Policy-GNN: Aggregation Optimizationfor Graph Neural Networks（2020 KDD）文献调研

Kwei-Herng Lai, Daochen Zha, Kaixiong Zhou, Xia Hu

{khlai037,daochen.zha,zkxiong,xiahu}@tamu.edu

Department of Computer Science and Engineering, Texas A&M University

**目录：**

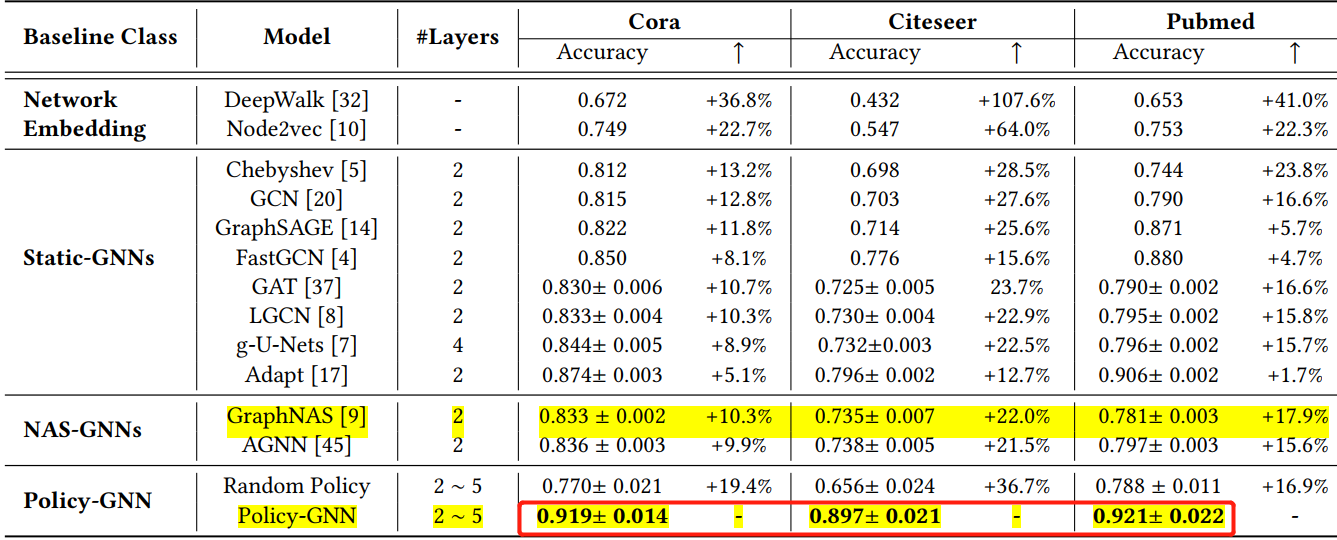
**1.policy\_gnn在cite任务的性能表现**

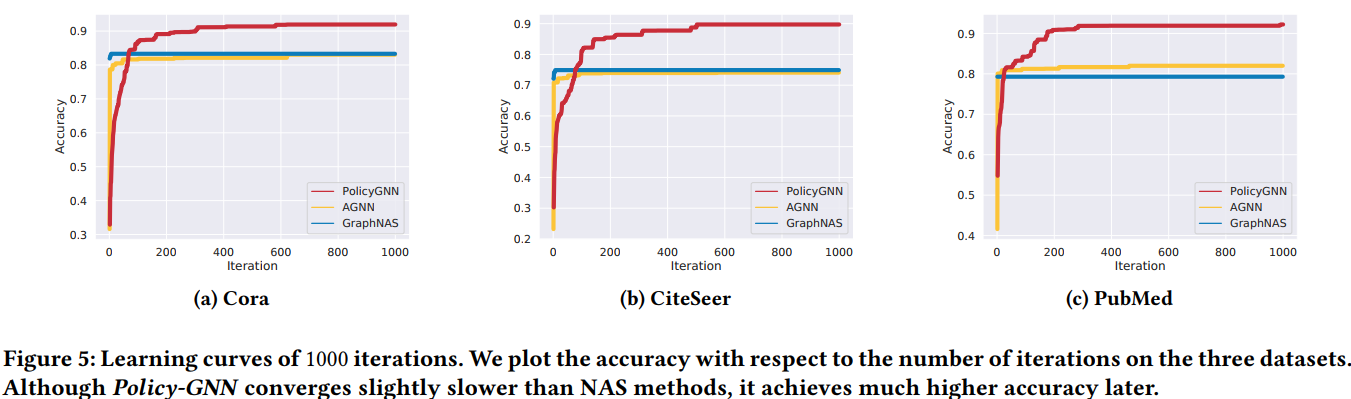
