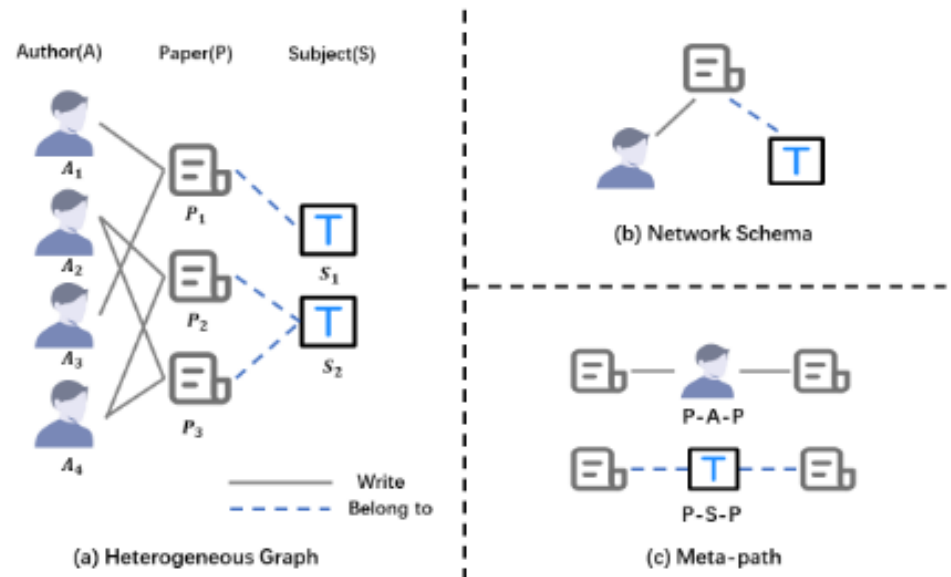
$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}, \phi, \varphi).$$

- Network Schema:

$$T_G = (\mathcal{A}, \mathcal{R})$$

- Meta-path:

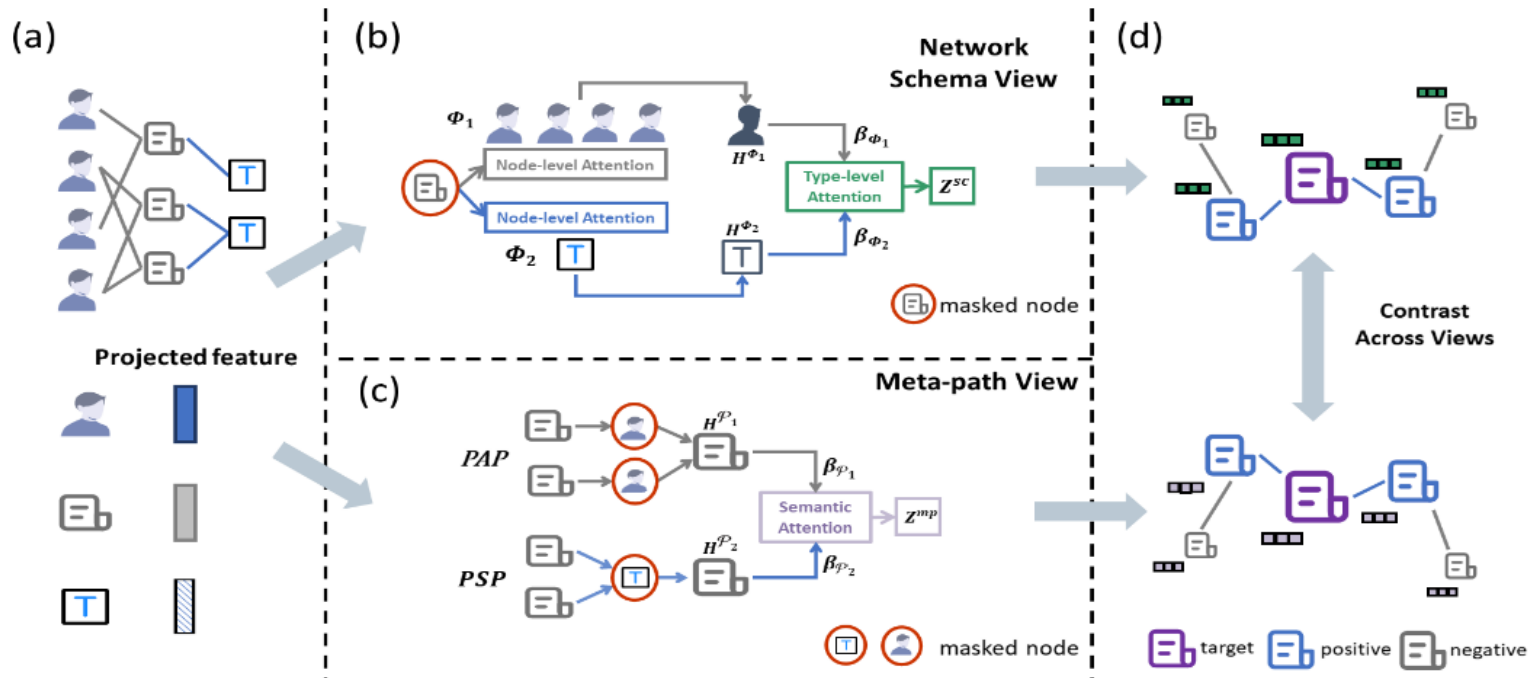
$$A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$$





