**Content**

**Basic:GNN NAS**

1. **GraphNAS: Graph Neural Architecture Search with Reinforcement Learning**

**2.Auto-GNN: Neural Architecture Search of Graph Neural Networks**

**3.** **Simplifying Architecture Search for Graph Neural Network**

**4.Evolutionary Architecture Search for Graph Neural Networks**

**Basic:Biology**

1. **AffinityNet: Semi-supervised Few-shot Learning forDisease**

**Type Prediction**

**Title:**

**Basic:**

**x**

**Introduction:**

**方案：**

**实验验证:**

**Data:**

**x**

**Method:**

**x**

[**Evaluation**](javascript:;)**:**

**x**

**Advantage/Disadvantage:**

Advantage:

**x**

Disadvantage:

**x**

**Result:**

**x**

**Code:**

**x**

**Future Work:**

**Title:**

**GraphNAS: Graph Neural Architecture Search with Reinforcement Learning**

**Basic: GNN NAS**

[x].Gao, Y., Yang, H., Zhang, P., Zhou, C., & Hu, Y. (2019). *GraphNAS: Graph Neural Architecture Search with Reinforcement Learning*. Arxiv.*April*.I <http://arxiv.org/abs/1904.09981>

[X].Gao, Y., Yang, H., Zhang, P., Zhou, C., & Hu, Y. (2020). ***Graph Neural Architecture Search*.** 1403–1409.ijcai.2020/195

**Introduction:**

**方案：**

**1.GNN Search Space(5 classes)**

**2.controller——>LSTM(policy model),1 encoder,5 decoder**

**2.1.sample action based on P (not select max p action directly)**

**3.GNN val\_score——>reward**

**4.reward+action\_entropy\*c——>action\_reward**

**5.moving\_average处理reward**

**6.policy loss——>controller**

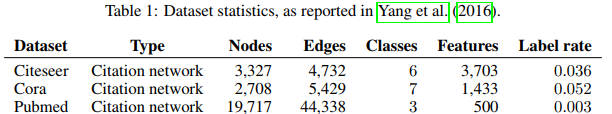
**实验验证：**



**Data:**

**1.Transductive Learning Task：**

**学术文献引用网络**



**数据集url：https://github.com/tkipf/gcn/tree/master/gcn/data**

**2.Inductive Learning Task：**

**PPI蛋白质相互作用网络**

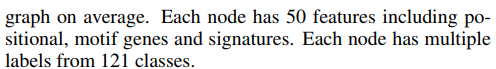
**(1).训练集：24 graphs,验证集：2 graphs, 测试集：2 graphs**

**(2).图平均节点数：2372**

**(3).单个节点特征维度：50**

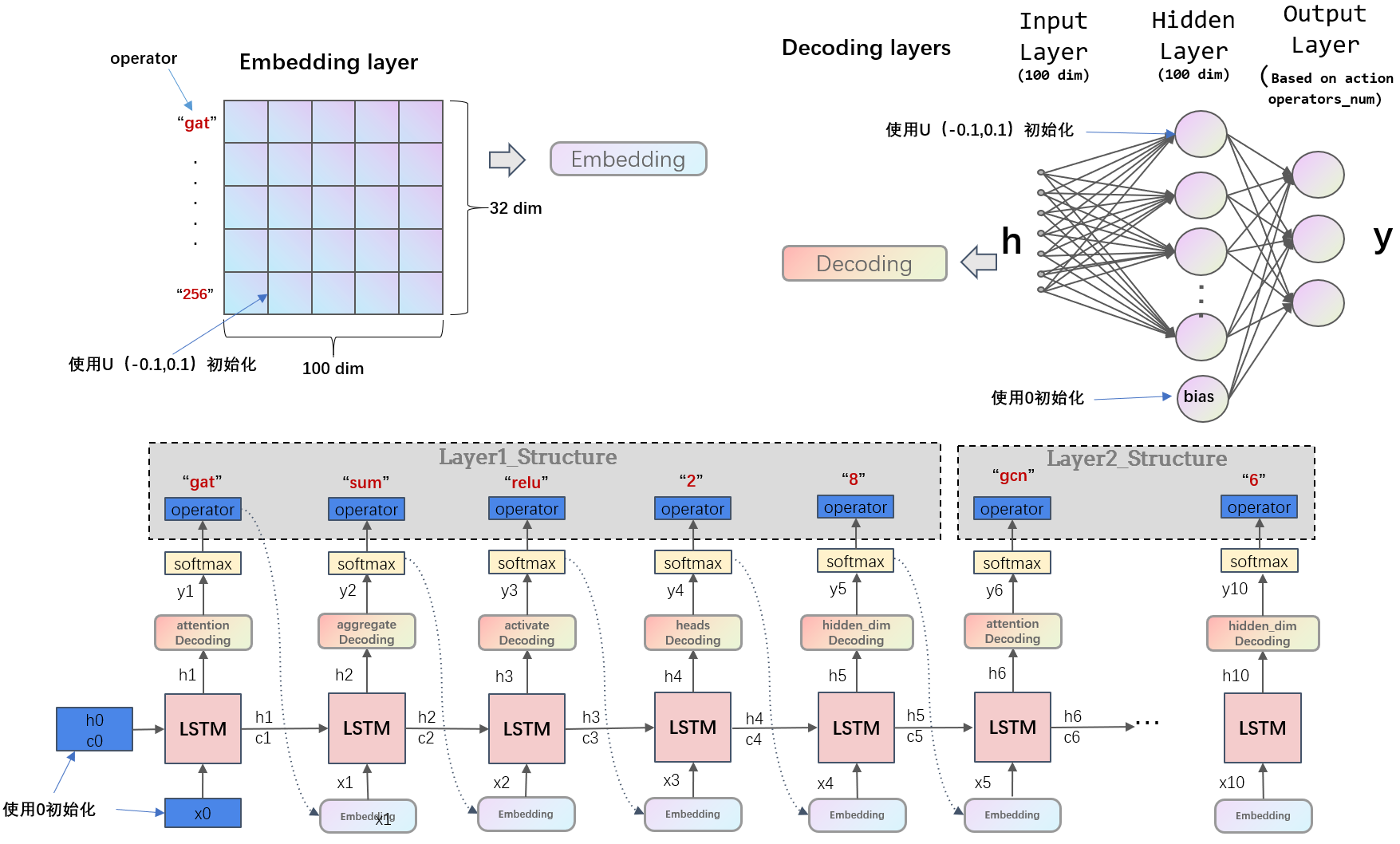
**(4).类别数：121类**





**Method:**

1.GNN 结构采样过程



2.基于强化学习训练controller

3.基于训练后的controller采样GNN验证GNN结构

4.挑选val\_acc top5 GNN 进行验证

5.选出最佳best structrue

[**Evaluation**](javascript:;)**:**

**实验1：**

**AUC**

**试验2：**

**Micro-F1**

**Advantage/Disadvantage:**

**Advantage:**

**1.reinforment learning + GNNnas**

**2.GNN search space**

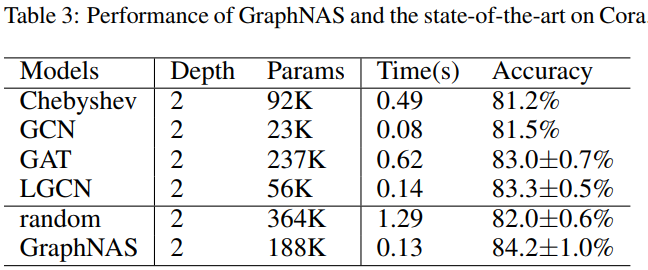
**Disadvantage:**

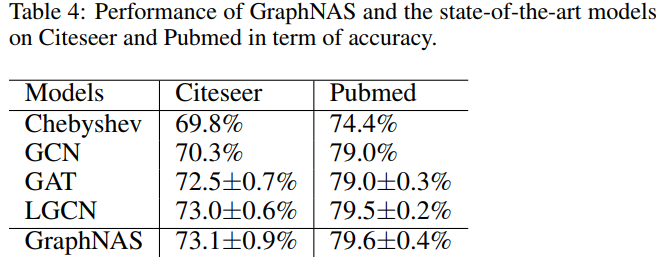
**1.搜索顺序是否可以更自由？(搜索顺利变更对效果影响意义不大，搜索目标是没有序列依赖关系)**