**2.Policy\_gnn方案分析**

**3.基于阅读文献的思考与小结**

**4.笔记**

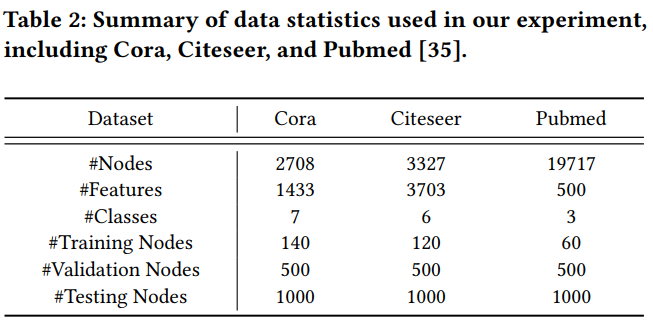
**1.policy\_gnn在cite任务的性能表现**





**2.Policy\_gnn方案分析**

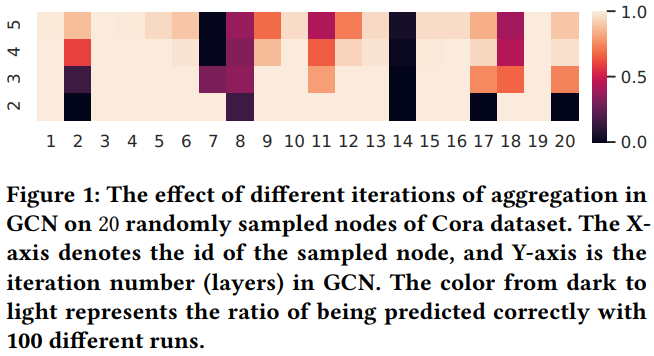
**数据集：**



**2.1.核心假设与验证**

**假设：在图结构中，不同节点应该使用不同阶数的邻居节点进行信息聚合，经过合适阶数邻居节点聚合后的节点具备更好的表达，参与下游任务能获取更优秀的性能；**

**假设验证：**

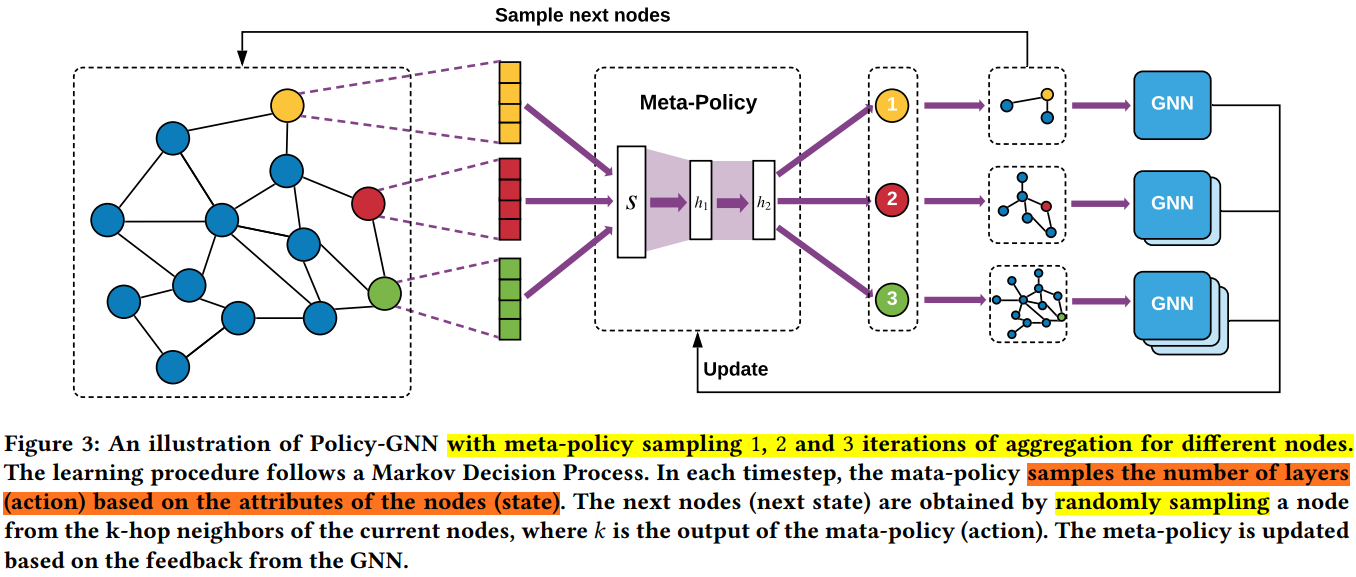


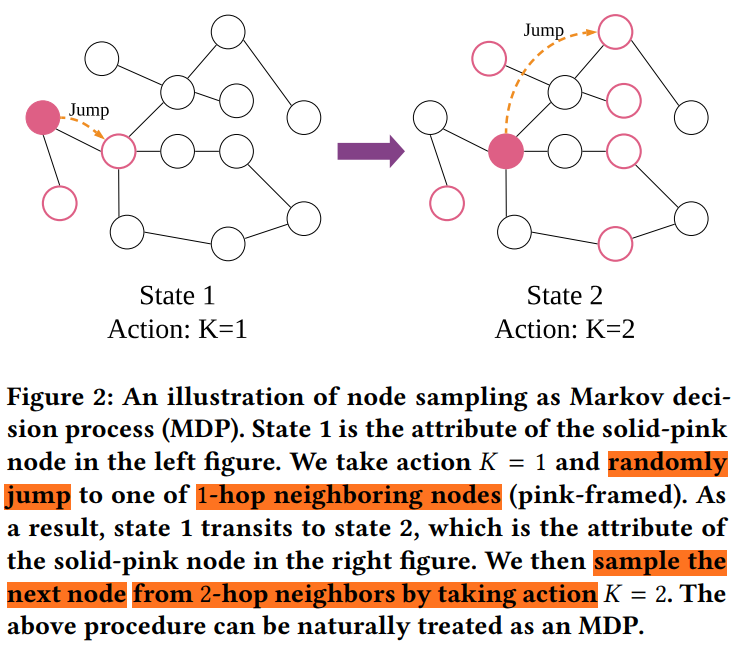