# Method

## ➤ Overall Architecture





# Method

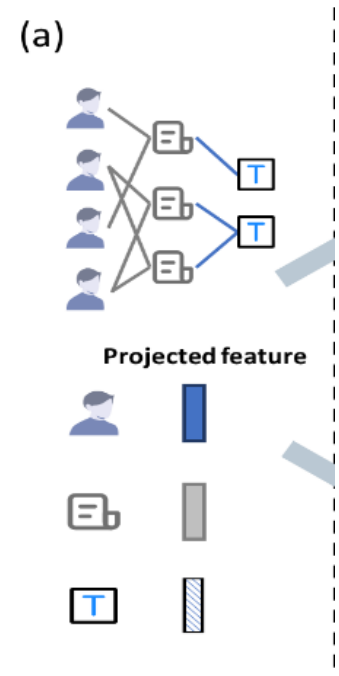
## ➤ Part 1: Node Feature Transformation

- 由于HIN中存在不同类型的节点，它们的特征通常位于不同的空间，因此首先需要将所有类型的节点特征投影到同一个向量空间。

- 文中对每一类型的节点使用一层MLP进行映射

$$h_i = \sigma(W_{\phi_i} \cdot x_i + b_{\phi_i})$$

- 其中 $h_i$ 是节点 $i$ 映射后的向量， $W_{\phi_i}$ 和 $b_{\phi_i}$ 是节点 $i$ 所对应类型 $\phi_i$ 的MLP层的参数。





# Method

## ➤ Part 2: Network Schema View Guided Encoder

- 假设节点*i*与*S*个其他类型的节点 $\{\phi_1, \phi_2, \dots, \phi_S\}$ 直接相连，则节点*i*的类型为 $\phi_m$ 的邻居节点总数可表示为 $N_i^{\phi_m}$ 。

- 首先对同类型的节点信息进行聚合——node-level attention

$$h_i^{\Phi_m} = \sigma \left( \sum_{j \in N_i^{\Phi_m}} \alpha_{i,j}^{\Phi_m} \cdot h_j \right), \quad \alpha_{i,j}^{\Phi_m} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}_{\Phi_m}^\top \cdot [h_i || h_j]))}{\sum_{l \in N_i^{\Phi_m}} \exp(\text{LeakyReLU}(\mathbf{a}_{\Phi_m}^\top \cdot [h_i || h_l]))}$$

- 然后对不同类型的信息进行聚合——type-level attention

$$w_{\Phi_m} = \frac{1}{|V|} \sum_{i \in V} \mathbf{a}_{sc}^\top \cdot \tanh(\mathbf{W}_{sc} h_i^{\Phi_m} + \mathbf{b}_{sc}),$$

$$\beta_{\Phi_m} = \frac{\exp(w_{\Phi_m})}{\sum_{i=1}^S \exp(w_{\Phi_i})}, \quad z_i^{sc} = \sum_{m=1}^S \beta_{\Phi_m} \cdot h_i^{\Phi_m}.$$

- 最终得到网络模式视图下节点*i*的表征 $z_i^{sc}$

# Method

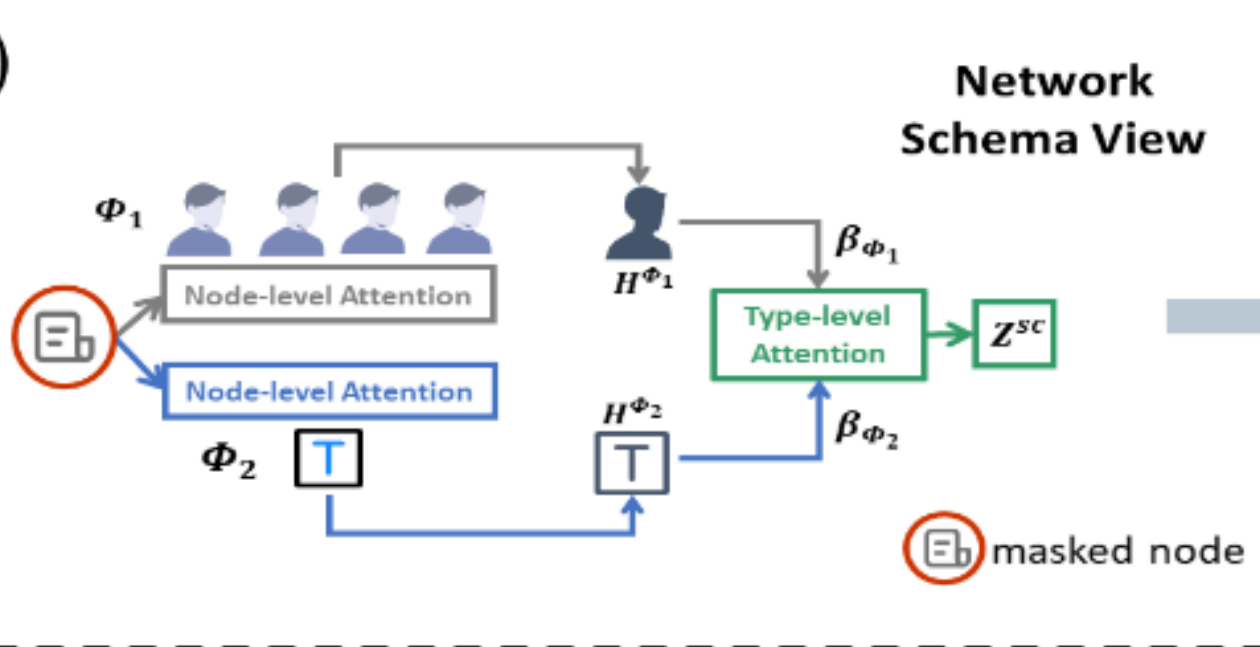
## ➤ Part 2: Network Schema View Guided Encoder