**2.得到 best structure 能否对局部组件微调**

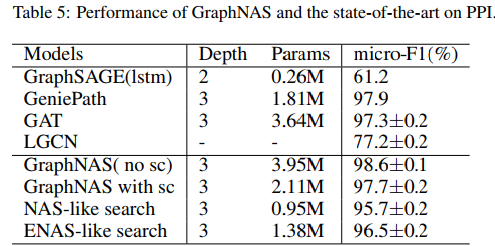
**Result:**

**实验1：**

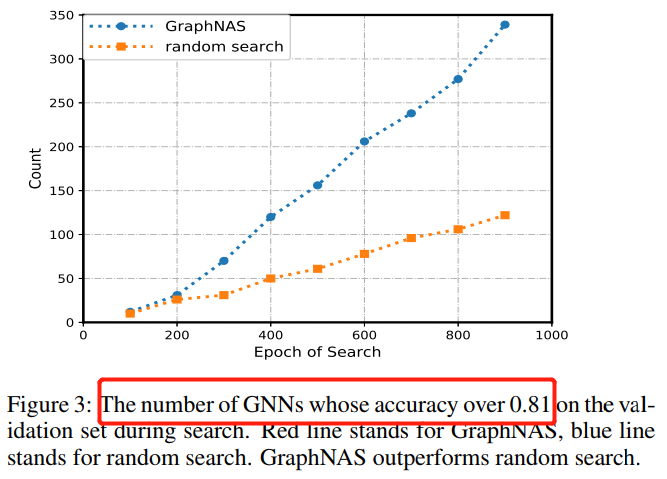


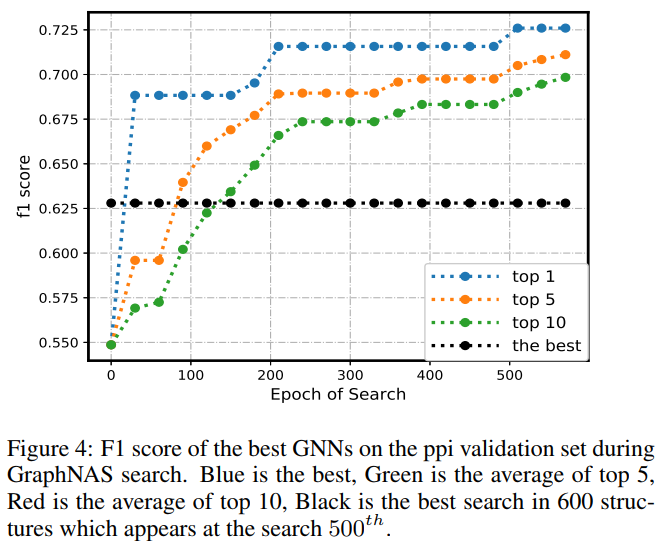


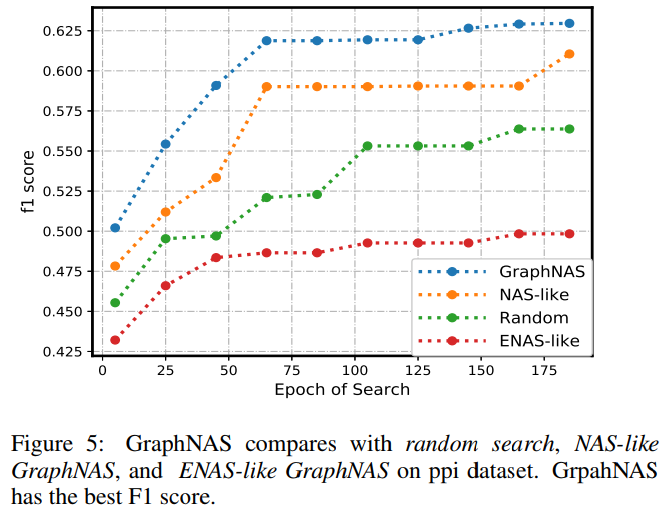
**实验2：**



**搜索效率与效果验证：**







**Code:可运行**

**Url:** [**https://github.com/GraphNAS/GraphNAS**](https://github.com/GraphNAS/GraphNAS)

**GNN库：**

**1.**[**https://github.com/dmlc/dgl**](https://github.com/dmlc/dgl)

**2.**[**https://github.com/rusty1s/pytorch\_geometric**](https://github.com/rusty1s/pytorch_geometric)

**Future Work:**

**Title:**

**AffinityNet: Semi-supervised Few-shot Learning forDisease Type Prediction**

**Basic:Biology**

[x].Ma, T., & Zhang, A. (2019). AffinityNet: Semi-Supervised Few-Shot Learning for Disease Type Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*, 1069–1076. https://doi.org/10.1609/aaai.v33i01.33011069

**Introduction:**

**方案：**

**1.基于attention相似度构图**

**2.基于特征提取器对单个特征进行特征筛选**

**3.基于GAT进行特征融合**

**4.小样本学习策略**

**实验验证：**

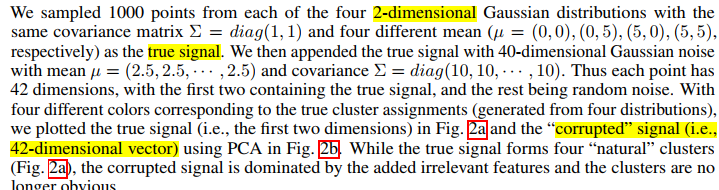
**1.癌症分类**

**2.癌症生存率估计**

**Data:**

**实验1：**

**42维特征向量=2维特征+40维高斯噪声，**

****

**数据集情况：**

**4个类，4000个样本**

**实验2:**

**url:** [**https://portal.gdc.cancer.gov**](https://portal.gdc.cancer.gov)

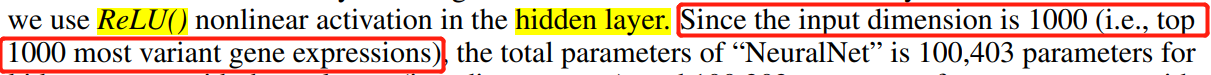
****

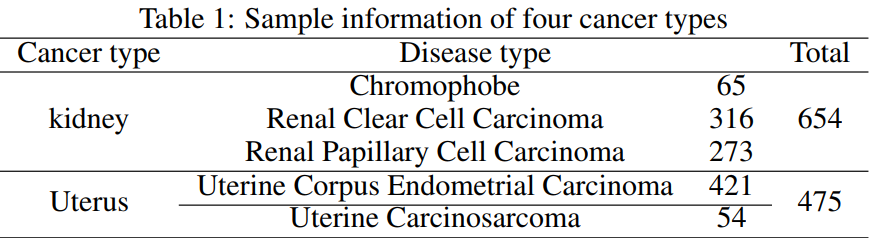
**数据集相关文献**

****

**数据集情况：**

**每个样本1000维**



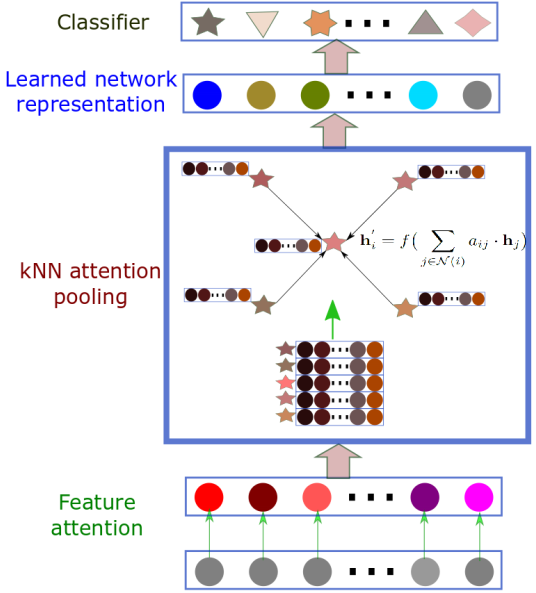


**实验3：**

**使用实验2中kindey数据集**

**Method:**

**模型架构:**

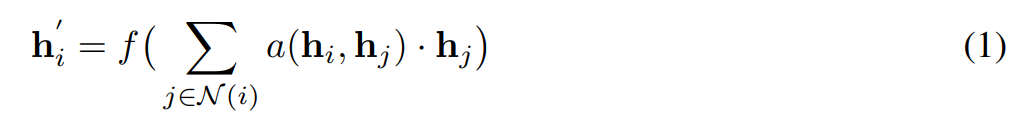


**1.Similarity graph 构造**

**基于attention核计算全部数据之间的相似度，以相似度作为构图依据构图**

**2.KNN attention pooling layer**

**(1).节点表达：**

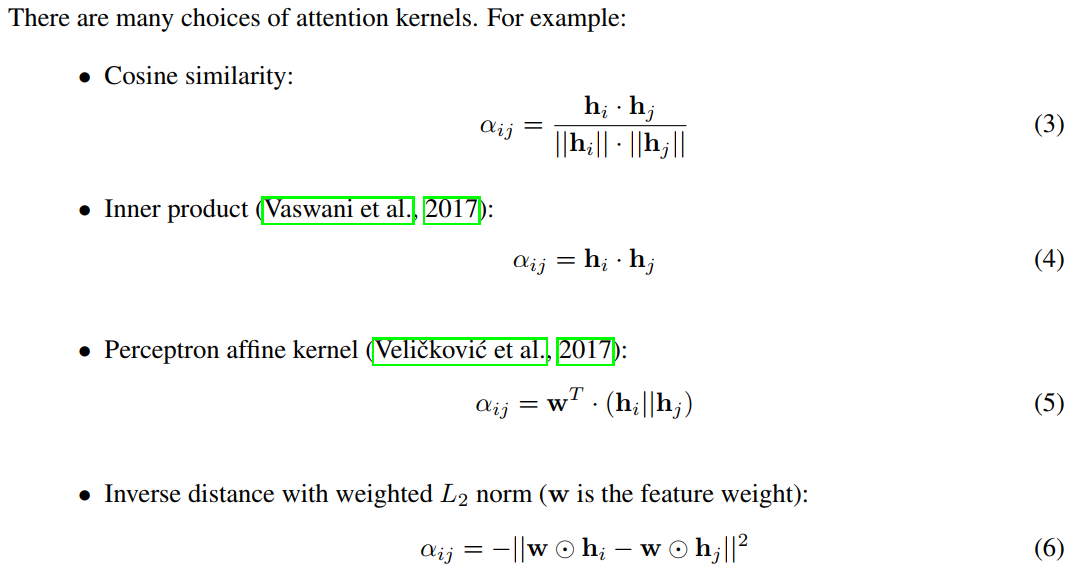


**(2).注意力机制：**



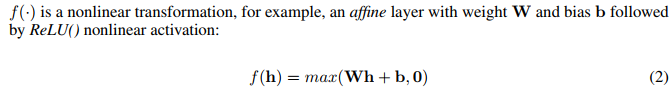








**(3).Relu激活函数：**



**(4).KNN Pooling操作：**

**对图规模大且节点具有高度的邻居节点选择，基于KNN思想，选择中心节点相似度最高的k个邻居节点聚合。**



**pooling在图像特征提取中的作用：**

**1.增加平移不变性**

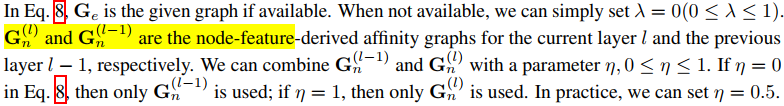
**2.保留主要特征/减少模型训练参数**

**本文pooling操作体现在基于attention kernel计算central node 与 neighbor nodes的similarity ,并以similarity来选择 k 个neighbor nodes起到减少central node聚合neighbor nodes数量的以达到类似图像pooling中第二个作用。**

**(5).Dynamic affinity graph操作：**

**每层图节点表示考了原始图节点表示，上一层图节点表示**





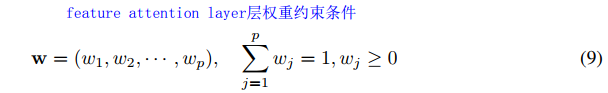
**4.Feature Attention Layer**

**通过有监督信号，对样本特征进行筛选**

**(1).节点表达**



**(2).权重约束**

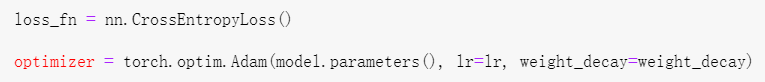


**5.Semi-supervised few-shot learning机制**

**6.损失函数与优化函数**

**损失函数：交叉熵损失**

**优化函数：Adam**



[**Evaluation**](javascript:;)**:**

**实验1：**

**1.training loss值**

**2.AUC**

**3.feature attention layer特征选择评估**

**实验2：**

**AMI评估**

**(https://www.jianshu.com/p/b9528df2f57a)**

**实验3：**

**1.Concordance index: 主要用于计算生存分析中的COX模型预测值与真实之间的区分度(https://www.jianshu.com/p/5e648f0f49ed)**

**2.基于Wilcoxon signed rank test的P值检验**

**3.Kaplan-Meier生存估计(https://zhuanlan.zhihu.com/p/97645982)**

**Advantage/Disadvantage:**

Advantage:

**1.数据组织：将独立非图结构数据依据相似度构图思想以图的形式进行组织并使用GAT对图数据进行融合表示;**

**2.卷积域选择：使用KNN思想选择卷积域范围并可实现每层卷积动态选择卷积域内邻居节点;**

**3.提出一种卷积层图表示融合的机制，增强卷积过程中图表示的平滑度;**

**4.加入单个样本特征选择机制并融入训练中学习选择权重;**

**5.基于训练好的特征选择器，定义聚类/分类计算中的相似度计算;**

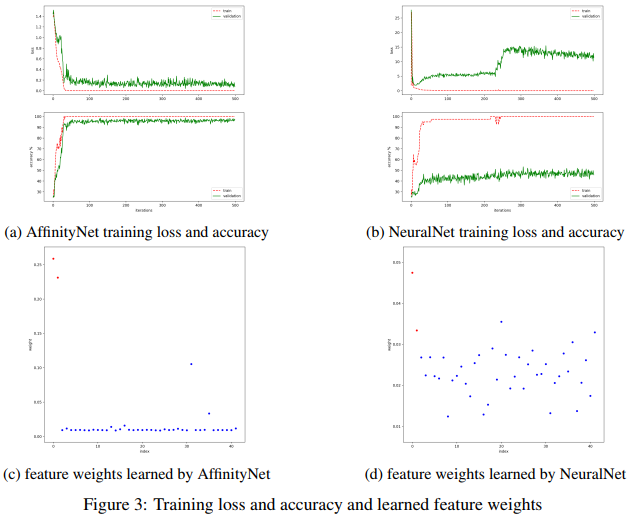
**6.做了很多生物信息学的实验来验证方法的有效性;**

Disadvantage:

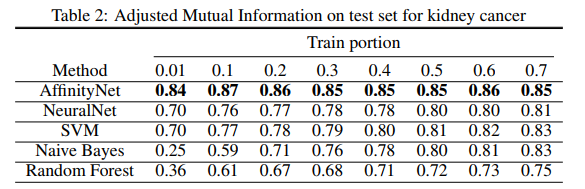
**1.对图结构数据挖掘没有特别优势**

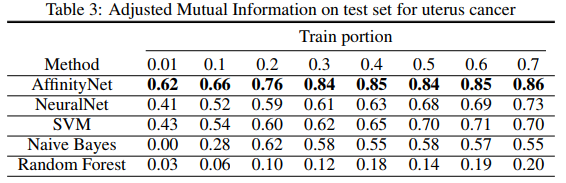
**Result:**

**实验1：**

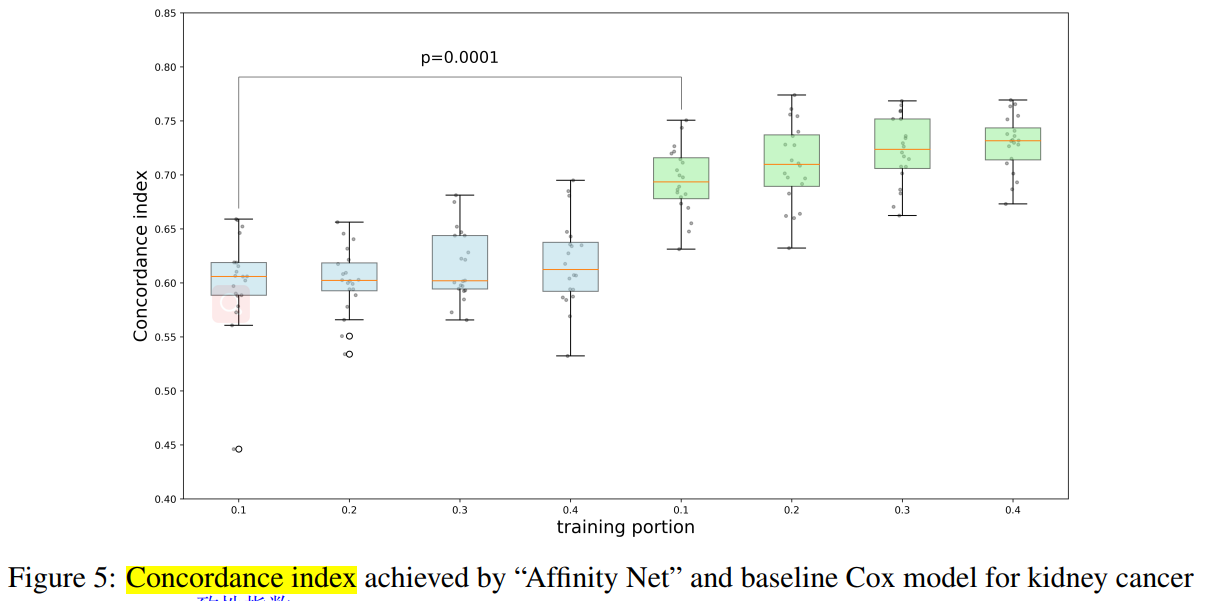


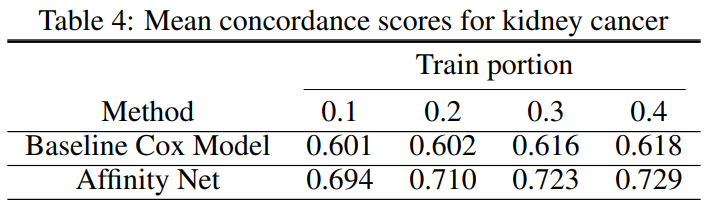
**实验2：**

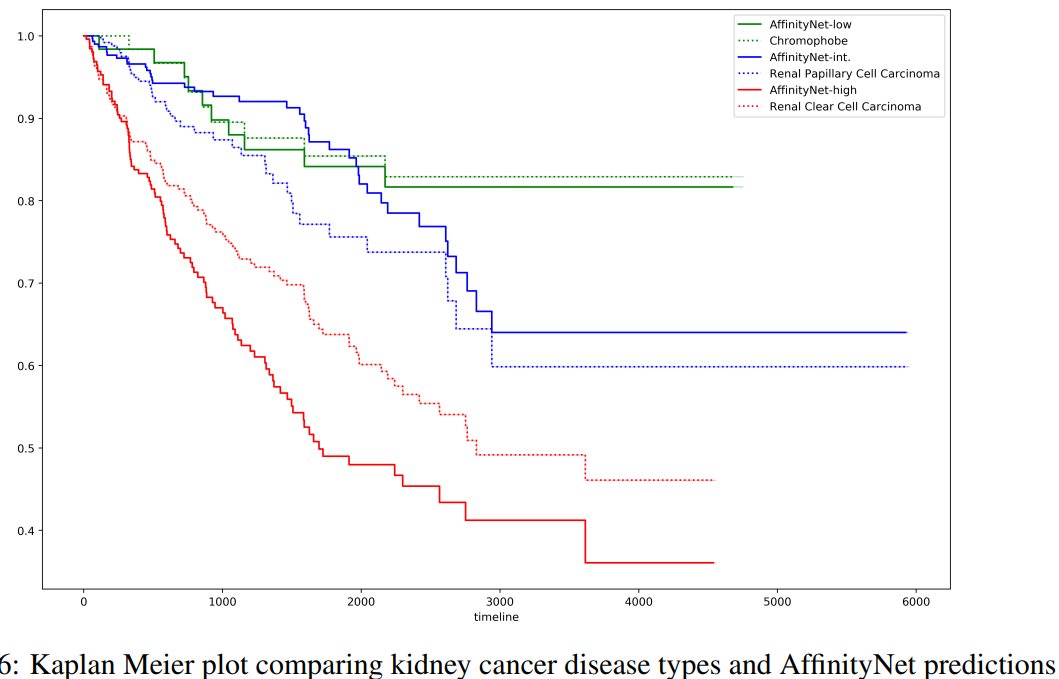




**试验3：**







**Code:可运行**

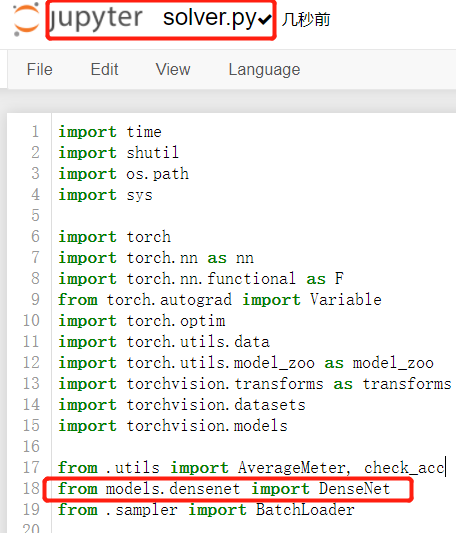
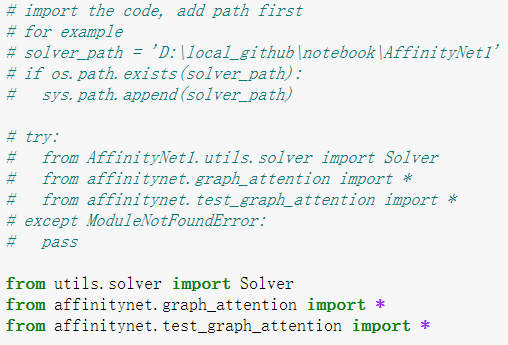
**url：https://github.com/BeautyOfWeb/AffinityNet.**

**复现实验1：**

**对比在不同比例训练情况下模型AUC与样本特征选择器效果**

****

**原始脚本修改：**



**Future Work:**

**Title:**

**Auto-GNN: Neural Architecture Search of Graph Neural Networks**

**Basic:**

**GNN NAS**

**Introduction:**

**方案：**

**假设： GNN架构中局部组件的变动对整体的效果提升明显**

**1.GNN Search Space (6 classes)**

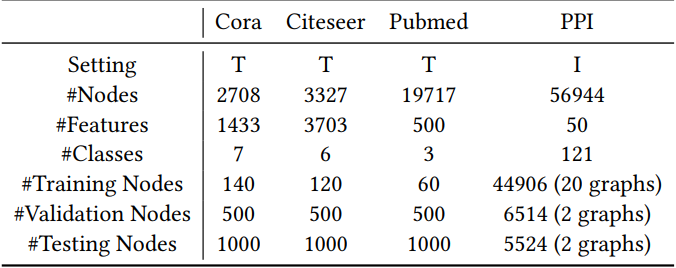
**2.controller——>6 RNN encoders for each class, 1 embeding layer**