**2.2总体建模思路**

**核心模型：基于频域卷积的基础GCN架构，基于Q-Learning机制的深度强化学习**

**核心问题：怎样自适应地对一个已知的图数据集的每一个节点选择一个合适的阶数进行信息聚合？**

**核心方法：使用基于值的深度强化学习——（DQN网络），学习到一个决策模型policy model,由学习后决策模型policy model来决定每个被用于训练节点应该选择的聚合信息的阶数。**





**补充说明**

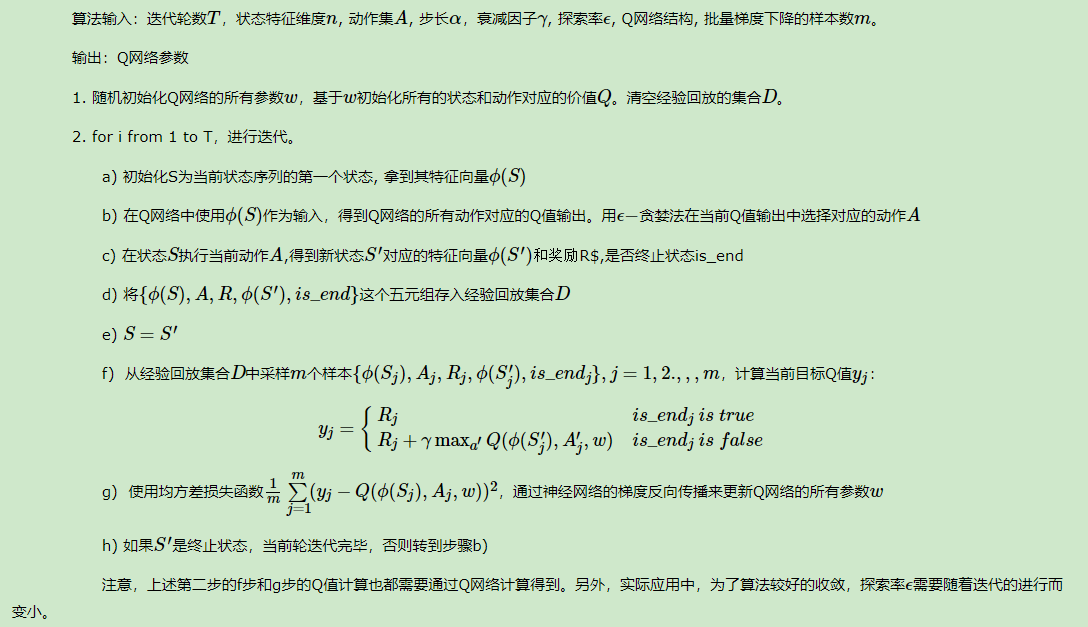
**(1).Meta-Policy是一个5层，结构为[32,64,128,64,32]的感知机，输入维度为节点的特征维度，输出维度为聚合的最大阶数，action = 5，最多聚合5阶邻居节点**