- 假设节点*i*与*S*个其他类型的节点 $\{\phi_1, \phi_2, \dots, \phi_S\}$ 直接相连，

(b)

- 盲

- 祭



- 最终得到网络模式视图下节点*i*的表征

$$z_i^{sc} = \sum_{m=1}^S \beta_{\Phi_m} \cdot h_i^{\Phi_m}.$$





# Method

## ➤ Part 2: Network Schema View Guided Encoder

- 假设节点*i*与*S*个其他类型的节点 $\{\phi_1, \phi_2, \dots, \phi_S\}$ 直接相连，则节点*i*的类型为 $\phi_m$ 的邻居节点总数可表示为 $N_i^{\phi_m}$ 。

- 首先对同类型的邻居节点信息进行聚合——node-level attention

$$h_i^{\Phi_m} = \sigma \left( \sum_{j \in N_i^{\Phi_m}} \alpha_{i,j}^{\Phi_m} \cdot h_j \right), \quad \alpha_{i,j}^{\Phi_m} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}_{\Phi_m}^\top \cdot [h_i || h_j]))}{\sum_{l \in N_i^{\Phi_m}} \exp(\text{LeakyReLU}(\mathbf{a}_{\Phi_m}^\top \cdot [h_i || h_l]))}$$

- 然后对不同类型的邻居节点信息进行聚合——type-level attention

$$w_{\Phi_m} = \frac{1}{|V|} \sum_{i \in V} \mathbf{a}_{sc}^\top \cdot \tanh(\mathbf{W}_{sc} h_i^{\Phi_m} + \mathbf{b}_{sc}),$$

$$\beta_{\Phi_m} = \frac{\exp(w_{\Phi_m})}{\sum_{i=1}^S \exp(w_{\Phi_i})},$$

$$z_i^{sc} = \sum_{m=1}^S \beta_{\Phi_m} \cdot h_i^{\Phi_m}.$$

- 最终得到网络模式视图下节点*i*的表征 $z_i^{sc}$

表征 $z_i^{sc}$ 的组成中不包含 $h_i$ 的信息，仅包含节点*i*的一阶邻居信息



# Method

## ➤ Part 3: Meta-path View Guided Encoder

- 假设从节点*i*出发有M种元路径{ $P_1, P_2, \dots, P_M$ }，则节点*i*通过类型为 $P_n$ 的元路径相连的邻居节点总数可表示为 $N_i^{P_n}$ 。

- 首先对通过同类型元路径相连的节点信息进行聚合

$$h_i^{P_n} = \frac{1}{d_i + 1} h_i + \sum_{j \in N_i^{P_n}} \frac{1}{\sqrt{(d_i + 1)(d_j + 1)}} h_j,$$

- 然后对不同类型元路径的信息进行聚合

$$w_{P_n} = \frac{1}{|V|} \sum_{i \in V} \mathbf{a}_{mp}^\top \cdot \tanh(\mathbf{W}_{mp} h_i^{P_n} + \mathbf{b}_{mp}),$$

