# 图神经网络导论

## 图结构学习

授课教师：周晟

浙江大学　软件学院

2022.12

# 课程内容

1. 研究背景
2. 基于优化的图结构学习
   - LDS
   - Pro-GNN
3. 基于图结构学习的图神经网络
   - AGCN
   - GRCN
   - PTDNet
   - Geom-GCN
   - IDGL
4. 基于生成模型的图结构学习
5. 无监督的图结构学习

# 上节课回顾

计算机网络

社交网络

交通网络

神经元网络

卫星网络

# 上节课回顾

## 困难与挑战

1. 传统图神经网络难以充分建模富信息网络特点
2. 不同类型富信息网络需要设计不同的图神经网络结构

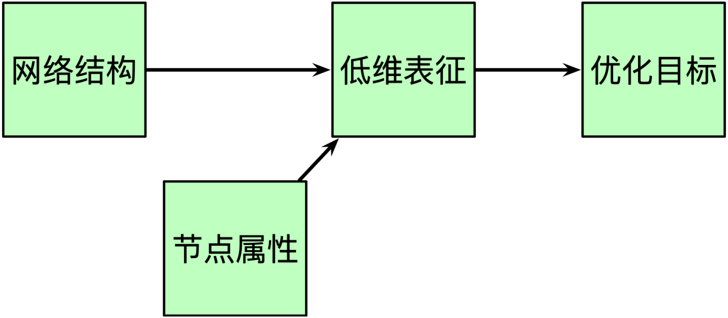
## 常见的富信息图神经网络

1. 有向图
2. 异构图
3. 动态图

# 研究背景

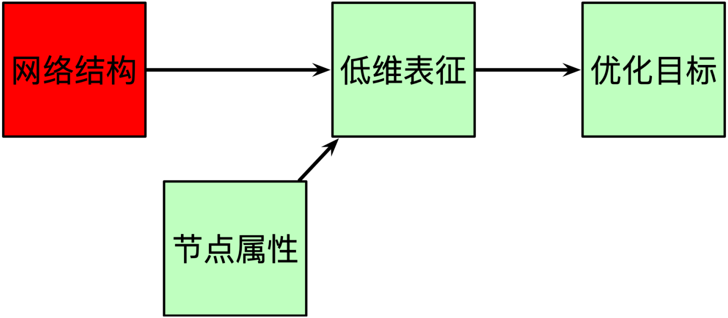
## 图神经网络的原理

通过将节点的信息沿着网络结构传递（Message Passing），捕获节点间的关系和节点的特征，学习更好的图（节点，边，图）表征。



图神经网络基本框架

# 研究背景


网络结构

低维表征

优化目标

节点属性

网络结构受损下的图神经网络

## 问题与挑战

实际场景中，网络结构往往面临受损或缺失的问题，使得依赖网络结构信息的图神经网络效果严重下降。

# 图结构缺失问题

1. 数据天然的稀疏性（社交网络关注量有限）
2. 数据采集、存储过程中丢失
3. 人为隐藏关系（犯罪）
4. 数据集本身没有关系型结构

:(

你的电脑遇到问题，需要重新启动。

我们只收集某些错误信息，然后为你重新启动。(完成100%)

如果你想了解更多信息，则可以稍后在线搜索此错误 ：INVALID_AFFINITY_SET

# 图结构冗余问题

1. 数据采集、存储过程中错误
2. 电商网络中，刷单行为（诈骗高发）
3. 人为注入，“浑水摸鱼”

# 图结构学习的意义

图结构学习对于图神经网络的理论和应用均有重要意义：
图结构方面：

1. 理解数据间的真实关系（因果）
2. 预测图结构演化趋势
3. 发掘数据中异常关系

图神经网络方面：

1. 更好的消息传递路径
2. 迭代升级的图神经网络框架
3. 鲁棒图神经网络

# 图数据的思考

我们希望生成什么样的图?（和真实世界的图尽量接近/相似）

1. 二八定律（Pareto principle，也被称为 80/20 法则、关键少数法则、八二法则、帕累托法则），是指约仅有 $20\%$ 的变因操纵着 $80\%$ 的局面。
2. 网络中节点的度的分布，并不是均匀的。
3. …

# 同质性（homophily）

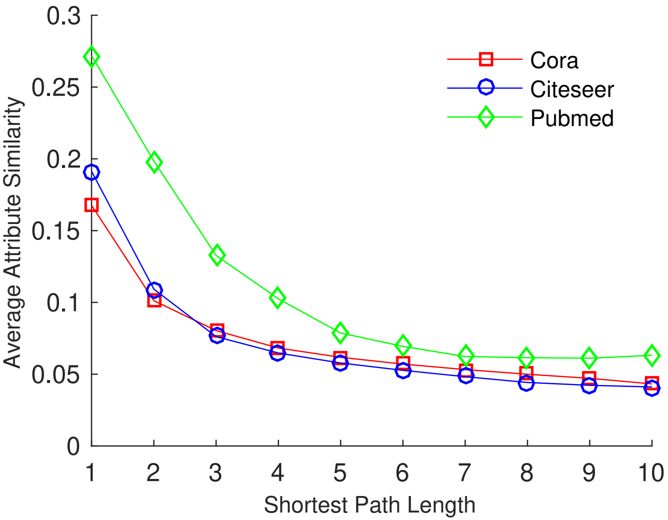
**同质性（homophily）**

同质性（homophily），指一个图网络中相邻节点相似的程度。在节点分类任务中，同质性通常被定义为相邻节点属于同一类别的概率。

可以证明，在同质性为 1 的极端情况下，GNN 可以达到100%的准确率。因此，如何产生更具有同质性的图结构是图结构学习中值得思考的问题。

# 特征平滑性（Smoothness）

真实数据集上节点属性和拓扑结构相关性分析



# 图统计量

除了节点级的局部特征，大规模真实图数据往往具备特殊的宏观性质。

常用如下四种统计量来描述一个图的全局性质：

1. 度分布，Degree distribution： P(k)
2. 聚类系数，Clustering coefficient： C
3. 连通区域，Connected components： s
4. 路径长度，Path length： h

# 图统计量

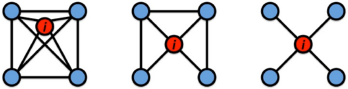
- 度分布，Degree distribution $P(k)$ 是指：度为 $K$ 的节点数量的分布，$N_k =$ 度为 $k$ 的节点数量，

$$P(k) = N_k/N$$

- 聚类系数，Clustering coefficient

$$C_i = \frac{2e_i}{k_i\left(k_i - 1\right)}$$

  - $e_i$ 是 i 节点的邻居中互相连边的数量
  - $k_i$ 是 i 节点的度



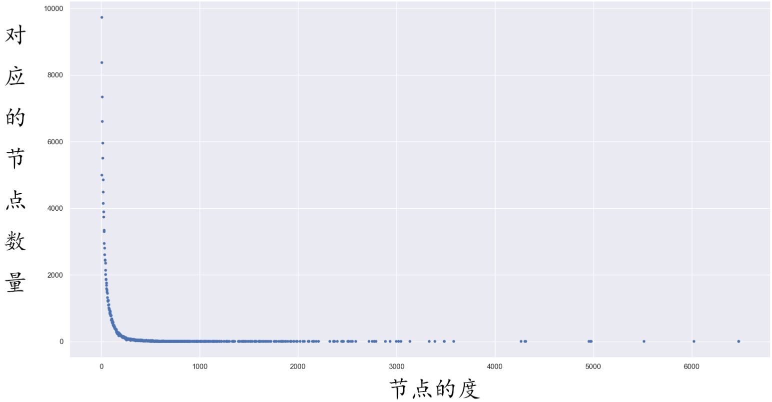
$C_i = 1$　　$C_i = 1/2$　　$C_i = 0$

$C_i \in [0,1]$

# 图统计量

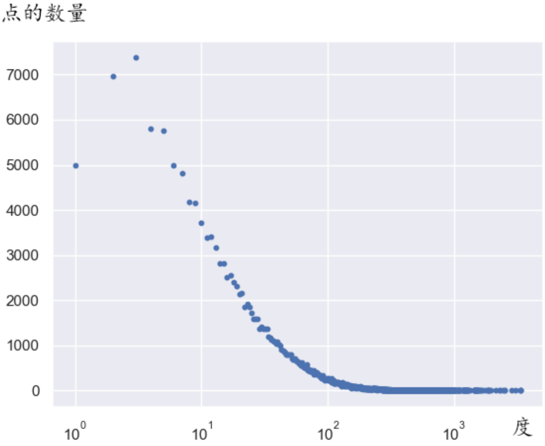
- 连通区域，Connected components，指图中联通区域按大小排序，其不同大小连通区域数量的分布
- 路径长度，Path length，指图中所有存在的最短路径的长度分布
- 以 Twitter 的社交网络图数据集为例，该数据集的邻接矩阵大小为 $157575 \times 157575$