**实验验证:**

**(1).半监督：**

**(2).有监督：**

**Data:**



**Method:**

**1.搜索空间定义**

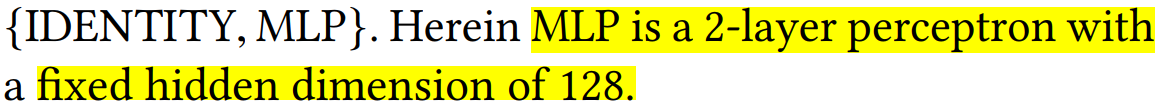




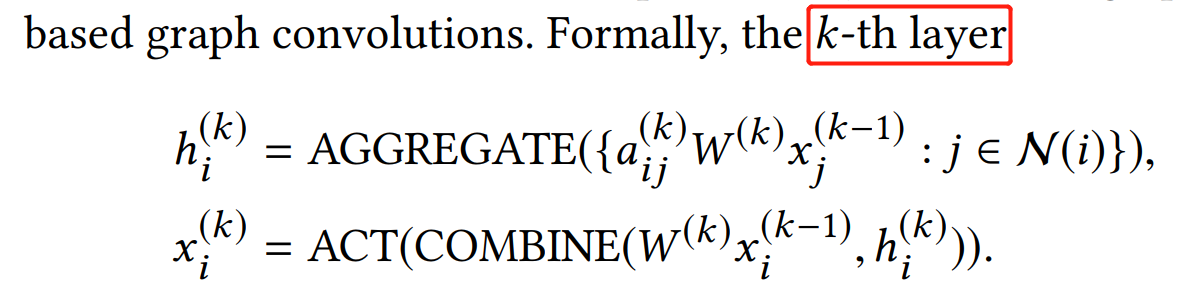




**——>Skip Connection function**



**第k层h(k),x(k)embedding表示：**





**2.controller设计**

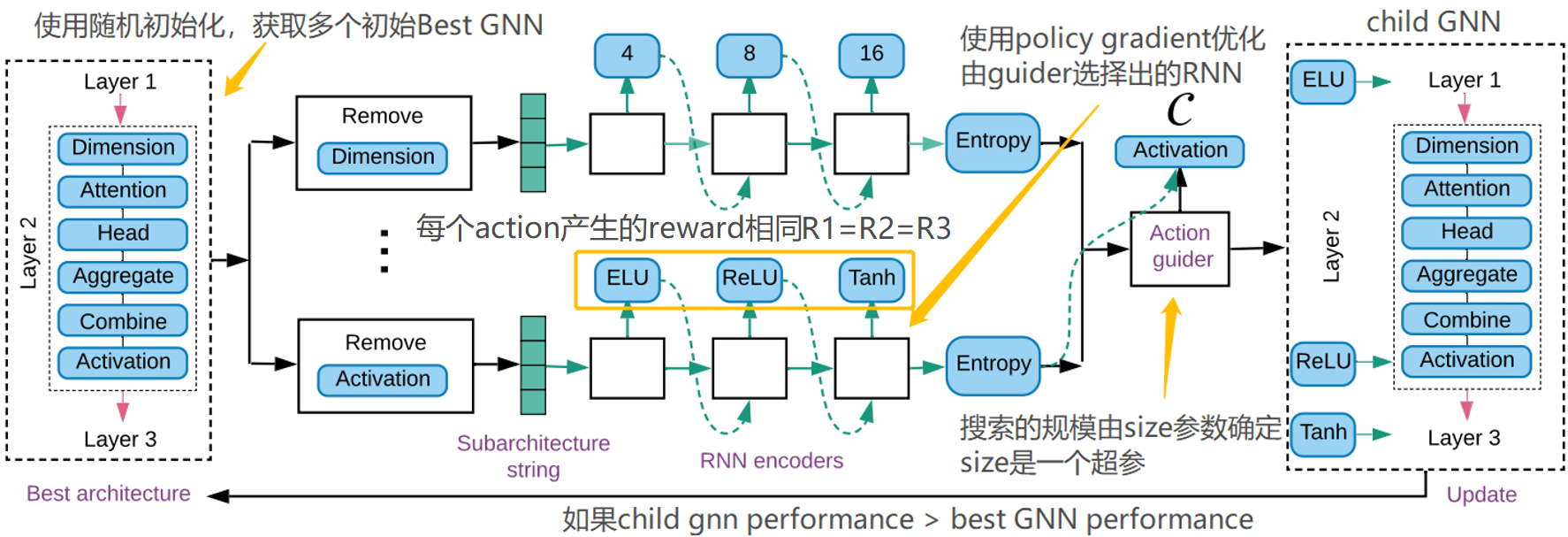
**2.1.explorer：**

**(1).随机初始化多个GNN Structrue**

**(2).筛选目前val\_accuracy最佳的GNN结构**

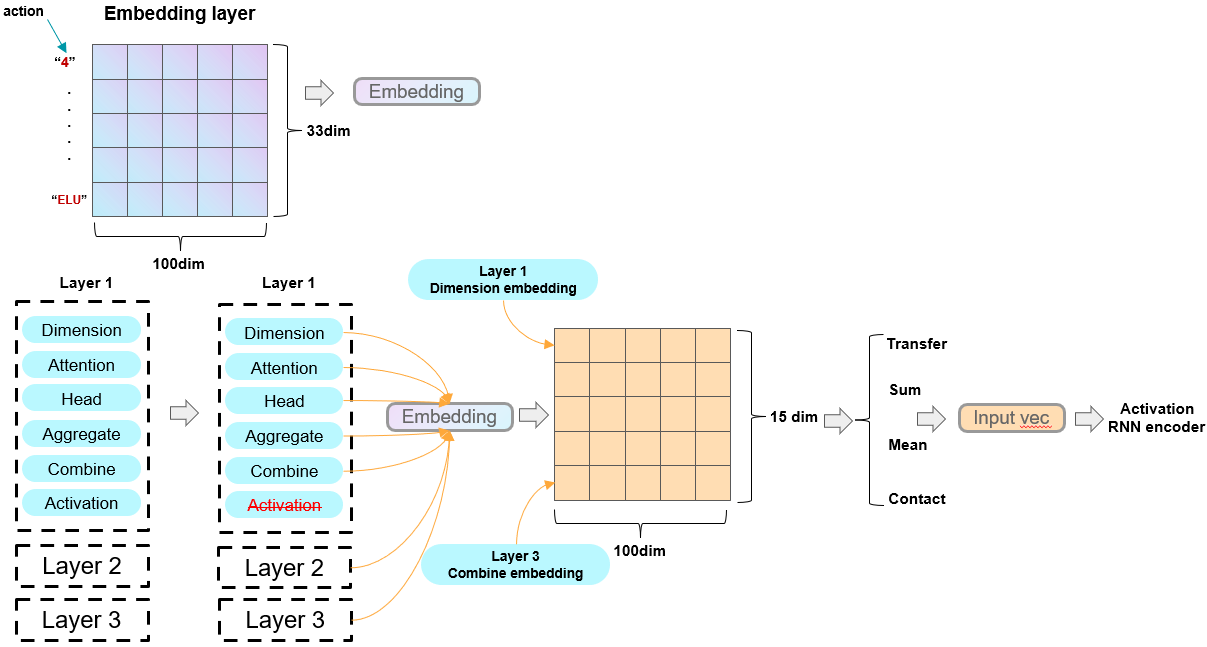
**当经过 modifier处理产生的child GNN val\_accuracy超过目前best GNN，使用child GNN代替目前best开始下一轮Search，否则best GNN不变作为下一轮搜索的基准**

**2.2.modifier:**



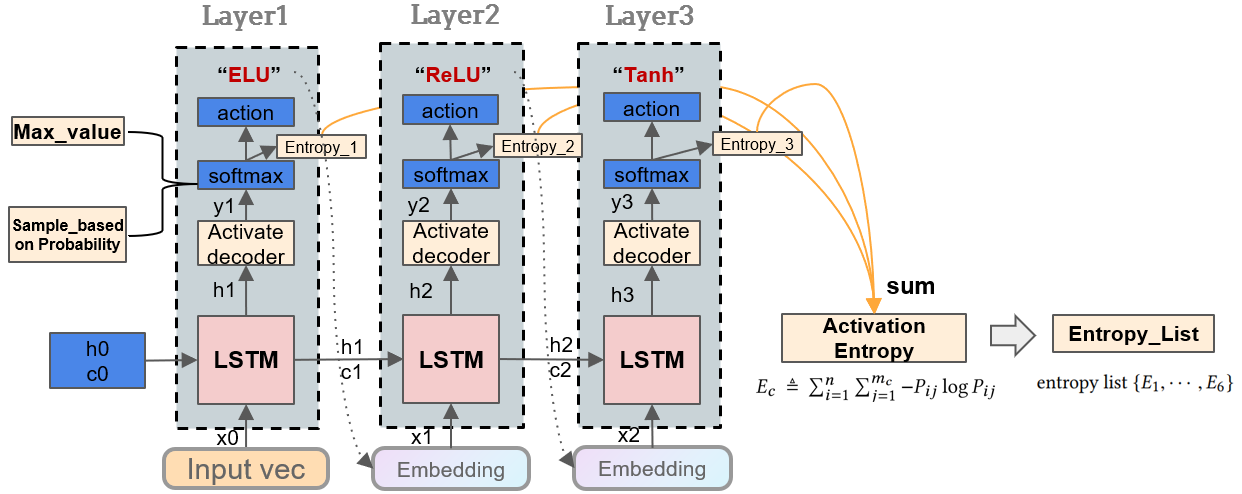
**(1).six RNN encoders:**

**1.input**



**依次去掉best GNN中每层的相应的action,按照上图形成6组输入**

**2.encoding**



**(2).action guider:**

**1.给定搜索规模size参数**

**2.从Entropy\_List中基于Entropy大小选取size个classes**

**3.选择方式1.basd on value,2. based on sample of probability**

**例子：[1,2,3,4,5,6], size = 2**

**select 5 ,6 对应的class(based on value)**

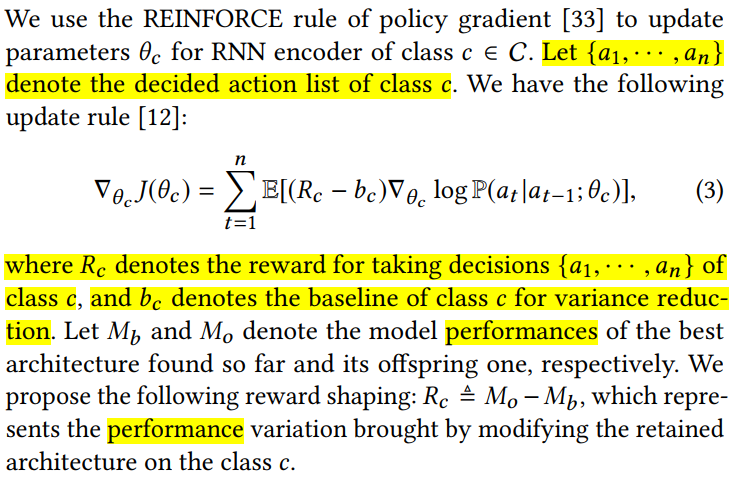
**select 5, 4 对应的class(based on sample of probability)**