$$\beta_{P_n} = \frac{\exp(w_{P_n})}{\sum_{i=1}^M \exp(w_{P_i})},$$

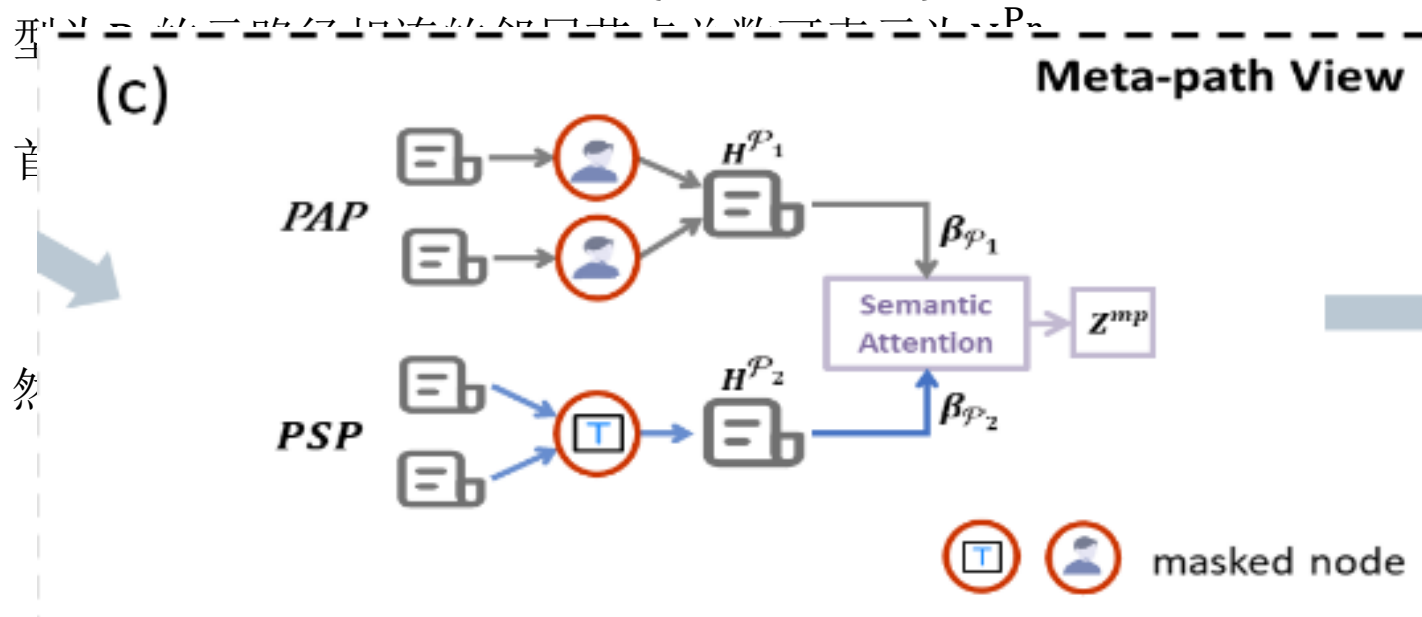
$$z_i^{mp} = \sum_{n=1}^M \beta_{P_n} \cdot h_i^{P_n},$$

- 最终得到元路径视图下节点*i*的表征 $z_i^{mp}$

# Method

## ➤ Part 3: Meta-path View Guided Encoder

- 假设从节点*i*出发有M种元路径 $\{P_1, P_2, \dots, P_M\}$ ，则节点*i*通过类



- 最终得到元路径视图下节点*i*的表征 $z_i^{mp}$



# Method

## ➤ Part 3: Meta-path View Guided Encoder

- 假设从节点*i*出发有M种元路径{ $P_1, P_2, \dots, P_M$ }，则节点*i*通过类型为 $P_n$ 的元路径相连的邻居节点总数可表示为 $N_i^{P_n}$ 。

- 首先对通过同类型元路径相连的节点信息进行聚合

$$h_i^{P_n} = \frac{1}{d_i + 1} h_i + \sum_{j \in N_i^{P_n}} \frac{1}{\sqrt{(d_i + 1)(d_j + 1)}} h_j,$$

- 然后对不同类型元路径的信息进行聚合

$$w_{P_n} = \frac{1}{|V|} \sum_{i \in V} \mathbf{a}_{mp}^\top \cdot \tanh(\mathbf{W}_{mp} h_i^{P_n} + \mathbf{b}_{mp}),$$

$$\beta_{P_n} = \frac{\exp(w_{P_n})}{\sum_{i=1}^M \exp(w_{P_i})},$$

$$z_i^{mp} = \sum_{n=1}^M \beta_{P_n} \cdot h_i^{P_n},$$

- 最终得到元路径视图下节点*i*的表征 $z_i^{mp}$

表征 $z_i^{mp}$ 的组成中包含 $h_i$ 的信息和节点*i*的高阶邻居信息，但不包含一阶邻居信息。

# Method

## ➤ Part 4: View Mask Mechanism

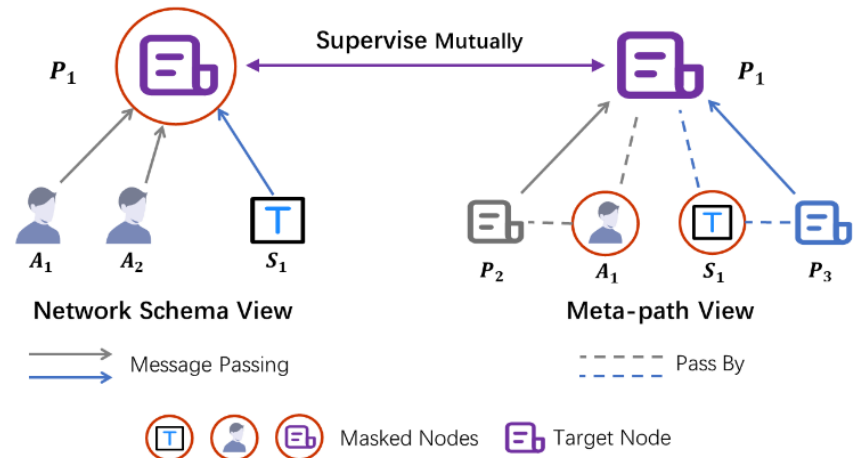
- 两个视图相当于在信息层面各有掩码，但是最终获得的信息又形成了互补且相关，可以用于监督另一个视图训练。