# 图统计量-以 Twitter 数据集为例



度分布

# 图统计量-以 Twitter 数据集为例



横轴指数化的度分布

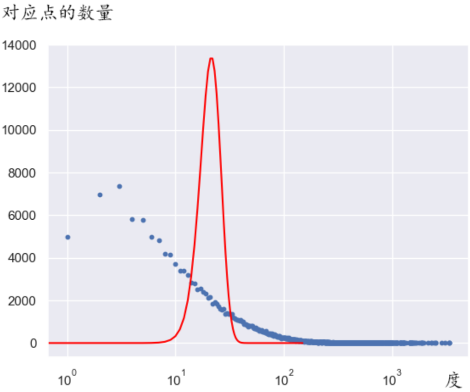
# 随机图的度分布

$$p(k) = \begin{pmatrix} N-1 \\ K \end{pmatrix} P^k (1-P)^{N-1-k}$$

- 一个点的度为 $k$ 的概率（有 $k$ 个点与之相连），等价于除它本身之外的 $N-1$ 个点选 $k$ 个和它相连，剩下 $N-1-k$ 和它不连的概率。$P$ 是两点之间存在一条边的概率，$P=\frac{E}{N*N}$。典型二项分布。
- 好的图结构学习结果应保留图的这种天然特性

# 随机图的度分布

- 以 Twitter 数据集为例，其中共有 157575 点 3530653 边，把真实分布和随机分布 (红色) 画在一起



# 图统计量-以 Twitter 数据集为例

随机 10000 个节点的 Clustering Coefficient 分布

均值为 0.34

# 随机图的聚类散度

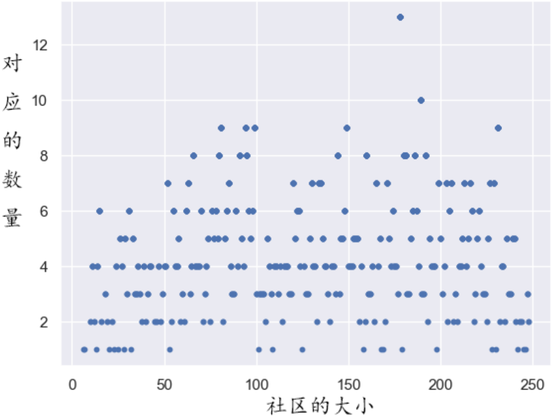
- 对于一个随机图中的节点来说，其平均聚类散度的期望值为：

$$E\left[C_i\right] = \frac{2 * E\left[e_i\right]}{k_i\left(k_i - 1\right)} = \frac{2 * p\frac{k_i(k_i-1)}{2}}{k_i\left(k_i - 1\right)} = p = \frac{\bar{k}}{n-1}$$

  其中 n 为节点数，$\bar{k}$ 是平均度数
- 对 Twitter 数据集而言，同等数量的节点和边构成的随机图，聚类散度的期望是 $\frac{3530653}{157575*157575} = 0.0001421$

# 图统计量-以 Twitter 数据集为例

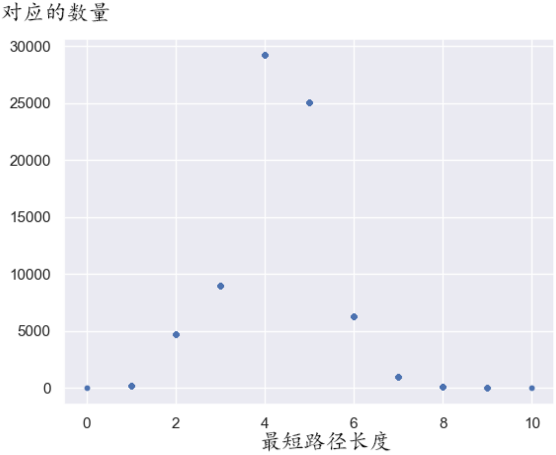
- 连通区域分布：Twitter 数据集是一个全连通图！



运行社区发现算法，得到的社区大小分布图

# 图统计量-以 Twitter 数据集为例

- 计算资源有限，随机 1000 个点的单源最短路径统计分布

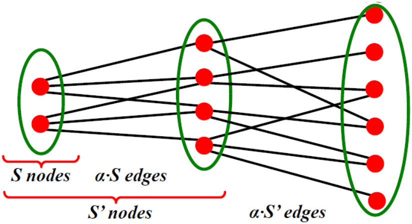


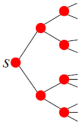
最短路径长度分布图

# 随机图的最短路径分布

- 如果图是连通的，那么 BFS 遍历树的第二层应该是初始点的邻接点，然后依次展开直到覆盖图中所有点。假设遍历的是随机图，那么这棵树的深度的期望就应该是

$$\log_{\text{平均的度}} \text{节点总数} = \log_{np} n = \log n / \log np$$





- 对于 Twitter 图来说，平均最短路径的期望为 $O((\log_{3530653/157575} 157575)) \approx 3.8489422080670335$

# 基于优化的图结构学习

## 研究动机

图结构可以看作节点之间是否有边的参数的集合，可以直接对图结构进行参数化学习。

## 困难与挑战

1. 优化目标如何确定?
2. 参数量大（$O(N^2)$）
3. 离散优化，难以训练

# Learning Discrete Structures for Graph Neural Networks (LDS)[1]

## 研究动机

真实世界的图结构往往是噪声的或者无法观测的，而不好的图结构也将限制基于图数据训练的图神经网络的效果，因此希望将图结构与图神经网络在统一的框架下进行学习。

## 亮点

1. 第一个对结构进行学习的工作，打开了后续 GSL 的大门。
2. 双层优化（Bilevel Progamming）影响了很多后续工作

---

[1] Learning Discrete Structures for Graph Neural Networks(ICML 2020)

# LDS: Bilevel Programming

$$\min_{A, w_A} F\left(w_A, A\right) \text{ such that } w_A \in \arg\min_{w} L(w, A).$$

$$L(w, A) = \sum_{v \in V_{train}} l(f_w(X, A)_v, y_v) + \Omega(w)$$

$$F(w_A, A) = \sum_{v \in V_{val}} l(f_{w_A}(X, A)_v, y_v)$$

**特点**：两个变量 $A, w_A$ 之间并不独立，$w_A$ 是 $A$ 确定情况下优化问题的最小解。

**难点**：内层优化 $w_A$ 通常不能得到解析解

# LDS： Reparameterization

**优化难点**

邻接矩阵中的元素是离散变量，无法直接求导

LDS 假设 A 从多个独立的伯努利分布中生成：
$A \sim \mathrm{Ber}(\theta), \theta \in (0,1)^{n\times n}$

优化问题变为：

$$\min_{\theta} \mathbb{E}_{A\sim \mathrm{Ber}(\theta)}[F(w_\theta, A)]$$

$$\text{such that } w_\theta = \arg\min_{w} \mathbb{E}_{A\sim \mathrm{Ber}(\theta)}[L(w, A)]$$

# LDS：内层优化

$$w_\theta = \arg\min_w \mathbb{E}_{A\sim\mathrm{Ber}(\theta)}[L(w,A)]$$

$$\mathbb{E}_{A\sim\mathrm{Ber}(\theta)}[L(w,A)] = \sum_{A\in\mathcal{H}_N} P_\theta(A)L(w,A)$$

由于采样过程的存在，上式不可能直接求出。

LDS 每次采样一个结构梯度下降：

$$w_{\theta,t+1} = w_{\theta,t} - \gamma\nabla L(w_{\theta,t}, A_t), A_t \sim \mathrm{Ber}(\theta)$$

# LDS：外层优化

$$\min_{\theta} \mathbb{E}_{A\sim \mathrm{Ber}(\theta)}[F(w_\theta, A)]$$

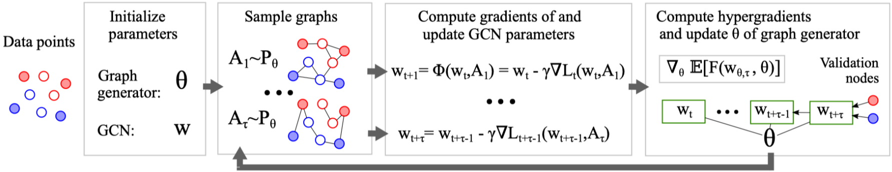
求解对 $\theta$ 的梯度，将对期望的梯度近似为梯度的期望：

$$\begin{aligned}\nabla_\theta \mathbb{E}_{A\sim \mathrm{Ber}(\theta)}[F(w_{\theta,T}, A)] &\approx \mathbb{E}_{A\sim \mathrm{Ber}(\theta)}[\nabla_A F(w_{\theta,T}, A)] \\ &= \mathbb{E}_A[\partial_w F(w_{\theta,T}, A)\nabla_A w_{\theta,T} + \partial_A F(w_{\theta,T}, A)]\end{aligned}$$

同样使用 SGD 迭代近似最优解：

$$\theta' = \theta - \eta(\partial_w F(w_{\theta,T}, A)\nabla_A w_{\theta,T} + \partial_A F(w_{\theta,T}, A)), A \sim \mathrm{Ber}(\theta)$$

# LDS：交替优化



进行 $T$ 次内层迭代：

$$w_{\theta,t+1} = w_{\theta,t} - \gamma \nabla L(w_{\theta,t}, A_t), A_t \sim \mathrm{Ber}(\theta)$$

进行一次外层迭代：

$$\theta' = \theta - \eta(\partial_w F(w_{\theta,T}, A) \nabla_A w_{\theta,T} + \partial_A F(w_{\theta,T}, A)), A \sim \mathrm{Ber}(\theta)$$