**(3).modification:**

**基于选择出来的classes对best GNN相应action进行replace获得child GNN验证其效果得到reward**

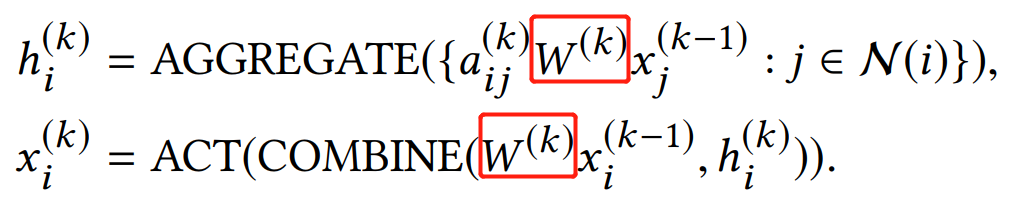
**2.3.trainer:**

**对guider选择出的class对应的RNN encoder使用policy gradient优化**

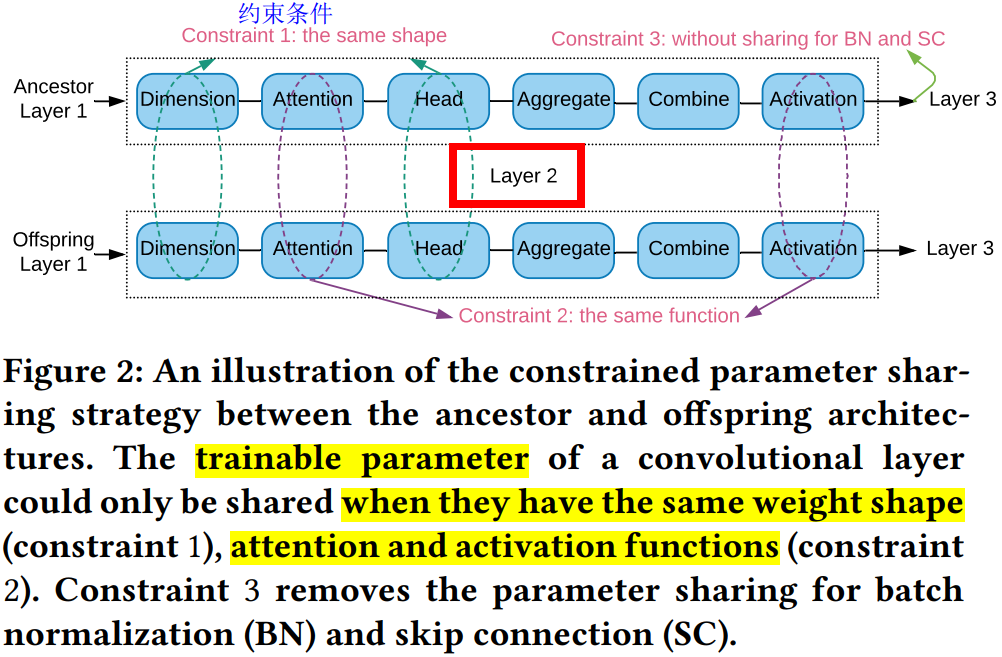


**3.GNN训练参数共享**

**(1).对于best gnn 与child gnn,当对应层dim,head参数相同，Attention，Activate方式一致，则在child gnn中共享best gnn中W权重参数**



**(2).共享参数child gnn cite任务训练轮数20epoch，PPI任务5epoch,不共享参数child gnn cite任务训练200 epoch，PPI任务20epoch**



[**Evaluation**](javascript:;)**:**

**G实验1**



**2层GNN结构**

**accuracy值**

**实验2**



**Micro-F1**

**3层GNN结构**

**Advantage/Disadvantage:**

Advantage:

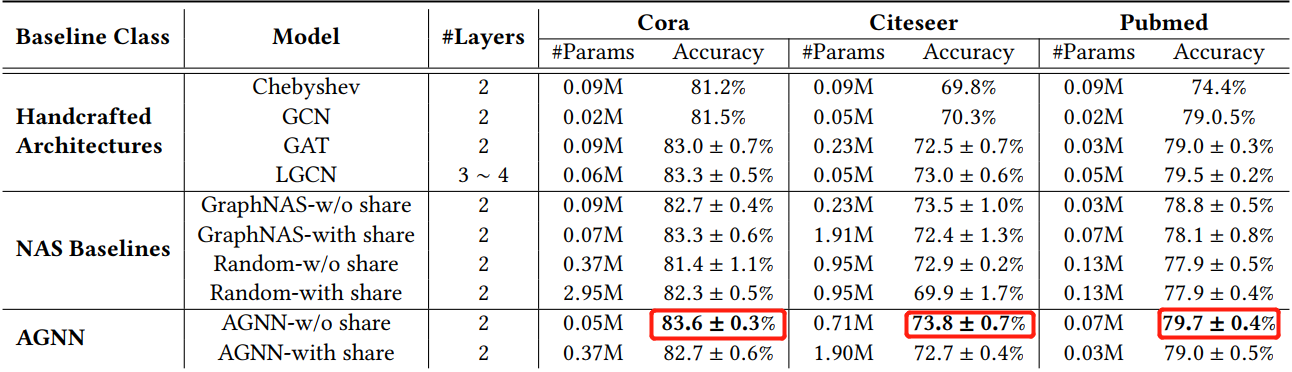
**GNN NAS**

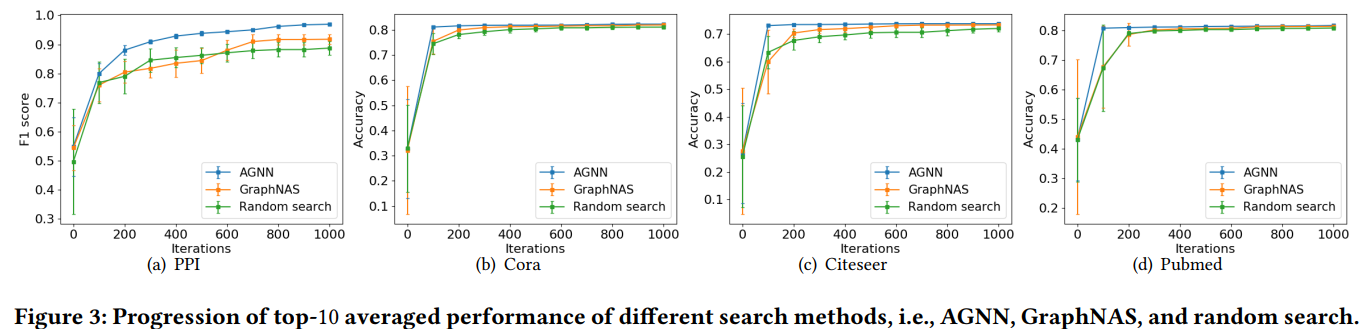
Disadvantage:

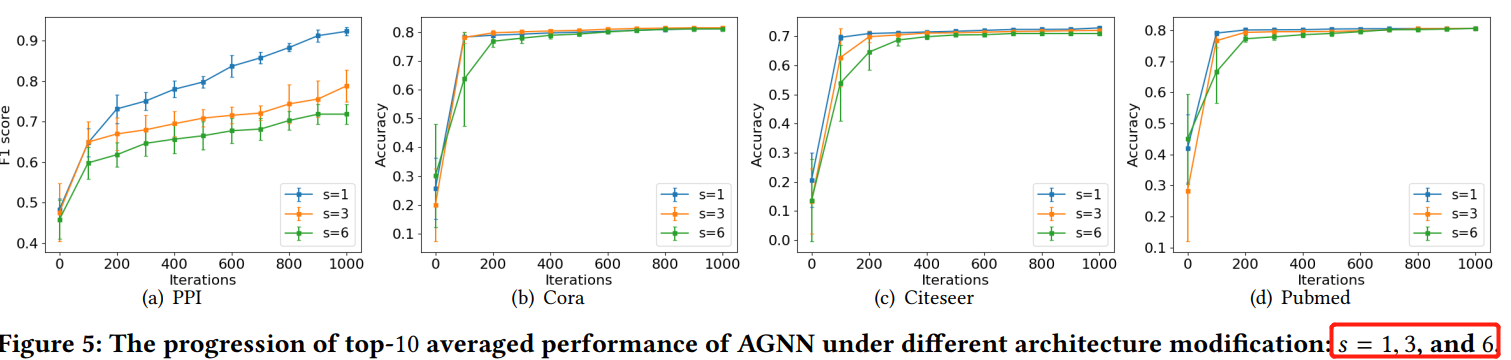
**GNN NAS**

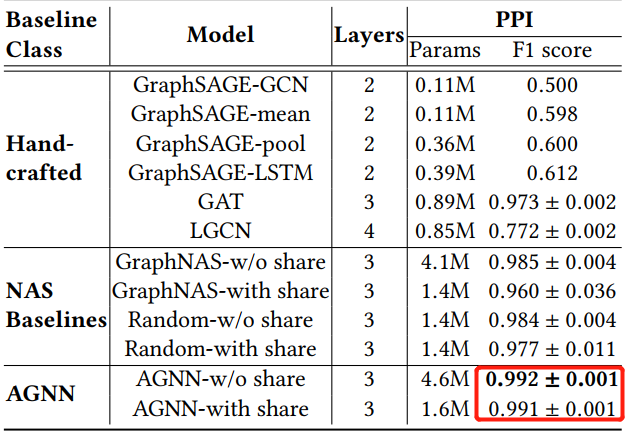
**Result:**

**1.效果对比**

**2.效率对比**







**Code:**

**没有代码**

**Future Work:**

**Title:**

**Simplifying Architecture Search for Graph Neural Network**

**Basic:**

**GNN NAS**

**Introduction:**

**方案：**

**1.GNN Search Space(3 classes)**

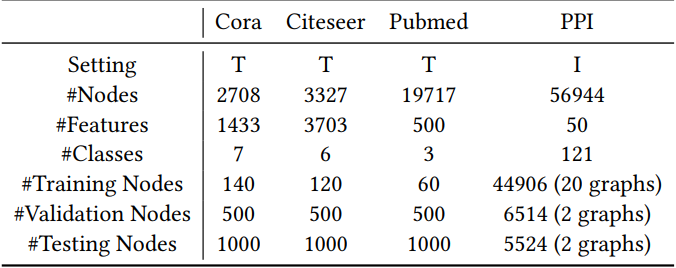
**2.Controller is GrapyNas Controller (1 encoder 1 LSTM 3 decoders)**