$$z_i^{sc} = \sum_{m=1}^S \beta_{\Phi_m} \cdot h_i^{\Phi_m}.$$

表征 $z_i^{sc}$ 的组成中不包含 $h_i$ 的信息，仅包含节点 $i$ 的一阶邻居信息

$$z_i^{mp} = \sum_{n=1}^M \beta_{\mathcal{P}_n} \cdot h_i^{\mathcal{P}_n},$$

表征 $z_i^{mp}$ 的组成中包含 $h_i$ 的信息和节点 $i$ 的高阶邻居信息，但不包含一阶邻居信息。





# Method

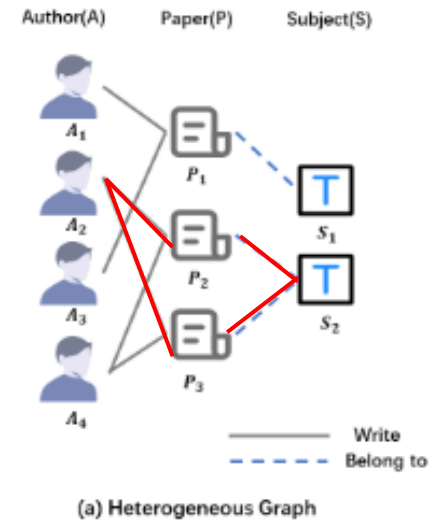
## ➤ Part 5: Collaboratively Contrastive Optimization

- 首先将 $z_i^{sc}$ 和 $z_i^{mp}$ 映射到同一个向量空间

$$z_i^{sc\_cross} = W^{(2)} \sigma \left( W^{(1)} z_i^{sc} + b^{(1)} \right) + b^{(2)},$$

$$z_i^{mp\_cross} = W^{(2)} \sigma \left( W^{(1)} z_i^{mp} + b^{(1)} \right) + b^{(2)},$$

- 作者在这里提出了两种区分正负样本的方法：
  1. 直接把另一个视图的表征当正样本（由掩码机制而产生的互补现象使得可以互相监督训练）
  2. 考虑到节点之间是高度相关的，做法一只考虑了当前节点 $i$ 的信息，而没有考虑其邻居的信息，作者于是提出如果两个节点 $i$ 和 $j$ 之间存在多条元路径那么节点 $j$ 是节点 $i$ 的正样本。
- 处理方法  
 用函数 $C_i(j)$ 来计算节点  $i$ 和 $j$ 之间存在的元路径条数，然后做排序，取最高的一部分节点 $j$ 为正样本，其余的为负样本。



$$C_i(j) = \sum_{n=1}^M \mathbb{1} \left( j \in N_i^{P_n} \right),$$



# Method

## ➤ Part 5: Collaboratively Contrastive Optimization

- 然后可以得到模型的优化目标函数：

$$\begin{aligned} & \mathcal{L}_i^{sc\_cross} \\ &= -\log \frac{\sum_{j \in \mathbb{P}_i} \exp\left(\text{sim}\left(z_i^{sc\_cross}, z_j^{mp\_cross}\right) / \tau\right)}{\sum_{k \in \{\mathbb{P}_i \cup \mathbb{N}_i\}} \exp\left(\text{sim}\left(z_i^{sc\_cross}, z_k^{mp\_cross}\right) / \tau\right)}, \end{aligned} \quad (11)$$

$$\begin{aligned} & \mathcal{L}_i^{mp\_cross} \\ &= -\log \frac{\sum_{j \in \mathbb{P}_i} \exp\left(\text{sim}\left(z_i^{mp\_cross}, z_j^{sc\_cross}\right) / \tau\right)}{\sum_{k \in \{\mathbb{P}_i \cup \mathbb{N}_i\}} \exp\left(\text{sim}\left(z_i^{mp\_cross}, z_k^{sc\_cross}\right) / \tau\right)}, \end{aligned} \quad (12)$$

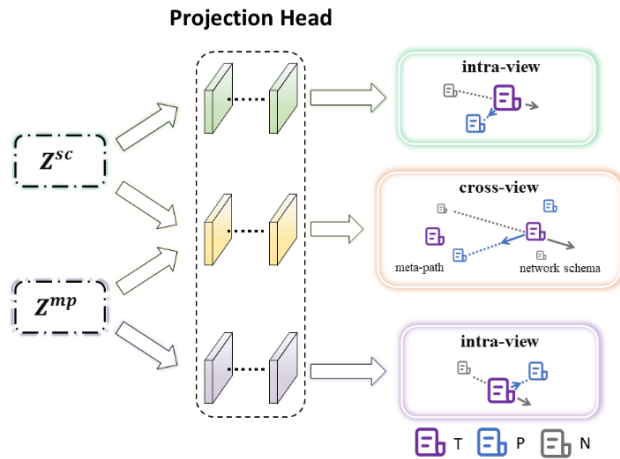
$$\mathcal{L} = \frac{1}{|V|} \sum_{i \in V} [\lambda \cdot \mathcal{L}_i^{sc\_cross} + (1 - \lambda) \cdot \mathcal{L}_i^{mp\_cross}], \quad (13)$$

- 节点  $i$  的损失函数由两部分组成：
- $\mathcal{L}_i^{sc\_cross}$ ：节点  $i$  在网络模式视图下的表征为  $\text{target}$ ，与  $i$  的正样本节点  $j$  在元路径视图下的表征算交叉熵。
- $\mathcal{L}_i^{mp\_cross}$ ：与  $\mathcal{L}_i^{sc\_cross}$  的计算方式刚好反过来。
- 二者加权得到最终  $\text{loss}$ ，实现了跨视图对比学习。