# LDS 总结

## 优点

1. 系统性提出 Graph Structure Learning 理念
2. 双层优化方法
3. 使用迭代方式优化图结构和 GCN 参数

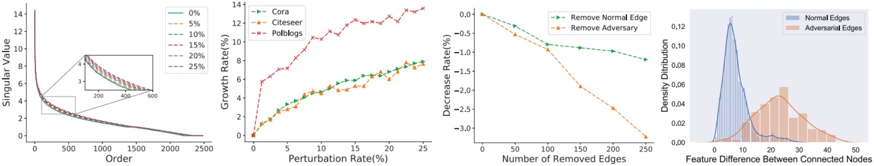
## 缺点

1. 参数量大，优化速度较慢
2. 边之间相互独立

# Graph Structure Learning for Robust Graph Neural Networks(Pro-GNN)

真实图结构的性质（properties）[2]：

1. 低秩
2. 稀疏
3. 邻接的节点特征相近



(a) Singular Values (b) Rank Growth (c) Rank Decrease Rate (d) Feature Smoothness

**Figure 1: An illustrative example on the property changes of the adjacency matrix by adversarial attacks**

---

[2] Graph Structure Learning for Robust Graph Neural Networks(KDD 2020)

# Pro-GNN: 性质

1. 如何利用这些性质引导图结构的学习?
ProGNN 采取了直接优化邻接矩阵 $S$的方法，将 $S$ 视作一个 $n*n$ 的参数矩阵，这样这些性质可以表达为有关 $S$ 的一系列损失函数。

## 图性质的损失函数

1. 低秩和稀疏：

$$\mathcal{L}_0 = \|A - S\|_F^2 + \alpha\|S\|_1 + \beta\|S\|_*, s.t., S = S^T$$

2. 特征平滑：

$$\mathcal{L}_s = tr(X^T\hat{L}X) = \frac{1}{2}\sum_{i,j=1}^{N} S_{ij}(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}})^2, s.t., S = S^T$$

# Pro-GNN: 优化

2. 如何联合学习图结构和图神经网络?
最终损失：

$$
\begin{aligned}
\arg\min_{S\in\mathcal{S},\theta} \mathcal{L} &= \mathcal{L}_0 + \lambda\mathcal{L}_f + \gamma\mathcal{L}_{GNN} \\
&= \|A - S\|_F^2 + \alpha\|S\|_1 + \beta\|S\|_* + \gamma\mathcal{L}_{GNN}(\theta, S, X, \mathcal{Y}_L) \\
&\quad + \lambda tr(X^T\hat{L}X), s.t. S = S^T
\end{aligned}
$$

由于一起求解较为困难，使用交替优化（Alternative Optimization）迭代求解。

### 交替优化

交替优化经常在 GSL 方法中出现，用来解决双层优化等较困难的优化问题。其做法是固定一个，优化另一个，交替进行。

# Pro-GNN: 优化

$$\arg\min_{S\in\mathcal{S},\theta}\mathcal{L}=\|A-S\|_F^2+\alpha\|S\|_1+\beta\|S\|_*+\gamma\mathcal{L}_{GNN}(\theta,S,X,\mathcal{Y}_L)$$
$$+\lambda tr(X^T\hat{L}X),s.t.S=S^T$$

优化 GNN:

$$\theta\leftarrow\eta'\frac{\partial\mathcal{L}_{GNN}(\theta,S,X,\mathcal{Y}_L)}{\partial\theta}$$

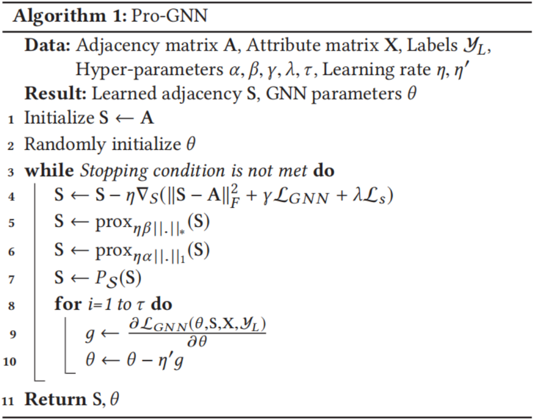
优化图结构 S：核范数和 $l_1$ 范数不可导，需使用 Proximal Optimization 数学工具[3]

---

[3] [Jin et al., 2020]

# Pro-GNN: 优化

整体算法:



**Algorithm 1:** Pro-GNN

```
Data: Adjacency matrix $\mathbf{A}$, Attribute matrix $\mathbf{X}$, Labels $\mathcal{Y}_L$,
      Hyper-parameters $\alpha, \beta, \gamma, \lambda, \tau$, Learning rate $\eta, \eta'$
Result: Learned adjacency $\mathbf{S}$, GNN parameters $\theta$
Initialize $\mathbf{S} \leftarrow \mathbf{A}$
Randomly initialize $\theta$
while Stopping condition is not met do
    $\mathbf{S} \leftarrow \mathbf{S} - \eta \nabla_{\mathbf{S}}(\|\mathbf{S} - \mathbf{A}\|_F^2 + \gamma \mathcal{L}_{GNN} + \lambda \mathcal{L}_s)$
    $\mathbf{S} \leftarrow \mathrm{prox}_{\eta\beta||.||_*}(\mathbf{S})$
    $\mathbf{S} \leftarrow \mathrm{prox}_{\eta\alpha||.||_1}(\mathbf{S})$
    $\mathbf{S} \leftarrow P_{\mathcal{S}}(\mathbf{S})$
    for i=1 to $\tau$ do
        $g \leftarrow \frac{\partial \mathcal{L}_{GNN}(\theta, \mathbf{S}, \mathbf{X}, \mathcal{Y}_L)}{\partial \theta}$
        $\theta \leftarrow \theta - \eta' g$
Return $\mathbf{S}, \theta$
```

每优化 $\tau$ 次 GNN 优化一次图结构 $S$

# 基于优化的图结构学习

## 优点

1. 简单直接
2. 可解释性强

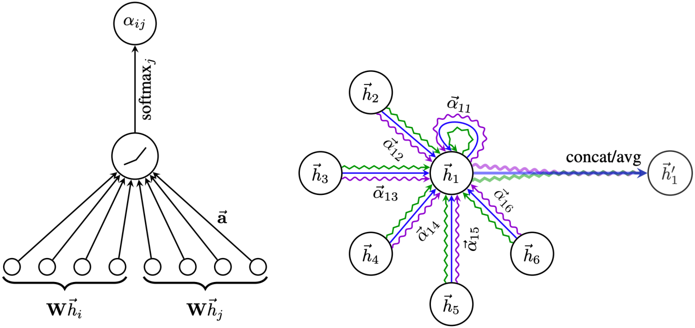
## 缺点

1. 复杂度高
2. 优化困难
3. 无法帮助 GNN

1. 研究背景
2. 基于优化的图结构学习
   - LDS
   - Pro-GNN
3. 基于图结构学习的图神经网络
   - AGCN
   - GRCN
   - PTDNet
   - Geom-GCN
   - IDGL
4. 基于生成模型的图结构学习
5. 无监督的图结构学习

# GAT 与图结构学习

$$\alpha_{ij} = \frac{\exp(\mathrm{LeakReLU}(\vec{a}^T[W\vec{h}_i||W\vec{h}_j]))}{\sum_{k\in\mathcal{N}_i}\exp(\mathrm{LeakyReLU}(\vec{a}^T[W\vec{h}_i||W\vec{h}_k]))}$$

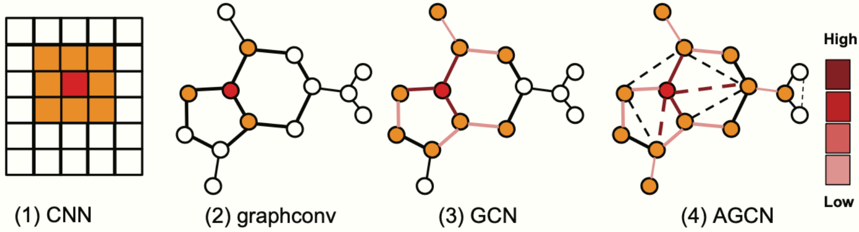


GAT 可看作一种特殊的图结构学习方法。

# Adaptive Graph Convolutional Neural Networks (AGCN)

## 研究动机

现有的图神经网络采用消息传递机制，但仅能在已观测的边上传递信息。由于网络稀疏性问题，实际可传递有效信息的节点并没有被连接。



AGCN 与 CNN、图卷积和 GCN 的对比[4]

# Adaptive Graph Convolutional Neural Networks

GCN 模型更新方式：

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

一个新的图结构即对应一个新的拉普拉斯矩阵。

## AGCN 对图结构学习的设想

- 图结构可参数化学习
- 参数量少（拉普拉斯矩阵参数化需要 $O(N^2)$ 的空间）
- 可被应用于不同的图结构（可优化更多任务，如点云分类）

解决方案：基于核（kernel）的方法

# Adaptive Graph Convolutional Neural Networks

## AGCN 的图结构学习策略

使用广义 Mahalanobis 距离实现基于核的距离度量

$$\mathbb{D}(x_i, x_j) = \sqrt{(x_i - x_j)^T M (x_i - x_j)}$$

- $x_i$ 是图卷积网络某一层中节点 $i$ 的特征。
- $M = W_d W_d^T$，$W_d \in \mathbb{R}^{d \times d}$ 是可学习的 kernel。
- $M = I$ 时等价于欧式距离

将参数量从 $O(N^2)$ 降至 $O(d^2)$!