**实验验证:**

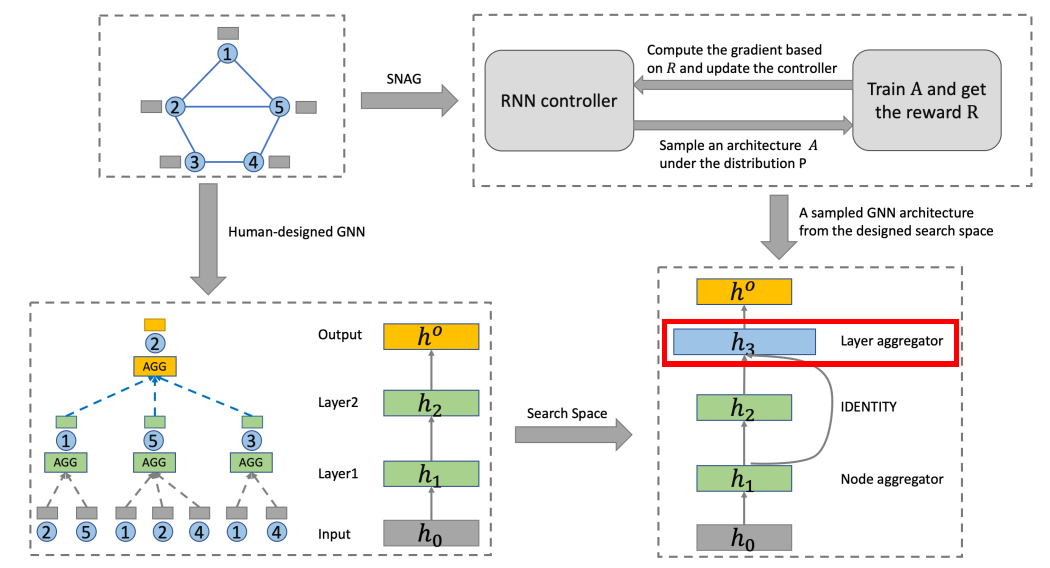
**(1).半监督：**

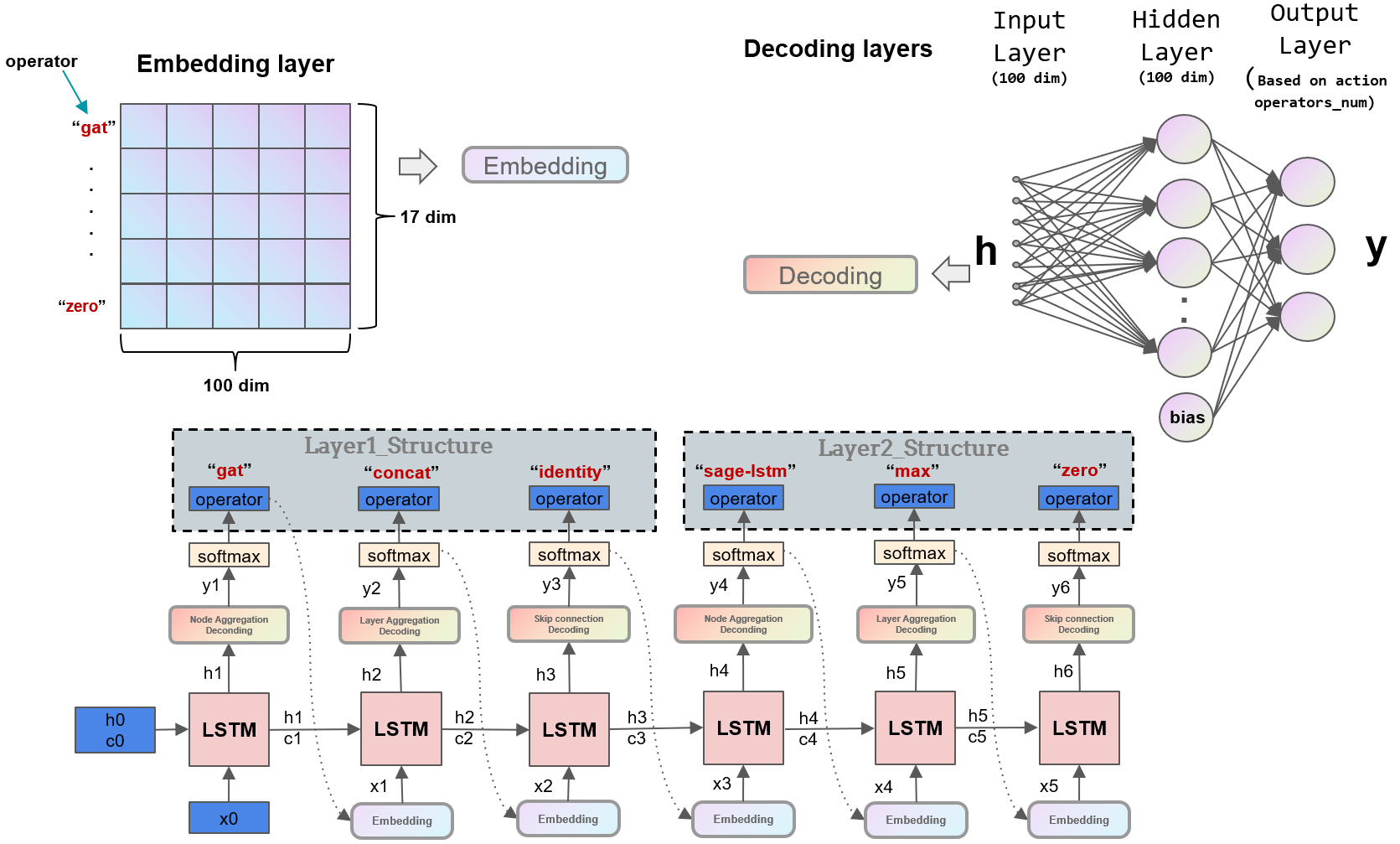
**(2).有监督：**

**Data:**



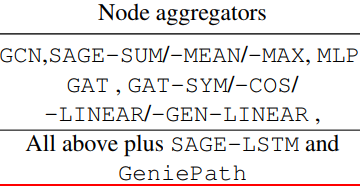
**Method:**





**1.搜索空间定义:**

**(1).Node aggregations action**



**(2)Layer aggregators action**



**(3)Skip Connection action**



**2.类似GraphNAS一样设计Controller**

**3.基于强化学习机制训练Controller**

**4.使用Controller筛选GNN结构在数据上验证**

[**Evaluation**](javascript:;)**:**

**实验1**



**2层GNN结构**

**accuracy值**

**实验2**



**Micro-F1**

**3层GNN结构**

**Advantage/Disadvantage:**

Advantage:

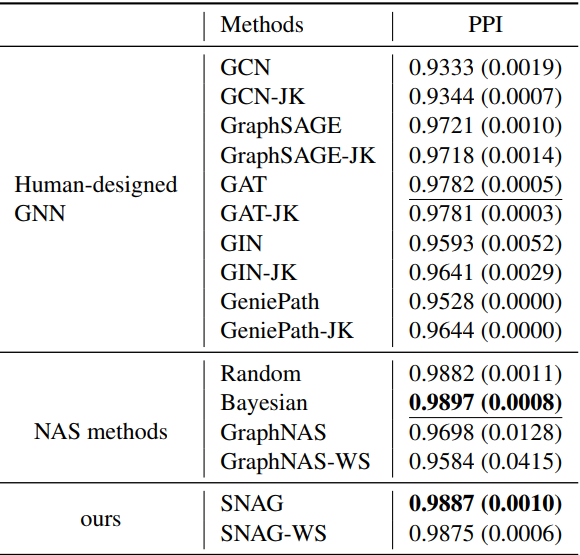
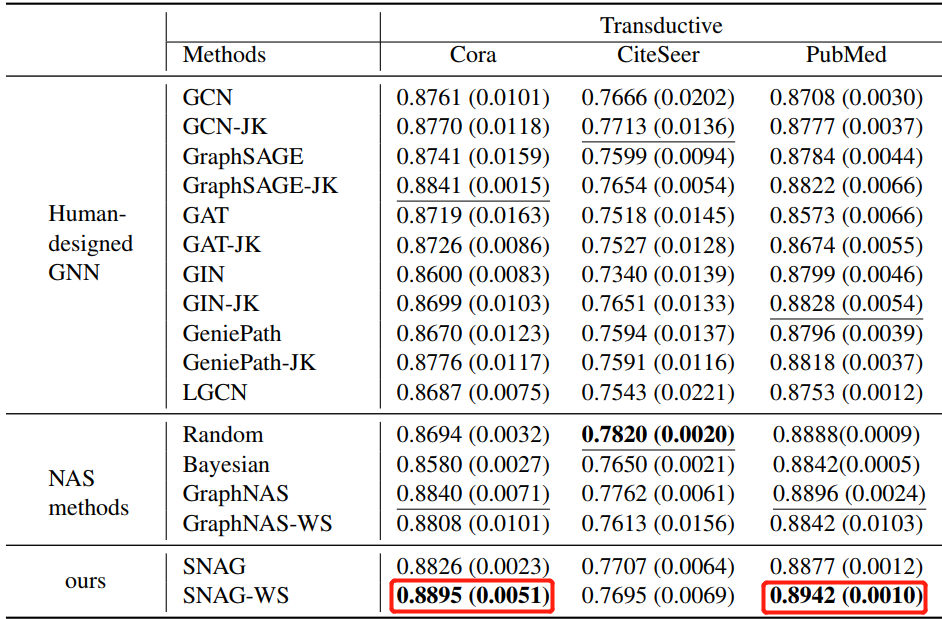
**x**

Disadvantage:

**x**

**Result:**

**效果对比:**



**Code:**

**无**

**Future Work:**

**Title:**

Evolutionary Architecture Search for Graph Neural Networks

**Basic:**

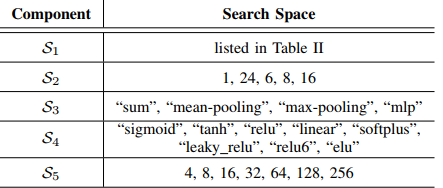
**GNN NAS**

**Introduction:**

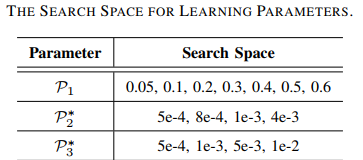
**方案：**

**1.search space**

**(1).Structrue Space**



**(2).Parameter Space**



**2.Embedding GNN Structure and Learing Parameters into intermediate population** **(PS/SP)**



**3.Using GA evolve Learning Parameters from P(k) with the same GNN Structrue to P(k+1) and select the best Learning Parameter**

**4.Using GA evolve GNN structrue from S(k) with the same best Learning parameters to S(k+1) to select the best GNN Structrue**

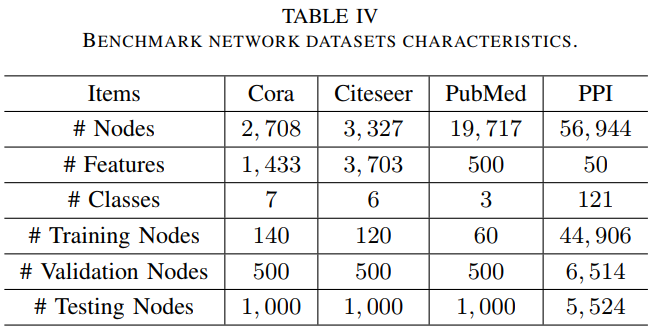
**5.Using the GNN task Accuracy forming the fitness**

**实验验证:**

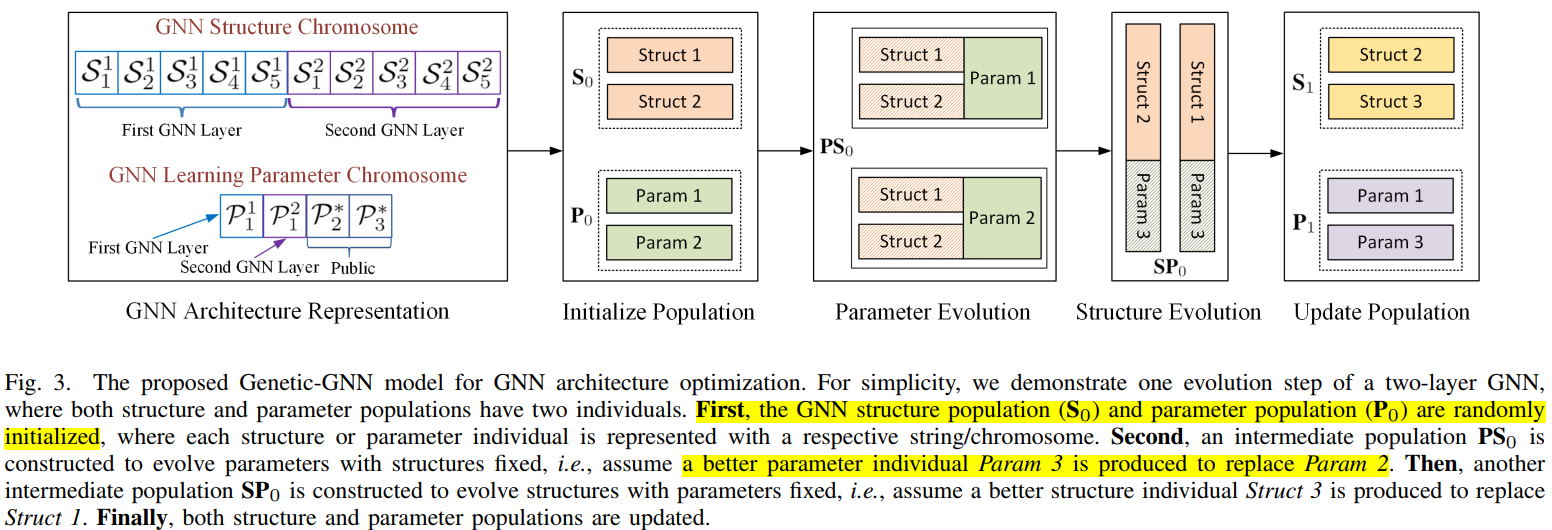
**(1).半监督：**

**(2).有监督：**

**Data:**



**Method:**



**1.遗传算法简介：**

(1).组成：

* 编码—>创造染色体，将问题转换化为数学问题
* 个体—>多个个体构建出一个种群
* 适应度函数—>评价个体好坏程度
* 遗传算子：

1.选择：

从前代种群中选择**多对(偶数个)，较优**个体，一对较优个体称为一对父母

选择的具体操作：

(1).对种群中个体按照适应度从小到大排序

(2).选择方案（列举两种）：

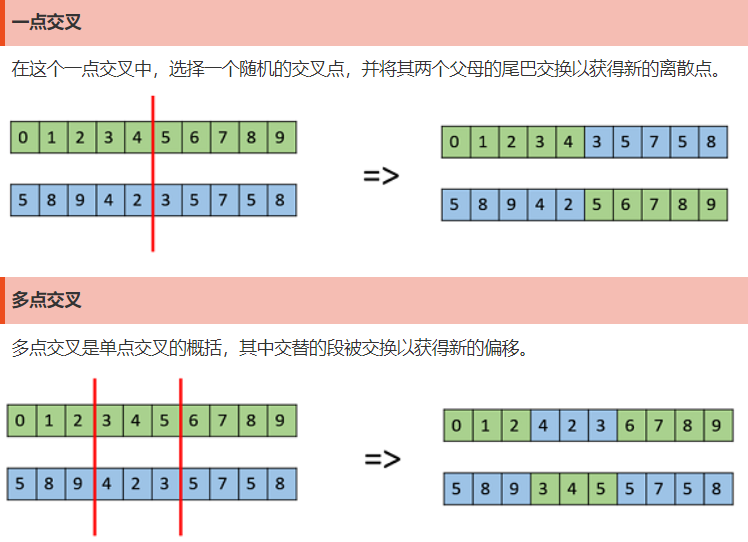
a.轮盘选择法选择法，个体被选中的概率与其适应度成正比**（不是最优个体就一定能被选择到）**

b.精英选择法，按适应度排序选择最优个体

交叉：

(1).两个待交叉的不同染色体(父母)，根据交叉概率(cross\_rate)，按照某种方式交换其部分基因

(2).交叉方案：单点交换，多点交叉等

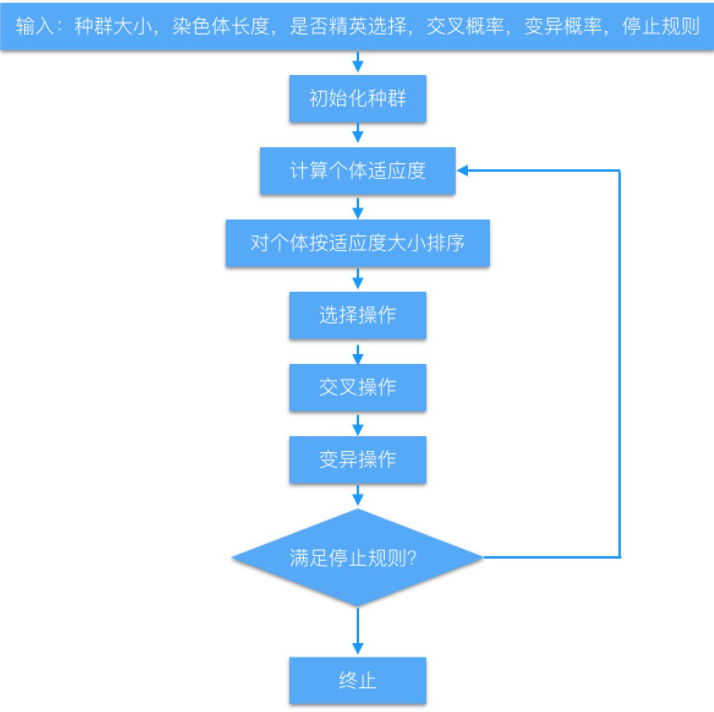


变异：

(1).染色体按照变异概率(mutate\_rate)进行染色体的变异

(2).变异方案：交换突变，争夺突变等

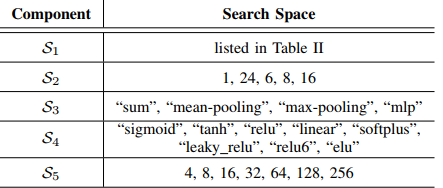




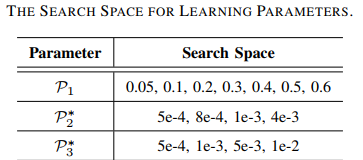
**2.本论文方案分析：**

**(1)编码：**

**structrue space**



**parameter space**



**Embedding GNN structure and Learing Parameters into a String**

**Structrue embedding:**



**Learning parameter embedding:**





**->只对第一层GNN使用**



**(3).种群构建：**

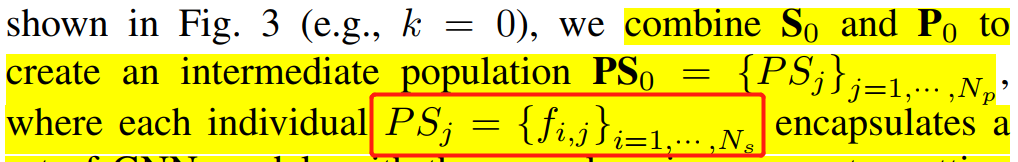
随机初始化20个Structure Embedding , 随机初始化6个 Learning Parameter Embedding;

**(4).中间种群构建：**



**(4.1).对于进化Learning parameter时，构建中间种群PS:**

将20个 Structure Embedding与 6个Learning Parameter Embedding组合, 得到种群数量20 \* 6 = 120;