# Method

## ➤ Part 6: Model Extension——Heco++

- 通过跨视图对比学习，Heco学到的是关于节点的不变性信息。然而在某些特殊情况下，还需要结合view-specific的信息才能达到更好的效果。



$$z_i^{sc\_intra} = W_{sc}^{(2)} \sigma \left( W_{sc}^{(1)} z_i^{sc} + b_{sc}^{(1)} \right) + b_{sc}^{(2)},$$

$$z_i^{mp\_intra} = W_{mp}^{(2)} \sigma \left( W_{mp}^{(1)} z_i^{mp} + b_{mp}^{(1)} \right) + b_{mp}^{(2)},$$

$$\mathcal{L}_i^{sc\_intra} = -\log \frac{\sum_{j \in \mathbb{P}_i} \exp \left( \text{sim} \left( z_i^{sc\_intra}, z_j^{sc\_intra} \right) / \tau_{sc} \right)}{\sum_{k \in \{\mathbb{P}_i \cup \mathbb{N}_i\}} \exp \left( \text{sim} \left( z_i^{sc\_intra}, z_k^{sc\_intra} \right) / \tau_{sc} \right)}, \quad (16)$$

$$\mathcal{L}_i^{mp\_intra} = -\log \frac{\sum_{j \in \mathbb{P}_i} \exp \left( \text{sim} \left( z_i^{mp\_intra}, z_j^{mp\_intra} \right) / \tau_{mp} \right)}{\sum_{k \in \{\mathbb{P}_i \cup \mathbb{N}_i\}} \exp \left( \text{sim} \left( z_i^{mp\_intra}, z_k^{mp\_intra} \right) / \tau_{mp} \right)}, \quad (17)$$

$$\mathcal{J} = \mathcal{L} + \frac{1}{|V|} \sum_{i \in V} [\lambda_1 \cdot \mathcal{L}_i^{sc\_intra} + \lambda_2 \cdot \mathcal{L}_i^{mp\_intra}], \quad (18)$$





# Experiment

## ➤ 基本设置:

- 数据集：四个开源异质数据集。ACM , DBLP , Freebase, AMiner
- Baseline：无监督同质图神经网络3个（GraphSAGE, GAE, DGI）、无监督异质图神经网络4个（Mp2vec , HERec , HetGNN , DMGI）、半监督异质图神经网络1个（HAN）。
- 下游任务：节点分类、节点聚类。



# Experiment

## ➤ 主要结果:

TABLE 1  
 Quantitative results on node classification. (bold: best; underline: runner-up)