# Adaptive Graph Convolutional Neural Networks

给定距离度量，使用 Gaussian Kernel 归一化得到稠密的邻接矩阵 $\tilde{A}$。

$$\tilde{A}_{ij} \leftrightarrow \mathbb{G}_{x_i,x_j} = \exp(-\mathbb{D}(x_i, x_j)/(2\sigma^2))$$

将其作为残差结构与原图结构叠加，得到更新后的拉普拉斯矩阵并用于卷积图神经网络：

$$\hat{L} = L + \alpha L_{res} = L + \alpha(I - \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2})$$

为什么要加上原来的图结构?

# Adaptive Graph Convolutional Neural Networks

总结与归纳：

**优点**

1. 模型简单，启发了很多后续的图结构学习工作
2. 参数量小，便于优化

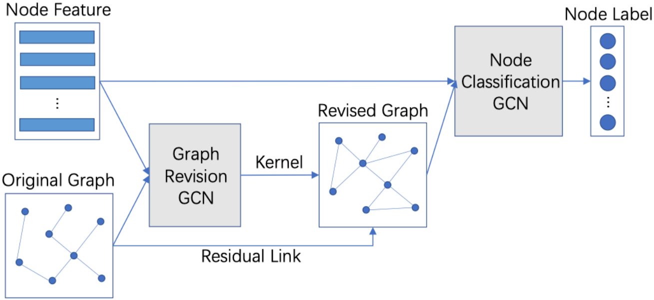
**缺点**

1. 结构仅通过线性变换学习
2. 学习的拉普拉斯矩阵过于稠密
3. 图神经网络没有给结构学习带来信息
4. 模型缺乏明确的结构学习优化目标

# Graph-Revised Convolutional Network(GRCN)

## 研究动机

图结构学习不应只增加边，稀疏性是图数据的天然特性。



GRCN 模型[5]

---

[5] Graph-Revised Convolutional Network(ECML-PKDD 2020)

# GRCN:Graph-Revised GCN

## Graph-Revised GCN 模块

利用原始的节点特征和图结构，学习节点之间基于表征 $z_i, z_j$ 的邻近性 $S_{ij}$ 进而更新图结构 $\widetilde{A}$。

$$Z = GCN_g(A, X)$$

$$S_{ij} = k\left(z_i, z_j\right)$$

$$\widetilde{A} = A + S$$

相比于 AGCN，这里使用了更为“暴力”的相似度学习方式。

# GRCN：图稀疏化

## Graph Sparsification 模块

直接将稠密相似度矩阵 $S$ 与原始邻接矩阵 $A$ 相加，会导致邻接矩阵非常稠密，因此使用 KNN 进行稀疏化。

$$S_{ij}^{(K)} = \begin{cases} S_{ij}, S_{ij} \in \operatorname{top} K\left(S_i\right) \\ 0, S_{ij} \notin \operatorname{top} K\left(S_i\right). \end{cases}$$

为了保持邻接矩阵的对称性，做对称性变换

$$\widehat{S}_{ij} = \begin{cases} \max\left(S_{ij}^{(K)}, S_{ji}^{(K)}\right), & S_{ij}^{(K)}, S_{ij}^{(K)} \geq 0 \\ \min\left(S_{ij}^{(K)}, S_{ji}^{(K)}\right), & S_{ij}^{(K)}, S_{ij}^{(K)} \leq 0 \end{cases}$$

# GRCN：模型训练

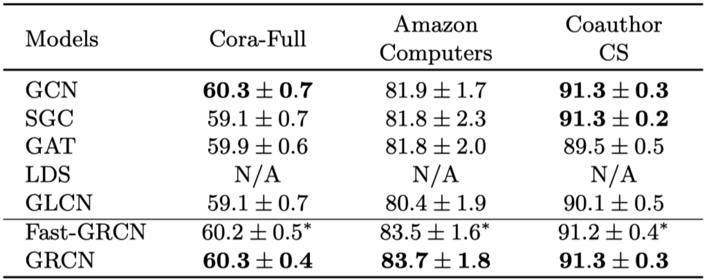
将更新之后的图结构与原始特征一起训练 Node Classification GCN 模块：

$$\widehat{Y} = GCN_c(\widetilde{A}, X)$$

$$\zeta = -\sum_{i \in \mathcal{Y}_L} \sum_{j=1}^{C} Y_{ij} \ln \widehat{Y}_{ij}$$

整个过程完全使用神经网络进行学习和优化。

# GRCN：实验效果



| Models | Cora-Full | Amazon Computers | Coauthor CS |
|---|---|---|---|
| GCN | **60.3 ± 0.7** | 81.9 ± 1.7 | **91.3 ± 0.3** |
| SGC | 59.1 ± 0.7 | 81.8 ± 2.3 | **91.3 ± 0.2** |
| GAT | 59.9 ± 0.6 | 81.8 ± 2.0 | 89.5 ± 0.5 |
| LDS | N/A | N/A | N/A |
| GLCN | 59.1 ± 0.7 | 80.4 ± 1.9 | 90.1 ± 0.5 |
| Fast-GRCN | 60.2 ± 0.5* | 83.5 ± 1.6* | 91.2 ± 0.4* |
| GRCN | **60.3 ± 0.4** | **83.7 ± 1.8** | **91.3 ± 0.3** |

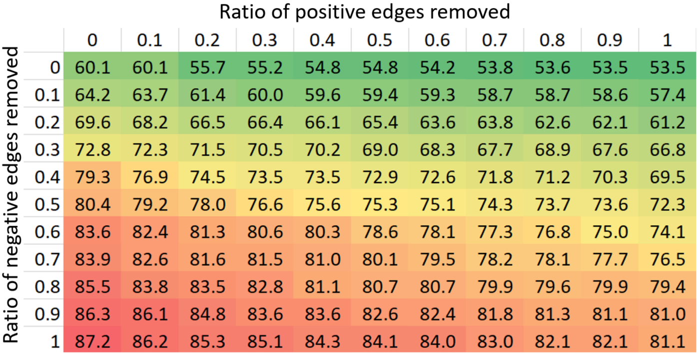
## 思考

GRCN 在什么情况下可以取得效果的提升?

# Learning to Drop: Robust Graph Neural Network via Topological Denoising

### 研究动机

网络中观察到的边往往存在与任务无关的边，通过这些节点传递信息往往给节点表征学习带来噪声。



Ratio of positive edges removed

Ratio of negative edges removed

| | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 60.1 | 60.1 | 55.7 | 55.2 | 54.8 | 54.8 | 54.2 | 53.8 | 53.6 | 53.5 | 53.5 |
| 0.1 | 64.2 | 63.7 | 61.4 | 60.0 | 59.6 | 59.4 | 59.3 | 58.7 | 58.7 | 58.6 | 57.4 |
| 0.2 | 69.6 | 68.2 | 66.5 | 66.4 | 66.1 | 65.4 | 63.6 | 63.8 | 62.6 | 62.1 | 61.2 |
| 0.3 | 72.8 | 72.3 | 71.5 | 70.5 | 70.2 | 69.0 | 68.3 | 67.7 | 68.9 | 67.6 | 66.8 |
| 0.4 | 79.3 | 76.9 | 74.5 | 73.5 | 73.5 | 72.9 | 72.6 | 71.8 | 71.2 | 70.3 | 69.5 |
| 0.5 | 80.4 | 79.2 | 78.0 | 76.6 | 75.6 | 75.3 | 75.1 | 74.3 | 73.7 | 73.6 | 72.3 |
| 0.6 | 83.6 | 82.4 | 81.3 | 80.6 | 80.3 | 78.6 | 78.1 | 77.3 | 76.8 | 75.0 | 74.1 |
| 0.7 | 83.9 | 82.6 | 81.6 | 81.5 | 81.0 | 80.1 | 79.5 | 78.2 | 78.1 | 77.7 | 76.5 |
| 0.8 | 85.5 | 83.8 | 83.5 | 82.8 | 81.1 | 80.7 | 80.7 | 79.9 | 79.6 | 79.9 | 79.4 |
| 0.9 | 86.3 | 86.1 | 84.8 | 83.6 | 83.6 | 82.6 | 82.4 | 81.8 | 81.3 | 81.1 | 81.0 |
| 1 | 87.2 | 86.2 | 85.3 | 85.1 | 84.3 | 84.1 | 84.0 | 83.0 | 82.1 | 82.1 | 81.1 |

# PTDNet: 图的稀疏化

Dropout 的思想被用于神经网络用以加强模型鲁棒性，
PTDNet[6]希望通过删除部分边来提升图神经网络的鲁棒性。

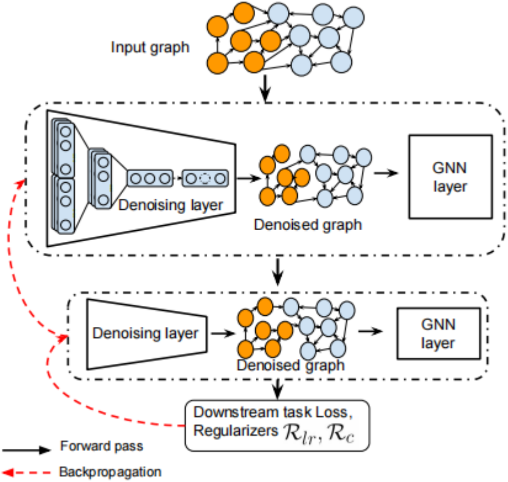
## 图的稀疏化和图结构学习

1. 图的稀疏化可以视作一种图结构学习，但仅在初始存在的边上进行，即可以去除冗余的边，但不能增加新边（后者往往需要耗费更多空间）。
2. 该不同之处通常表现在得到新的结构后与原结构乘还是加。