**(4.2).对于进化GNN structure时，构建中间种群SP:**

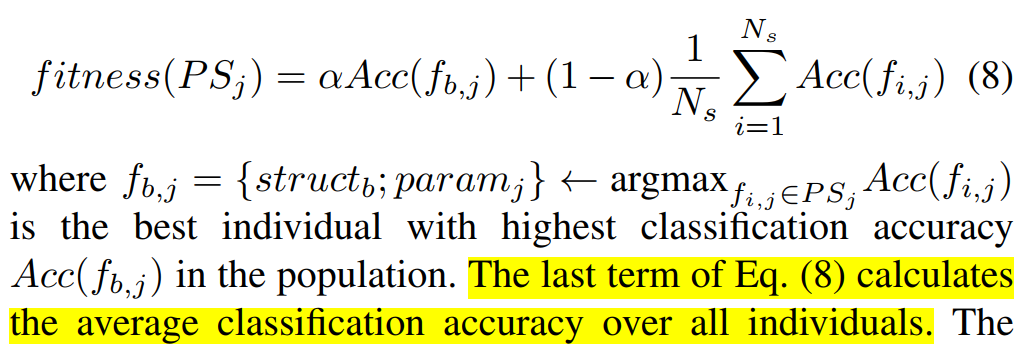
将20个Structure Embedding与 Best Learning Parameter Embedding组合，得到种群数量20\*1

**(5).PS/SP中间种群计算适应度**

**(5.1).对Learning parameter计算中间种群PS适应度:**

**j代表一个Learning parameter**

**fitness(PSj)表示相同Learning parameter j下由所有Structrue组成的fitness**



**(5.2)对于GNN Structure计算中间种群SP适应度:**



**(6).选择方式：轮盘/精英（文章未给出），**

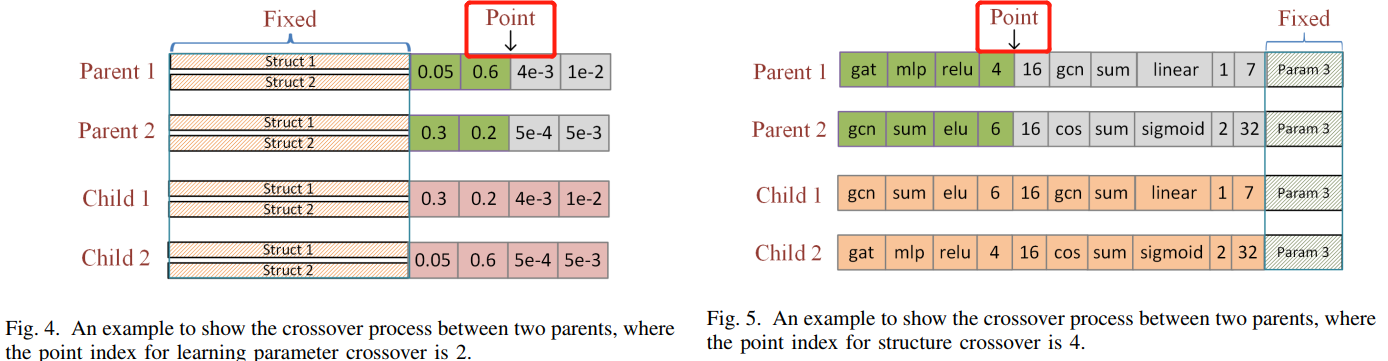
**(6.1).对于进化Learning Parameter**

**从种群120 个中间种群PS中，选择4个parennts，得到2个childs**

**(6.2).对于进化Structrue**

**从20个中间种群SP中，选择10个parents，得到5个childs(文章给出是4个childs)**

**(7).交叉方式：单点交叉，文章未给出交叉概率cross\_rate**



**(7).变异方式：文章未给出，猜测是单点在相应选择空间中变异，变异概率0.02**

**(8).基于PS/SP fitness function分别评估2个PS childs适应度与5个 SP child的适应度**

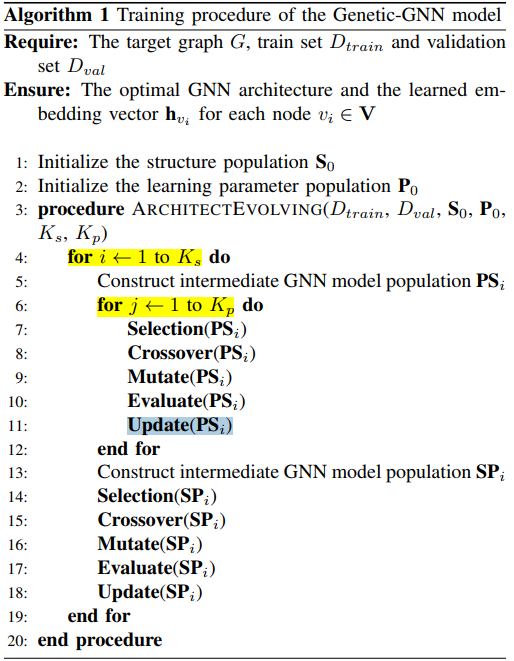
**(9).基于child适应度更新中间种群PS(k)\SP(k),如果child fitness好于中间种群的最差个体fitness,则使用child 替换掉中间种群中最差个体，保持种群规模不变**

**(10).下一轮迭代从更新后的中间种群PS(k+1)\SP(k+1)开始**

**(11).一个epoch的停止条件，GNN structrue进化50代，其中在每代进化中，Learning parameter 进化10代**

**(12)基于适应度，最终从20个SP中间种群中选出最佳Structure + Learning Parameter；**

**伪代码：**



[**Evaluation**](javascript:;)**:**

**实验1**



**2层GNN结构**

**accuracy值**

**实验2**



**Micro-F1**

**3层GNN结构**

**Advantage/Disadvantage:**

Advantage:

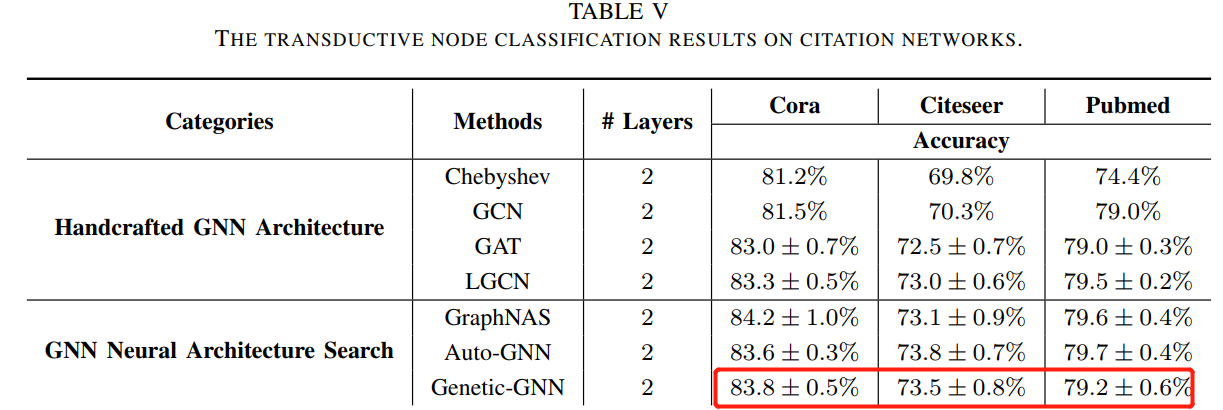
**x**

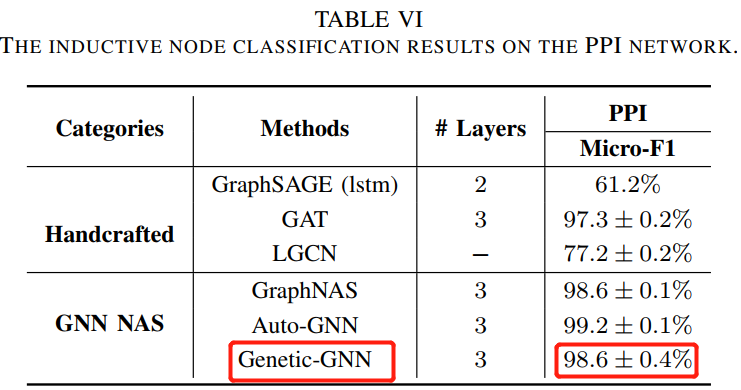
Disadvantage:

**x**

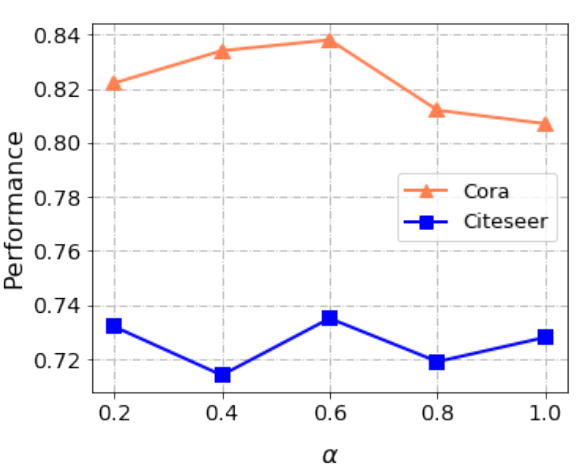
**Result:**

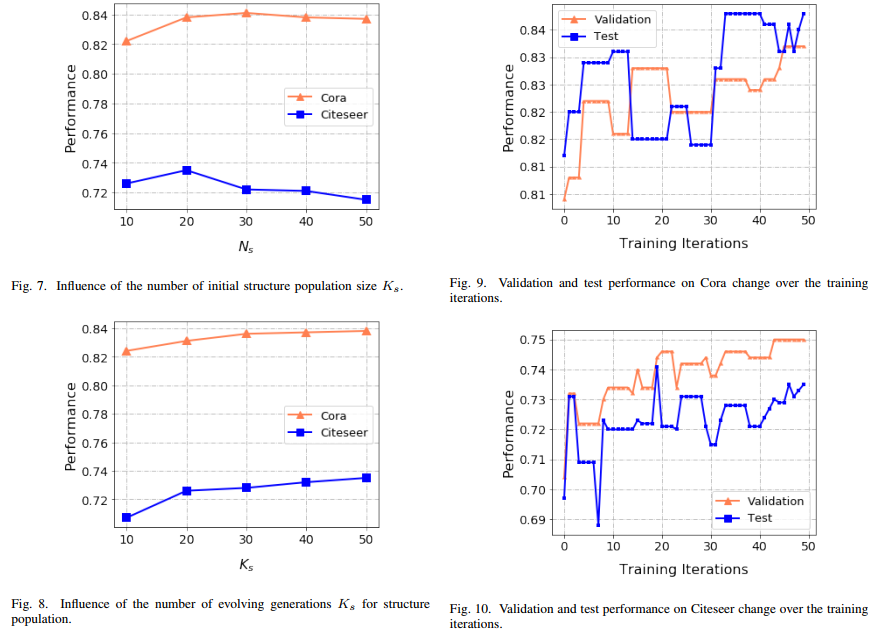
**效果验证**





**参数分析**





**Code:**

**无**

**Future Work:**

**参考：https://www.zhihu.com/question/23293449**