Datasets	Metric	Split	GraphSAGE	GAE	Mp2vec	HERec	HetGNN	HAN	DGI	DMGI	HeCo	HeCo++
ACM	Ma-F1	20	47.13±4.7	62.72±3.1	51.91±0.9	55.13±1.5	72.11±0.9	85.66±2.1	79.27±3.8	87.86±0.2	88.56±0.8	89.33±0.5
		40	55.96±6.8	61.61±3.2	62.41±0.6	61.21±0.8	72.02±0.4	87.47±1.1	80.23±3.3	86.23±0.8	87.61±0.5	88.70±0.7
		60	56.59±5.7	61.67±2.9	61.13±0.4	64.35±0.8	74.33±0.6	88.41±1.1	80.03±3.3	87.97±0.4	89.04±0.5	89.51±0.7
	Mi-F1	20	49.72±5.5	68.02±1.9	53.13±0.9	57.47±1.5	71.89±1.1	85.11±2.2	79.63±3.5	87.60±0.8	88.13±0.8	88.96±0.5
		40	60.98±3.5	66.38±1.9	64.43±0.6	62.62±0.9	74.46±0.8	87.21±1.2	80.41±3.0	86.02±0.9	87.45±0.5	88.40±0.8
		60	60.72±4.3	65.71±2.2	62.72±0.3	65.15±0.9	76.08±0.7	88.10±1.2	80.15±3.2	87.82±0.5	88.71±0.5	89.30±0.7
	AUC	20	65.88±3.7	79.50±2.4	71.66±0.7	75.44±1.3	84.36±1.0	93.47±1.5	91.47±2.3	96.72±0.3	96.49±0.3	97.25±0.2
		40	71.06±5.2	79.14±2.5	80.48±0.4	79.84±0.5	85.01±0.6	94.84±0.9	91.52±2.3	96.35±0.3	96.40±0.4	97.08±0.2
		60	70.45±6.2	77.90±2.8	79.33±0.4	81.64±0.7	87.64±0.7	94.68±1.4	91.41±1.9	96.79±0.2	96.55±0.3	97.50±0.2
DBLP	Ma-F1	20	71.97±8.4	90.90±0.1	88.98±0.2	89.57±0.4	89.51±1.1	89.31±0.9	87.93±2.4	89.94±0.4	91.28±0.2	91.40±0.2
		40	73.69±8.4	89.60±0.3	88.68±0.2	89.73±0.4	88.61±0.8	88.87±1.0	88.62±0.6	89.25±0.4	90.34±0.3	90.56±0.2
		60	73.86±8.1	90.08±0.2	90.25±0.1	90.18±0.3	89.56±0.5	89.20±0.8	89.19±0.9	89.46±0.6	90.64±0.3	91.01±0.3
	Mi-F1	20	71.44±8.7	91.55±0.1	89.67±0.1	90.24±0.4	90.11±1.0	90.16±0.9	88.72±2.6	90.78±0.3	91.97±0.2	92.03±0.1
		40	73.61±8.6	90.00±0.3	89.14±0.2	90.15±0.4	89.03±0.7	89.47±0.9	89.22±0.5	89.92±0.4	90.76±0.3	90.87±0.2
		60	74.05±8.3	90.95±0.2	91.17±0.1	91.01±0.3	90.43±0.6	90.34±0.8	90.35±0.8	90.66±0.5	91.59±0.2	91.86±0.2
	AUC	20	90.59±4.3	98.15±0.1	97.69±0.0	98.21±0.2	97.96±0.4	98.07±0.6	96.99±1.4	97.75±0.3	98.32±0.1	98.39±0.1
		40	91.42±4.0	97.85±0.1	97.08±0.0	97.93±0.1	97.70±0.3	97.48±0.6	97.12±0.4	97.23±0.2	98.06±0.1	98.17±0.1
		60	91.73±3.8	98.37±0.1	98.00±0.0	98.49±0.1	97.97±0.2	97.96±0.5	97.76±0.5	97.72±0.4	98.59±0.1	98.62±0.1
Freebase	Ma-F1	20	45.14±4.5	53.81±0.6	53.96±0.7	55.78±0.5	52.72±1.0	53.16±2.8	54.90±0.7	55.79±0.9	59.23±0.7	59.87±1.0
		40	44.88±4.1	52.44±2.3	57.80±1.1	59.28±0.6	48.57±0.5	59.63±2.3	53.40±1.4	49.88±1.9	61.19±0.6	61.33±0.5
		60	45.16±3.1	50.65±0.4	55.94±0.7	56.50±0.4	52.37±0.8	56.77±1.7	53.81±1.1	52.10±0.7	60.13±1.3	60.86±1.0
	Mi-F1	20	54.83±3.0	55.20±0.7	56.23±0.8	57.92±0.5	56.85±0.9	57.24±3.2	58.16±0.9	58.26±0.9	61.72±0.6	62.29±1.9
		40	57.08±3.2	56.05±2.0	61.01±1.3	62.71±0.7	53.96±1.1	63.74±2.7	57.82±0.8	54.28±1.6	64.03±0.7	64.27±0.5
		60	55.92±3.2	53.85±0.4	58.74±0.8	58.57±0.5	56.84±0.7	61.06±2.0	57.96±0.7	56.69±1.2	63.61±1.6	64.15±0.9
	AUC	20	67.63±5.0	73.03±0.7	71.78±0.7	73.89±0.4	70.84±0.7	73.26±2.1	72.80±0.6	73.19±1.2	76.22±0.8	76.68±0.7
		40	66.42±4.7	74.05±0.9	75.51±0.8	76.08±0.4	69.48±0.2	77.74±1.2	72.97±1.1	70.77±1.6	78.44±0.5	79.51±0.3
		60	66.78±3.5	71.75±0.4	74.78±0.4	74.89±0.4	71.01±0.5	75.69±1.5	73.32±0.9	73.17±1.4	78.04±0.4	78.27±0.7
AMiner	Ma-F1	20	42.46±2.5	60.22±2.0	54.78±0.5	58.32±1.1	50.06±0.9	56.07±3.2	51.61±3.2	59.50±2.1	71.38±1.1	72.28±1.4
		40	45.77±1.5	65.66±1.5	64.77±0.5	64.50±0.7	58.97±0.9	63.85±1.5	54.72±2.6	61.92±2.1	73.75±0.5	75.35±0.5
		60	44.91±2.0	63.74±1.6	60.65±0.3	65.53±0.7	57.34±1.4	62.02±1.2	55.45±2.4	61.15±2.5	75.80±1.8	76.28±0.6
	Mi-F1	20	49.68±3.1	65.78±2.9	60.82±0.4	63.64±1.1	61.49±2.5	68.86±4.6	62.39±3.9	63.93±3.3	78.81±1.3	80.00±1.0
		40	52.10±2.2	71.34±1.8	69.66±0.6	71.57±0.7	68.47±2.2	76.89±1.6	63.87±2.9	63.60±2.5	80.53±0.7	82.01±0.6
		60	51.36±2.2	67.70±1.9	63.92±0.5	69.76±0.8	65.61±2.2	74.73±1.4	63.10±3.0	62.51±2.6	82.46±1.4	82.80±0.7
	AUC	20	70.86±2.5	85.39±1.0	81.22±0.3	83.35±0.5	77.96±1.4	78.92±2.3	75.89±2.2	85.34±0.9	90.82±0.6	91.59±0.6
		40	74.44±1.3	88.29±1.0	88.82±0.2	88.70±0.4	83.14±1.6	80.72±2.1	77.86±2.1	88.02±1.3	92.11±0.6	93.46±0.2
		60	74.16±1.3	86.92±0.8	85.57±0.2	87.74±0.5	84.77±0.9	80.39±1.5	77.21±1.4	86.20±1.7	92.40±0.7	93.68±0.3