---

[6] Learning to Drop: Robust Graph Neural Network via Topological Denoising(WSDM 2021)

# PTDNet: 降噪网络

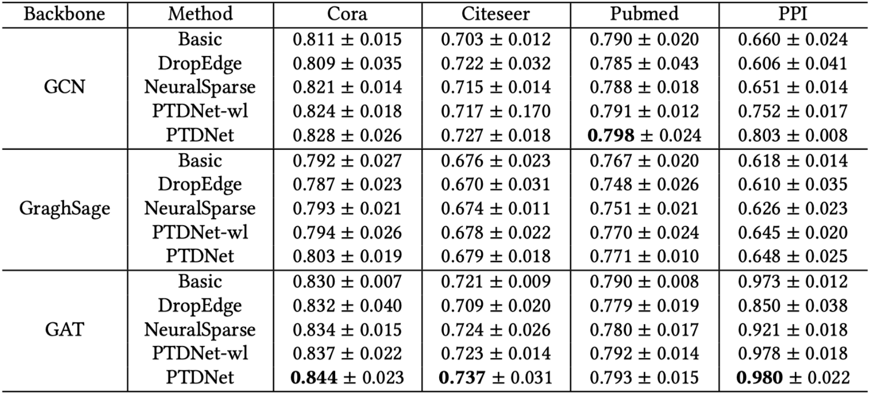


将节点间边的概率基于节点的表征，并用神经网络进行学习。

# PTDNet: 降噪网络

1. 每层 GCN 均配备一个降噪网络，使用上一层表征:$\alpha_{ij}^{l} = MLP^{l}(\vec{h}_i^l, \vec{h}_j^l)$
2. 重采样，使概率模型可导:
$\beta_{ij}^{l} = sigmoid((\log \epsilon - \log(1 - \epsilon) + \alpha_{ij}^{l})/\tau), \epsilon \sim Uniform(0, 1)$
3. 将其映射至 0-1 之间:$Z_{ij}^{l} = P_{\mathcal{S}}(\beta_{ij}^{l})$
4. 与原结构相乘（这里决定了无法添加新边）:$S^l = A \odot Z^l$
5. 卷积:$H^{l+1} = GCN^{l}(S^l, X)$

# PTDNet: 实验效果



| Backbone | Method | Cora | Citeseer | Pubmed | PPI |
|---|---|---|---|---|---|
| GCN | Basic | 0.811 ± 0.015 | 0.703 ± 0.012 | 0.790 ± 0.020 | 0.660 ± 0.024 |
| | DropEdge | 0.809 ± 0.035 | 0.722 ± 0.032 | 0.785 ± 0.043 | 0.606 ± 0.041 |
| | NeuralSparse | 0.821 ± 0.014 | 0.715 ± 0.014 | 0.788 ± 0.018 | 0.651 ± 0.014 |
| | PTDNet-wl | 0.824 ± 0.018 | 0.717 ± 0.170 | 0.791 ± 0.012 | 0.752 ± 0.017 |
| | PTDNet | 0.828 ± 0.026 | 0.727 ± 0.018 | **0.798** ± 0.024 | 0.803 ± 0.008 |
| GraghSage | Basic | 0.792 ± 0.027 | 0.676 ± 0.023 | 0.767 ± 0.020 | 0.618 ± 0.014 |
| | DropEdge | 0.787 ± 0.023 | 0.670 ± 0.031 | 0.748 ± 0.026 | 0.610 ± 0.035 |
| | NeuralSparse | 0.793 ± 0.021 | 0.674 ± 0.011 | 0.751 ± 0.021 | 0.626 ± 0.023 |
| | PTDNet-wl | 0.794 ± 0.026 | 0.678 ± 0.022 | 0.770 ± 0.024 | 0.645 ± 0.020 |
| | PTDNet | 0.803 ± 0.019 | 0.679 ± 0.018 | 0.771 ± 0.010 | 0.648 ± 0.025 |
| GAT | Basic | 0.830 ± 0.007 | 0.721 ± 0.009 | 0.790 ± 0.008 | 0.973 ± 0.012 |
| | DropEdge | 0.832 ± 0.040 | 0.709 ± 0.020 | 0.779 ± 0.019 | 0.850 ± 0.038 |
| | NeuralSparse | 0.834 ± 0.015 | 0.724 ± 0.026 | 0.780 ± 0.017 | 0.921 ± 0.018 |
| | PTDNet-wl | 0.837 ± 0.022 | 0.723 ± 0.014 | 0.792 ± 0.014 | 0.978 ± 0.018 |
| | PTDNet | **0.844** ± 0.023 | **0.737** ± 0.031 | 0.793 ± 0.015 | **0.980** ± 0.022 |

# Geometric Graph Convolutional Networks(Geom-GCN)

## 研究动机

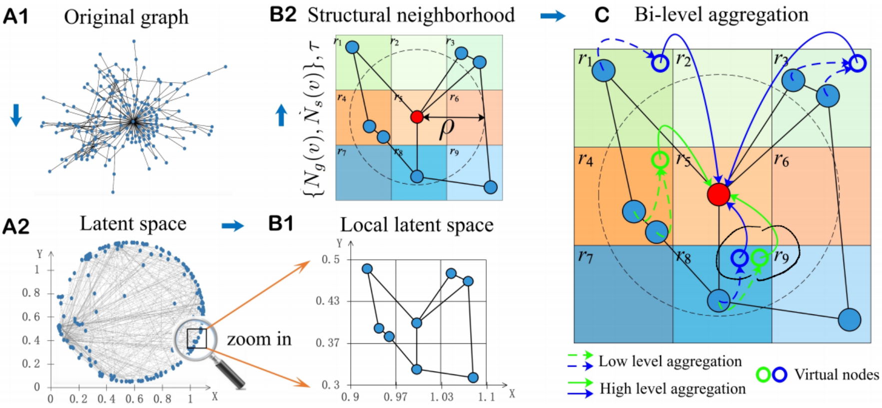
- 现有的基于消息传递机制的 GNN，容易损失长距离、全局结构信息，在同质性低（相邻节点倾向于不同类别）的图上表现不佳。
- 简单堆叠深层 GNN 产生过度平滑问题，一个簇中的节点表征经过多层消息传播趋同。

Geom-GCN[7]的解决思路：通过结构学习使得需要传递消息的节点对成为邻居。

---

[7] Geometric Graph Convolutional Networks(ICLR 2020)

# Geom-GCN

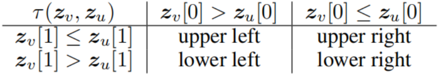


- 使用图嵌入方法，获得节点的隐空间表征。
- 根据这些表征，在图中构建新边。
- 为区分边的类型，改动 GCN 的消息传播方式

# Geom-GCN

实现步骤:

1. 使用 embedding method 获得图中节点隐空间表征:$e_v = f(v)$
2. 定义规则从隐空间中产生新边:
   $N_s(v) = \{u|u \in V, d(e_u, e_v) < \rho\}$，即在隐空间中相近的节点间构建新边。
3. 定义边的类型:$r = \tau(e_u, e_v), u \in N_g(v) \cup N_s(v)$
   (论文中隐空间表征 2 维，将其按位置关系简单分为 4 类，考虑边的来源总共为 8 类)



| $\tau(\boldsymbol{z}_v, \boldsymbol{z}_u)$ | $\boldsymbol{z}_v[0] > \boldsymbol{z}_u[0]$ | $\boldsymbol{z}_v[0] \leq \boldsymbol{z}_u[0]$ |
|---|---|---|
| $\boldsymbol{z}_v[1] \leq \boldsymbol{z}_u[1]$ | upper left | upper right |
| $\boldsymbol{z}_v[1] > \boldsymbol{z}_u[1]$ | lower left | lower right |

边类型的产生

# Geom-GCN

实现步骤:

1. 使用 embedding method 获得图中节点隐空间表征:$e_v = f(v)$
2. 定义规则从隐空间中产生新边:
   $N_s(v) = \{u | u \in V, d(e_u, e_v) < \rho\}$，即在隐空间中相近的节点间构建新边。
3. 定义边的类型:$r = \tau(e_u, e_v), u \in N_g(v) \cup N_s(v)$
4. 不同类型的边分别进行传播，之后再聚合

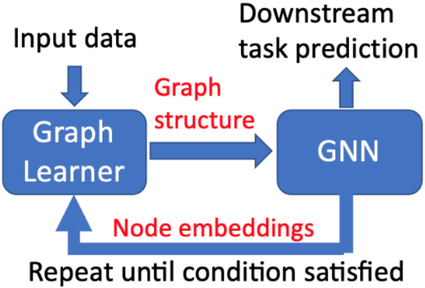
$$\boldsymbol{e}_{(i,r)}^{v,l+1} = \sum_{u \in N_i(v)} \delta\left(\tau\left(\boldsymbol{z}_v, \boldsymbol{z}_u\right), r\right) (\deg(v) \deg(u))^{\frac{1}{2}} \boldsymbol{h}_u^l$$

$$\boldsymbol{h}_v^{l+1} = \sigma\left(\boldsymbol{W}_l \cdot \prod_{i \in \{g,s\}} \|_{r \in R} \boldsymbol{e}_{(i,r)}^{v,l+1}\right)$$