**(2).强化学习中reward计算：**

**(3)基于epsilon-greedy-policy**

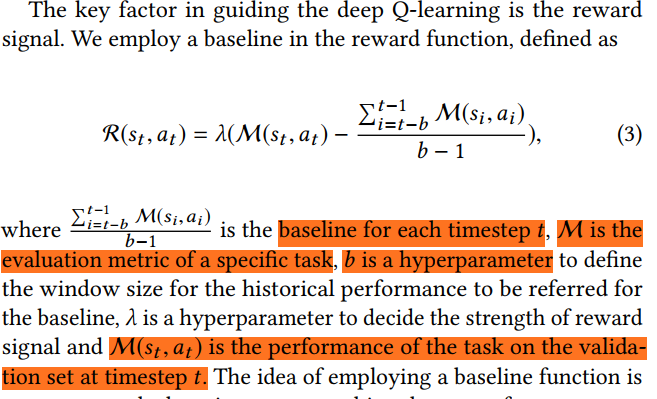
**3.1具有epsilon的概率随机选择本节点聚合的阶数和（1-epsilon）的概率择经过policy model确定的最大Q值的阶数进行聚合，epsilon从1随着迭代逐渐减小至0.2**

**(4).policy model训练流程：**

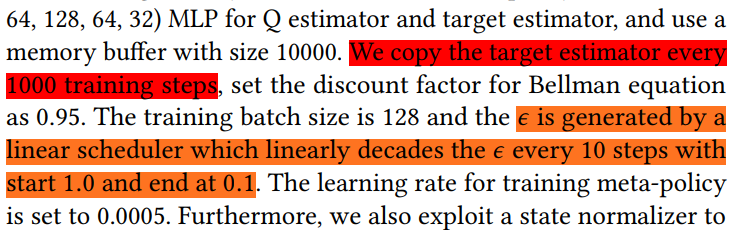


**文章Human-level control through deep reinforcement learning给出了验证**

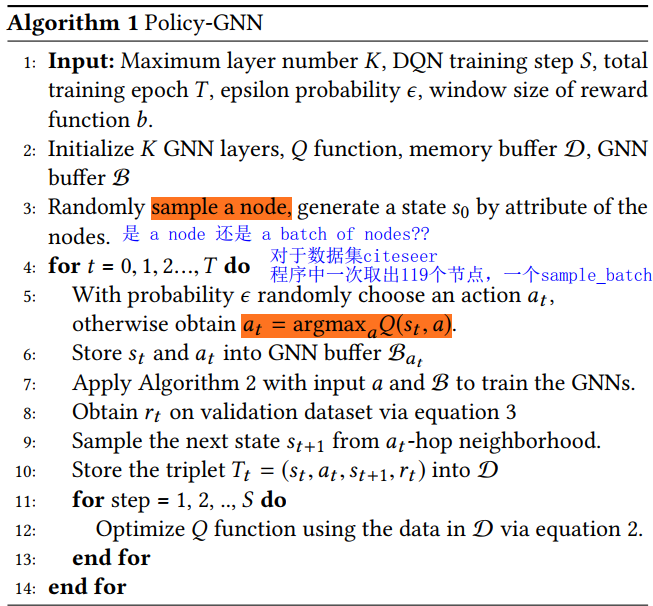