# Experiment

➤ 主要结果:

TABLE 2  
 Quantitative results on node clustering. (bold: best; underline: runner-up)

Datasets	ACM		DBLP		Freebase		AMiner	
Metrics	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
GraphSage	29.20	27.72	51.50	36.40	9.05	10.49	15.74	10.10
GAE	27.42	24.49	72.59	77.31	19.03	14.10	28.58	20.90
Mp2vec	48.43	34.65	73.55	77.70	16.47	17.32	30.80	25.26
HERec	47.54	35.67	70.21	73.99	19.76	19.36	27.82	20.16
HetGNN	41.53	34.81	69.79	75.34	12.25	15.01	21.46	26.60
DGI	51.73	41.16	59.23	61.85	18.34	11.29	22.06	15.93
DMGI	51.66	46.64	70.06	75.46	16.98	16.91	19.24	20.09
HeCo	<u>56.87</u>	<u>56.94</u>	<u>74.51</u>	<u>80.17</u>	<u>20.38</u>	<u>20.98</u>	<u>32.26</u>	<u>28.64</u>
HeCo++	<b>60.82</b>	<b>60.09</b>	<b>75.39</b>	<b>81.20</b>	<b>20.62</b>	<b>21.88</b>	<b>38.07</b>	<b>36.44</b>



# Experiment

➤ 主要结果:

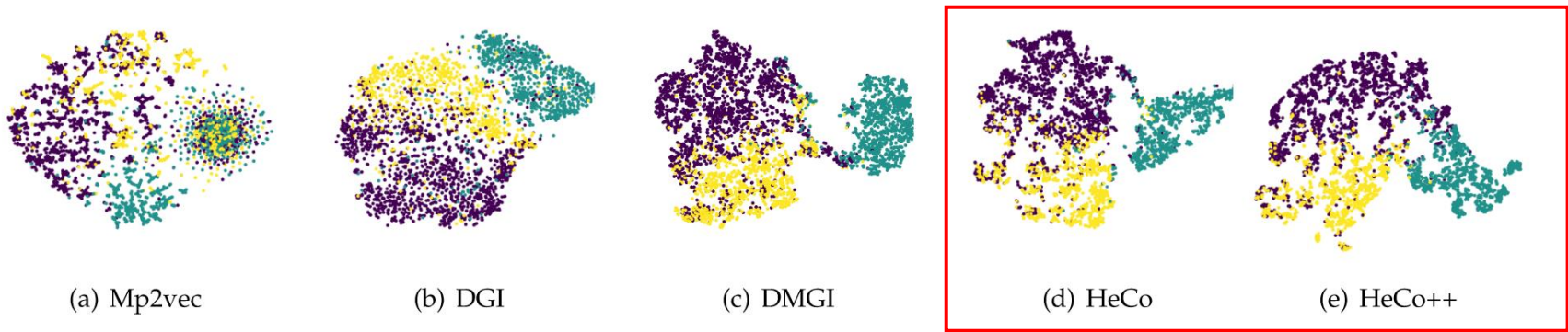


Fig. 5. Visualization of the learned node embedding on ACM. The Silhouette scores for (a) (b) (c) (d) (e) are 0.0292, 0.1862, 0.3015, 0.3642 and **0.3885**, respectively.

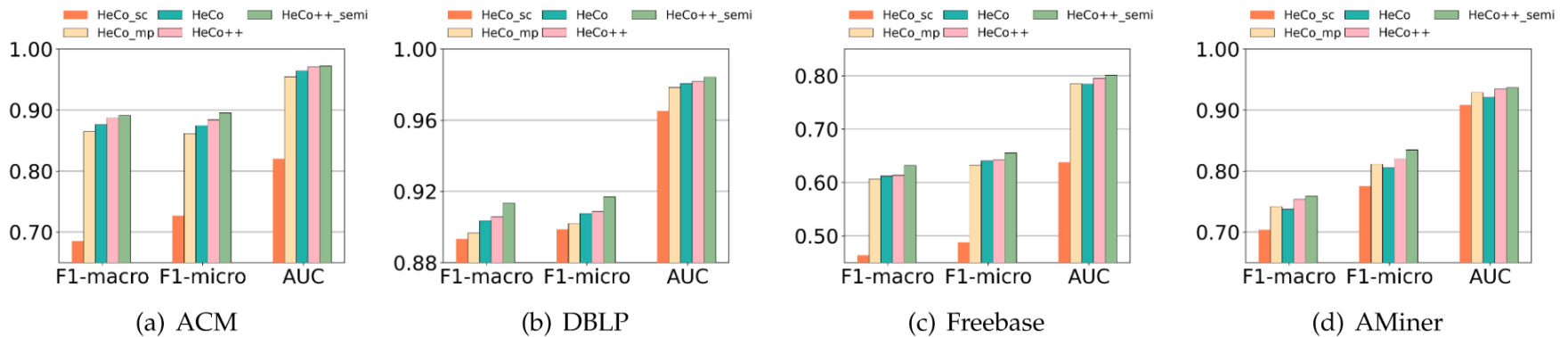


Fig. 6. The comparison of HeCo and its variants.

# 感谢观看



讲者：王培宇

北京航空航天大学  
BIGSCITY交流研讨会  
2023年12月25日