# Iterative Deep Graph Learning for Graph Neural Networks: Better and Robust Node Embeddings

## 研究动机

图结构与图神经网络学习可以相互帮助：好的图结构可以学习好的图神经网络，好的图神经网络结果可以帮助建模更准确的网络结构。



# 从图结构学习到图神经网络

- 图结构沿用 AGCN 的思想：融合原始网络结构与学习过程中的网络结构

$$\tilde{\mathbf{A}}^{(t)} = \lambda \mathbf{L}^{(0)} + (1-\lambda)\left\{\eta \mathbf{f}\left(\mathbf{A}^{(t)}\right) + (1-\eta)\mathbf{f}\left(\mathbf{A}^{(1)}\right)\right\}$$

- 图神经网络沿用经典的 GCN：

$$\mathbf{Z} = \mathrm{ReLU}\left(\mathbf{MP}(\mathbf{X}, \tilde{\mathbf{A}})\mathbf{W}_1\right), \widehat{\mathbf{y}} = \sigma\left(\mathbf{MP}(\mathbf{Z}, \tilde{\mathbf{A}})\mathbf{W}_2\right)$$

$$\mathcal{L}_{\mathsf{pred}} = \ell(\hat{\mathbf{y}}, \mathbf{y})$$

# 根据图神经网络学习图结构

## 基本假设

具有相似的低维表征的节点对，有更大的概率生成边。

基于节点表征的边概率计算：

- 第一层：基于向量的相似度计算

$$S_{ij} = \cos\left(\vec{w}_p \odot \vec{z}_i, \vec{w}_p \odot \vec{z}_j\right)$$

- 第二层：基于多头向量的相似度计算

$$S_{ij} = \frac{1}{m}\sum_{p=1}^{m}\cos\left(\vec{w}_p \odot \vec{z}_i, \vec{w}_p \odot \vec{z}_j\right)$$

# 根据图神经网络学习图结构

**使用节点表征相似度预测图结构的缺陷：**

1. 计算复杂度高
2. 网络过于稠密

IDGL 的解决方案：

- 第三层：Anchor-based 度量学习：

$$a_{ik}^{p} = \cos\left(\mathbf{w}_p \odot \mathbf{v}_i, \mathbf{w}_p \odot \mathbf{u}_k\right), \quad a_{ik} = \frac{1}{m}\sum_{p=1}^{m} a_{ik}^{p}$$

- 第四层：图稀疏化。只选择每个节点最近的 K 个节点，或相似度大于阈值 $\epsilon$ 的邻居节点。

# 图结构质量控制

## IDGL 迭代优化的缺陷

图结构学习模块缺少监督信号，容易陷入局部平凡解 (Trival Solution)

好的图结构应具有如下性质：

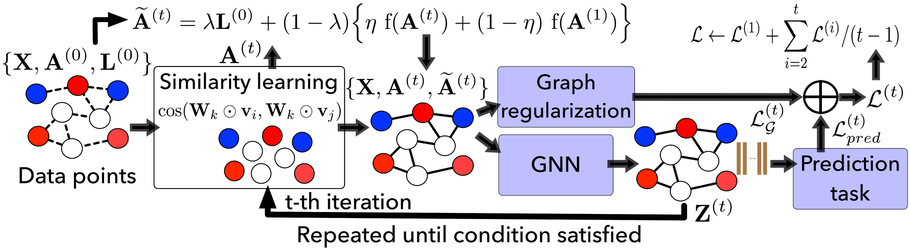
1. 平滑性：临近的节点的属性相似度应该逐渐下降

$$\Omega(\mathbf{A}, \mathbf{X}) = \frac{1}{2n^2} \sum_{i,j} A_{ij} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2 = \frac{1}{n^2} \operatorname{tr}\left(\mathbf{X}^T \mathbf{L} \mathbf{X}\right)$$

2. 连通性：网络中的节点尽可能构成联通图
3. 稀疏性：网络中的边总体稀疏

$$f(\mathbf{A}) = \frac{-\beta}{n} \mathbf{1}^T \log(\mathbf{A}\mathbf{1}) + \frac{\gamma}{n^2} \|\mathbf{A}\|_F^2$$

# IDGL 模型



IDGL 方法示意图

## 优点

1. 提出了统一的迭代学习框架
2. 明确了结构学习的优化目标
3. 使用多种手段兼顾效率与精度

# 循环优化图神经网络

一般形式：

1. 使用已有结构学习节点表征 $z$

$$\mathbf{z} = GNN(X, A)$$

2. 利用节点表征相似度学习图结构：

$$\tilde{\boldsymbol{A}}_{ij} = \phi\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)$$

3. 融合输入结构和学习图结构

$$\boldsymbol{A}^{\star} = g(\boldsymbol{A}, \tilde{\boldsymbol{A}})$$

迭代循环，直至收敛。

# 基于节点相似度的图结构生成

1. Gaussian Kernel

$$\phi\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right) = \sqrt{\left(\boldsymbol{z}_i - \boldsymbol{z}_j\right)^{\top} \boldsymbol{M} \left(\boldsymbol{z}_i - \boldsymbol{z}_j\right)}$$

$$\tilde{\boldsymbol{A}}_{ij} = \exp\left(-\frac{\phi\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)}{2\sigma^2}\right)$$

2. 向量内积

$$\tilde{\boldsymbol{A}} = \sigma\left(\boldsymbol{Z} Z^{\top}\right)$$

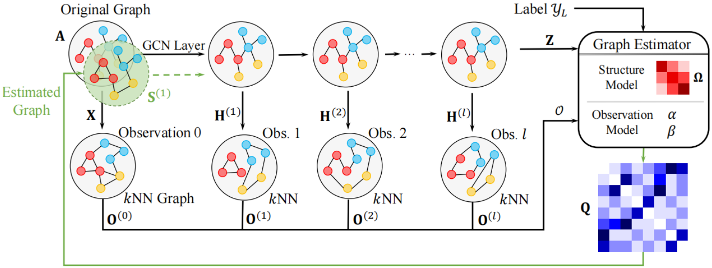
3. Cosine 相似度

$$\phi\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right) = \frac{\boldsymbol{z}_i \boldsymbol{z}_j^{\top}}{\left\|\boldsymbol{z}_i\right\|_2 \left\|\boldsymbol{z}_j\right\|_2}.$$

1. 研究背景
2. 基于优化的图结构学习
   - LDS
   - Pro-GNN
3. 基于图结构学习的图神经网络
   - AGCN
   - GRCN
   - PTDNet
   - Geom-GCN
   - IDGL
4. **基于生成模型的图结构学习**
5. 无监督的图结构学习

# Graph Structure Estimation Neural Networks

GEN 是一种概率建模 GSL 方法，相比之前这一类方法考虑图结构的内在产生过程，往往对这一过程参数化，并使用合适的优化方法与节点分类任务结合。

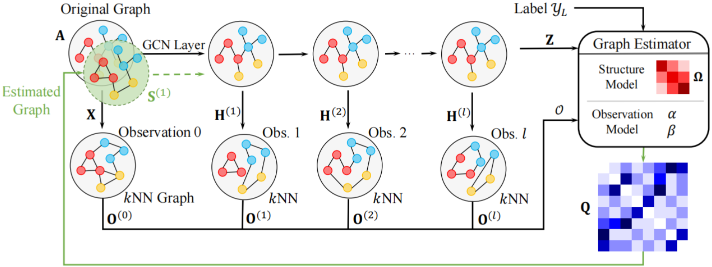


GEN 的模型示意图

# Graph Structure Estimation Neural Networks

## 研究动机

现有的图结构学习方法仅考虑图数据的判别特性（节点之间是否有边），对网络的生成过程缺乏可解释性。



GEN 的模型示意图[8]

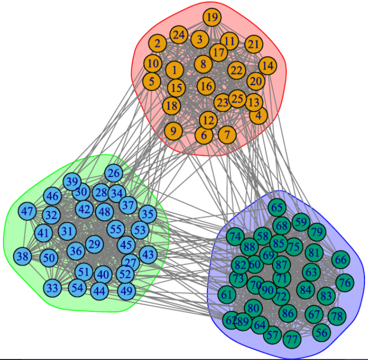
---

[8] [Wang et al., 2021]

# SBM 模型简介

## Stochastic Block Model(SBM)

Stochastic Block Model(SBM) 是一种经典的网络生成模型。它假设网络中边的生成是由两个节点的社区分布决定，即属于同一个社区的节点有更大的概率生成边，不同社区的节点生成边的概率更小。



# GEN 的图结构建模

在 SBM 模型的假设下，网络中存在边的概率只与两个节点所属的社区有关。给定参数矩阵 $\Omega \in \mathcal{S}^{|C|*|C|}$，观测到网络的概率为：

$$P(G|\Omega, Z, \mathcal{Y}_L) = \prod_{i<j} \Omega_{c_i c_j}^{G_{ij}} (1 - \Omega_{c_i c_j})^{1-G_{ij}}$$

$c_i$ 表示节点 $i$ 的类别。当其属于训练集时，使用真实标签；当属于无标签节点时，使用预测标签。

$$c_i = \begin{cases} y_i & \text{if } v_i \in \mathcal{V}_L \\ z_i & \text{otherwise} \end{cases}$$

# GEN 的图结构建模

**模型假设**

给定最优的图结构 G，真实观测的图结构由最优图结构随机采样后生成。

定义最优图结构下有边和没有边，在观测图结构下存在边的概率为 $\alpha$，　$\beta$，观测图结构的概率为：

$$P(O \mid \mathrm{G}, \alpha, \beta) = \prod_{i<j} [\alpha^{E_{ij}} (1-\alpha)^{M-E_{ij}}]^{G_{ij}} \times [\beta^{E_{ij}} (1-\beta)^{M-E_{ij}}]^{1-G_{ij}}$$

其中 $E_{ij}$ 为给定的一组 $M$ 个观测图结构中 $ij$ 之间出现边的次数，

# GEN 的模型优化

GEN 模型需要优化的参数有最优图结构 $G$ 和图神经网络参数 $\Theta$。然而参数优化存在两大类困难：

1. 两种参数高度耦合（迭代求解）
2. 最优图结构 $Q$ 为离散变量（转化为伯努利分布参数）

优化 GNN:

$$\theta \leftarrow \eta' \frac{\partial \mathcal{L}_{GNN}(\theta, S, X, \mathcal{Y}_L)}{\partial \theta}$$

优化图结构 S:

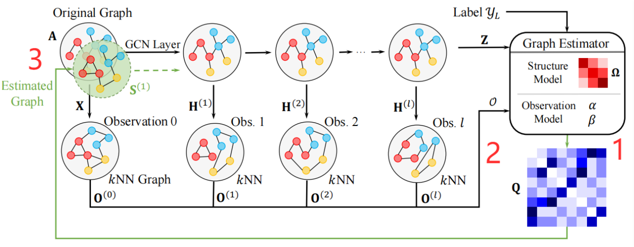
$$S \leftarrow EM(O^{(0)}, O^{(1)}, O^{(2)}, \ldots, O^{(l)}, O^{(l+1)})$$

式中 $O^{(i)}$ 为不同阶段的表征产生的 KNN 图。

# GEN

GEN 的基本思路:

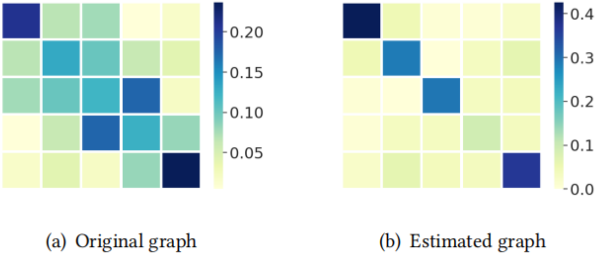
1. 假设存在一个图的最优结构，从一个SBM 模型中生成。
2. 假设该最优结构存在多个观测，从节点不同阶段的表示中各自生成 KNN 图作为观测。
3. 交替优化，在训练好的 GCN 基础上使用EM 算法求解最优结构，将其作为新的结构重新训练。



GEN 的模型示意图

# GEN: 同质性

在 toy dataset 上验证了:GEN 可以增加相同类别节点之间的边，减少连接不同类别的边。
体现了 GEN 可以学习具有同质性的图结构的能力。



(a) Original graph

(b) Estimated graph

GEN 产生结构的同质性

1. 研究背景
2. 基于优化的图结构学习
   - LDS
   - Pro-GNN
3. 基于图结构学习的图神经网络
   - AGCN
   - GRCN
   - PTDNet
   - Geom-GCN
   - IDGL
4. 基于生成模型的图结构学习
5. **无监督的图结构学习**

# Unsupervised GSL

以上的方法都基于有监督情况，结构学习依赖监督信息

## Drawbacks

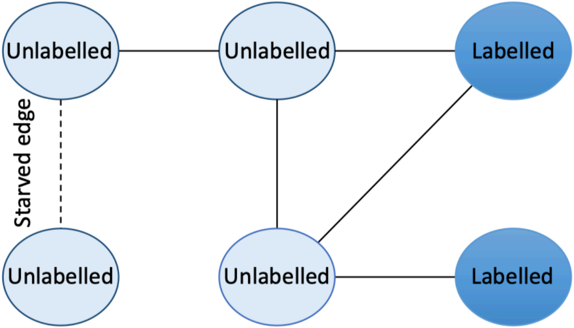
1. 不平衡。标签有限时有标签节点周围边受到监督信息引导，但距离有标签节点距离远的边难以学习。
2. 不实用。相较于监督情况，无监督情况更广泛。
3. 不通用。对于每个任务都要重新学习结构，不能学习一个通用的结构。

近期出现的 Unsupervised GSL 工作大多利用了对比学习的思想，关键在于如何设计不同的 view，如何融入结构学习。

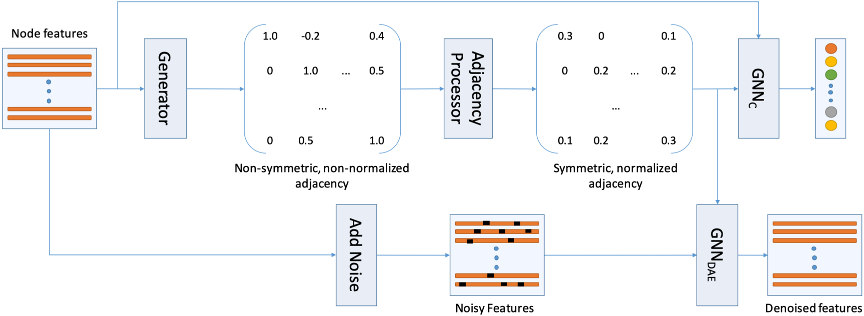
# SLAPS: Self-Supervision Improves Structure Learning for Graph Neural Networks

## 研究动机

传统的 GSL 方法，虽然使用了半监督模型帮助引导结构学习，但是并不是所有节点都会被惠及。

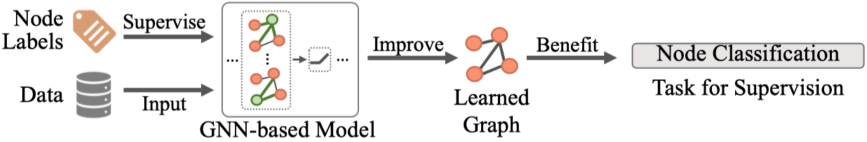


# SLAPS

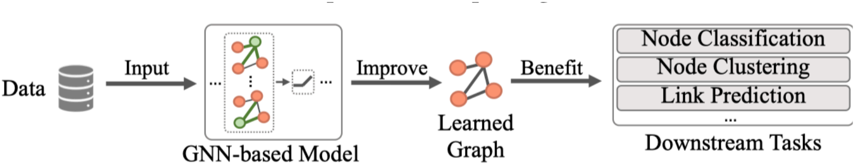


SLAPS 解决方案：引入 DAE 作为自监督信号

# Towards Unsupervised Deep Graph Structure Learning



有监督 GSL

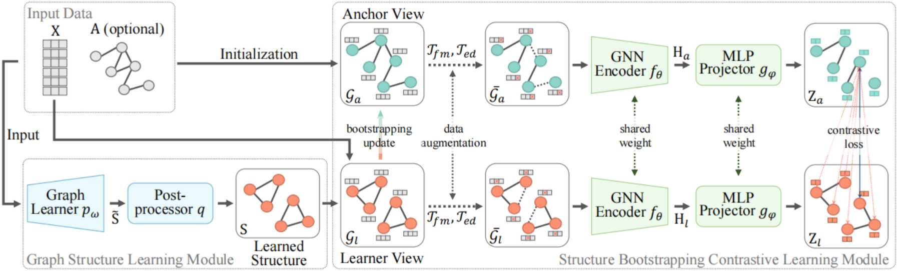


无监督 GSL[9]

---

[9] Towards Unsupervised Deep Graph Structure Learning(WWW 2022)

# Towards Unsupervised Deep Graph Structure Learning



可学习的 GSL 模块产生 learner view，原结构作为 anchor view，最大化两个 view 之间的互信息（MI）

# SUBLIME：Graph Learner

## Graph Learner

Graph Learner 用于学习图结构，本文讨论了多种传统的方法：

1. 直接优化
2. Metric Learning

1. 直接将整个图作为参数：

$$S = g(\Omega), \Omega \in R^{n\times n}$$

2. Learnable encoder+metric：

$$S = metric(encoder_w(X, A))$$

再经过后处理操作（对称化、稀疏化等）得到 learner view $G_l = (S, X)$

# SUBLIME：Graph Learner

Metric Learning based Graph Learner 的几种实现方式：

1. Attentive Learner：

$$\mathbf{E}^{(l)} = h_w^{(l)}\left(\mathbf{E}^{(l-1)}\right) = \sigma\left(\left[\mathbf{e}_1^{(l-1)} \odot \omega^{(l)}, \cdots, \mathbf{e}_n^{(l-1)} \odot \omega^{(l)}\right]^\top\right)$$

2. MLP Learner：

$$\mathbf{E}^{(l)} = h_w^{(l)}\left(\mathbf{E}^{(l-1)}\right) = \sigma\left(\mathbf{E}^{(l-1)}\Omega^{(l)}\right)$$

3. GNN Learner

$$\mathbf{E}^{(l)} = h_w^{(l)}\left(\mathbf{E}^{(l-1)}, \mathbf{A}\right) = \sigma\left(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{E}^{(l-1)}\ {}^{(l)}\right)$$

# SUBLIME：Post-processor

## Post-processor

Post-processor 用于处理 Graph Learner 学到的结果，以便于下游应用和训练。

图结构稀疏化 Sparsification：

$$\tilde{\mathbf{s}}_{ij}^{(sp)} = q_{sp}\left(\tilde{\mathbf{S}}_{ij}\right) = \begin{cases} \tilde{\mathrm{s}}_{ij}, & \tilde{\mathbf{S}}_{ij} \in \mathrm{top}-\mathrm{k}\left(\tilde{\mathbf{S}}_{i}\right), \\ 0, & \tilde{\mathbf{S}}_{ij} \notin \mathrm{top}-\mathrm{k}\left(\tilde{\mathbf{S}}_{i}\right), \end{cases}$$

图归一化：

$$\tilde{\mathbf{S}}^{(sym)} = q_{\mathsf{sym}}\left(q_{\mathsf{act}}\left(\tilde{\mathbf{S}}^{(sp)}\right)\right) = \frac{\sigma_q\left(\tilde{\mathbf{S}}^{(sp)}\right) + \sigma_q\left(\tilde{\mathbf{S}}^{(sp)}\right)^{\top}}{2}$$

# Contrastive Learning

得到 learner view $G_l(A, S)$ 后和预先指定的 anchor view（文中设置为原图 $G_a = (A, X)$）进行对比学习。
使用共享的 GCN+MLP 得到表征：

$$Z_l = g_\phi(f_\theta(\overline{G}_l)), Z_a = g_\phi(f_\theta(\overline{G}_a))$$

对比学习最大化互信息：

$$\mathcal{L} = \frac{1}{2n} \sum_{i=1}^{n} \left[ \ell\left(z_{l,i}, z_{a,i}\right) + \ell\left(z_{a,i}, z_{l,i}\right) \right]$$

$$\ell\left(z_{l,i}, z_{a,i}\right) = \log \frac{e^{\operatorname{sim}\left(\mathrm{z}_{l,i}, \mathrm{z}_{a,i}\right)/t}}{\sum_{k=1}^{n} e^{\operatorname{sim}\left(\mathrm{z}_{l,i}, \mathrm{z}_{a,k}\right)/t}}$$

# 更新 Anchor view

一直使用原图作为 anchor view 会带来原结构中的误差，文中每隔若干轮次更新，和当前学习的结构加权：

$$A_a = \tau A_a + (1 - \tau) S$$

# Reliable Representations Make A Stronger Defender: Unsupervised Structure Refinement for Robust GNN(STABLE)

## 研究动机

聚焦基于距离度量的 GSL 方法，认为监督情况下得到的表征容易受扰动，本身质量不佳，不能产生好的结构。

STABLE[10]解决方案：

1. 使用对比学习得到更可靠的表征
2. 利用更可靠的表征更新图结构

---

[10] Reliable Representations Make A Stronger Defender: Unsupervised Structure Refinement for Robust GNN(KDD 2022)

# STABLE: Representation Learning

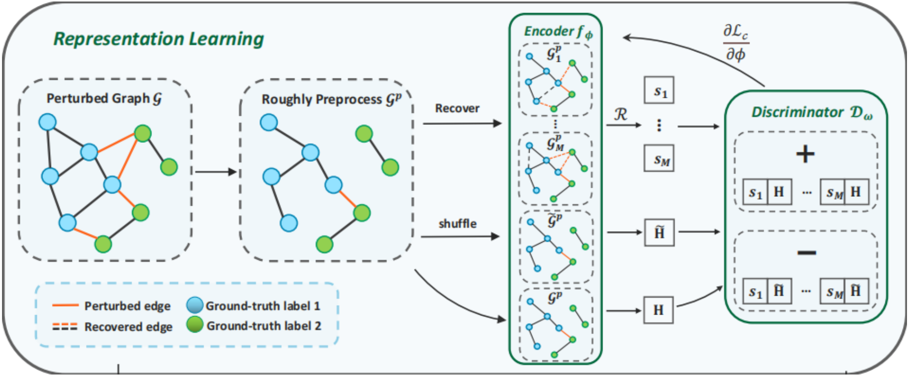
**基本假设**

实际对抗攻击过程，往往是增加边而不是删除边。

对于一个含噪声的图 $G$：

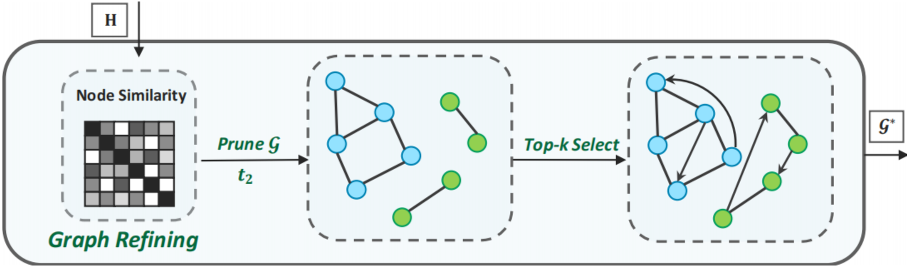
1. 删除一部分边，得到噪声程度低的图 $G^p$
2. 从 $G^p$ 中随机恢复被删除的边，构成 $M$ 个噪声程度较高的图 $G^p_1, G^p_2, ...G^p_M$
3. 要求从 $G^p$ 和 $G^p_1, G^p_2, ...G^p_M$ 得到的表征接近，同时 shuffle$G^p$ 得到 $\tilde{G}^p$ 作为负样本，要求 $\tilde{G}^p$ 和 $G^p_1, G^p_2, ...G^p_M$ 得到的表征接近

# STABLE: Representation Learning



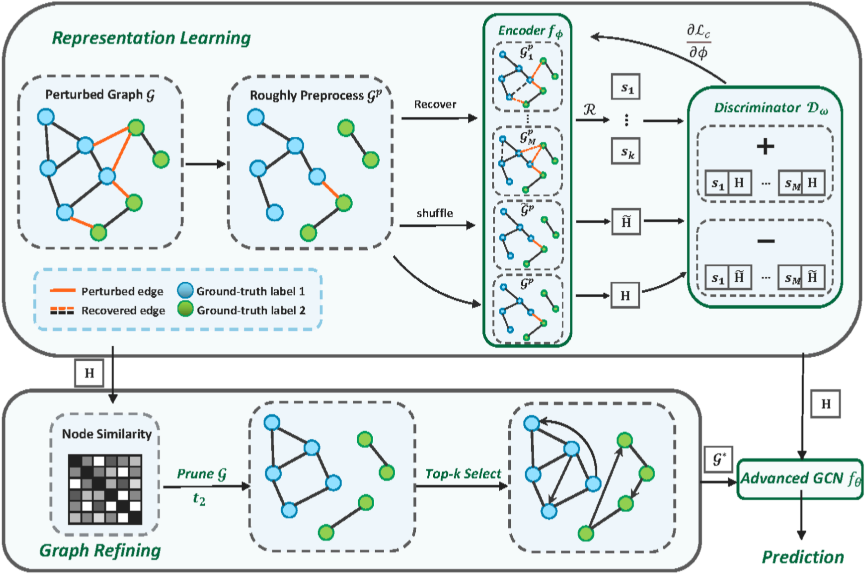
# STABLE: Graph Refining

在获得 reliable 的表征 $H$ 后，使用常规的相似度计算和后处理得到最终的结构：



该结构与任务无关的，从无监督方法获得，可以直接用于多个下游任务。

# STABLE


Representation Learning
Perturbed Graph $\mathcal{G}$
Roughly Preprocess $\mathcal{G}^p$
Recover
shuffle
Encoder $f_\phi$
$\mathcal{G}^p_1$
$\mathcal{G}^p_M$
$\tilde{\mathcal{G}}^p$
$\mathcal{G}^p$
$\mathcal{R}$
$s_1$
$s_k$
$\tilde{H}$
H
$\frac{\partial \mathcal{L}_c}{\partial \phi}$
Discriminator $\mathcal{D}_\omega$
+
$s_1$ H ··· $s_M$ H
−
$s_1$ $\tilde{H}$ ··· $s_M$ $\tilde{H}$
Perturbed edge
Recovered edge
Ground-truth label 1
Ground-truth label 2
H
Node Similarity
Prune $\mathcal{G}$
$t_2$
Top-k Select
Graph Refining
$\mathcal{G}^*$
H
Advanced GCN $f_\theta$
Prediction

# 未来展望

当前图结构学习的问题：

1. 图结构学习，只假设网络结构有问题，没有关注网络中节点属性的问题
2. 半监督信号能否指导全局节点的结构学习
3. 如何避免模型训练过程中的 error propagation
4. ...

图结构学习的端到端应用：

1. 图异常检测
2. 社区发现
3. 图对抗攻防
4. ...

# 课程总结

1. 研究背景
2. 基于优化的图结构学习
    - LDS
    - Pro-GNN
3. 基于图结构学习的图神经网络
    - AGCN
    - GRCN
    - PTDNet
    - Geom-GCN
    - IDGL
4. 基于生成模型的图结构学习
5. 无监督的图结构学习

# References I

- Jin, W., Ma, Y., Liu, X., Tang, X., Wang, S., and Tang, J. (2020).
  Graph structure learning for robust graph neural networks.
  In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 66–74.
- Wang, R., Mou, S., Wang, X., Xiao, W., Ju, Q., Shi, C., and Xie, X. (2021).
  Graph structure estimation neural networks.
  In *Proceedings of the Web Conference 2021*, pages 342–353.