**reward计算：**

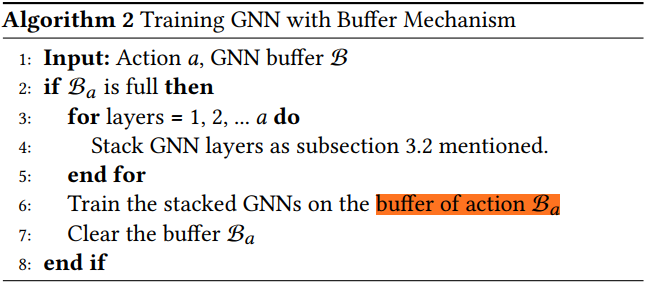


**target model设计**

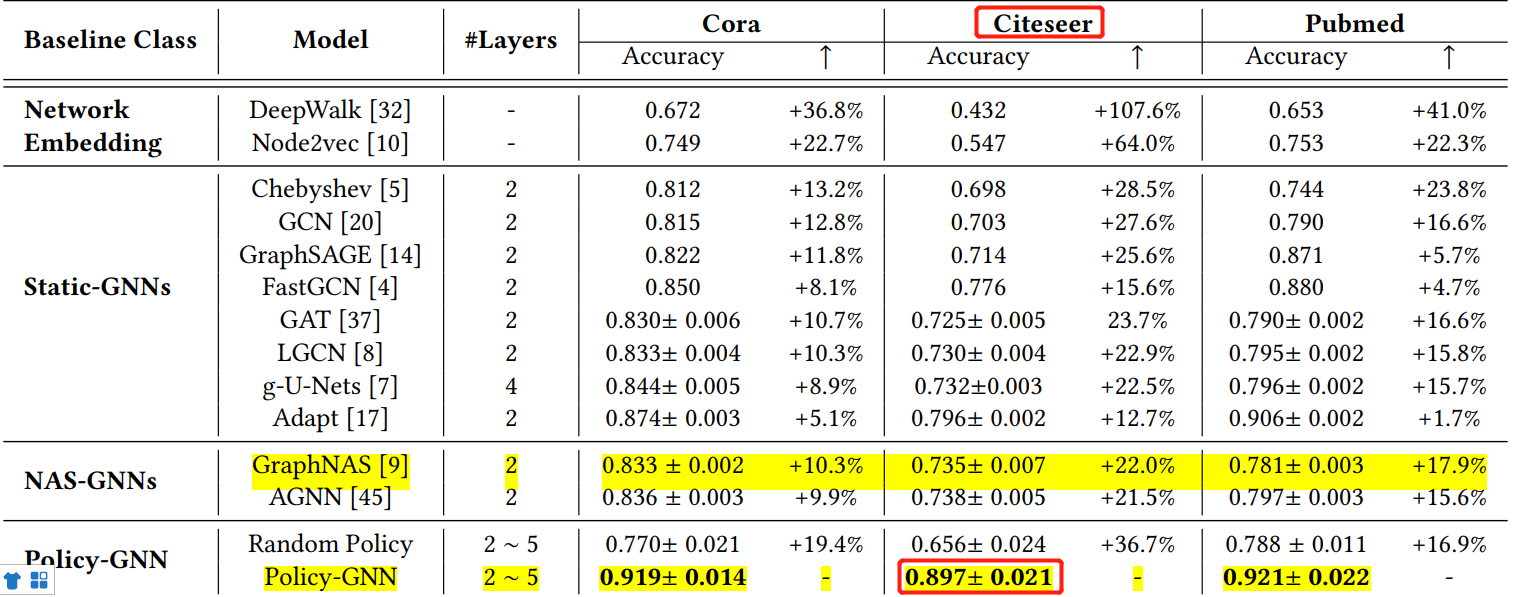


**(5).使用训练的好的policy model在数据集中确定每个训练节点应该聚合的阶数，使用GCN进行实现图卷积操作，接下游节点分类任务**





**实验验证：**

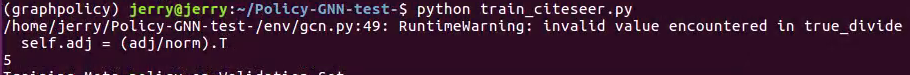






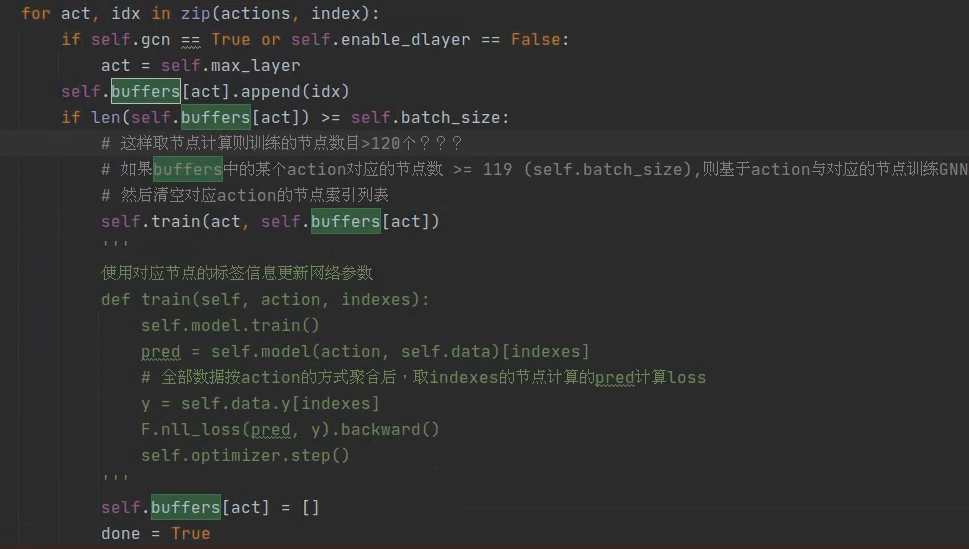


**程序warning**

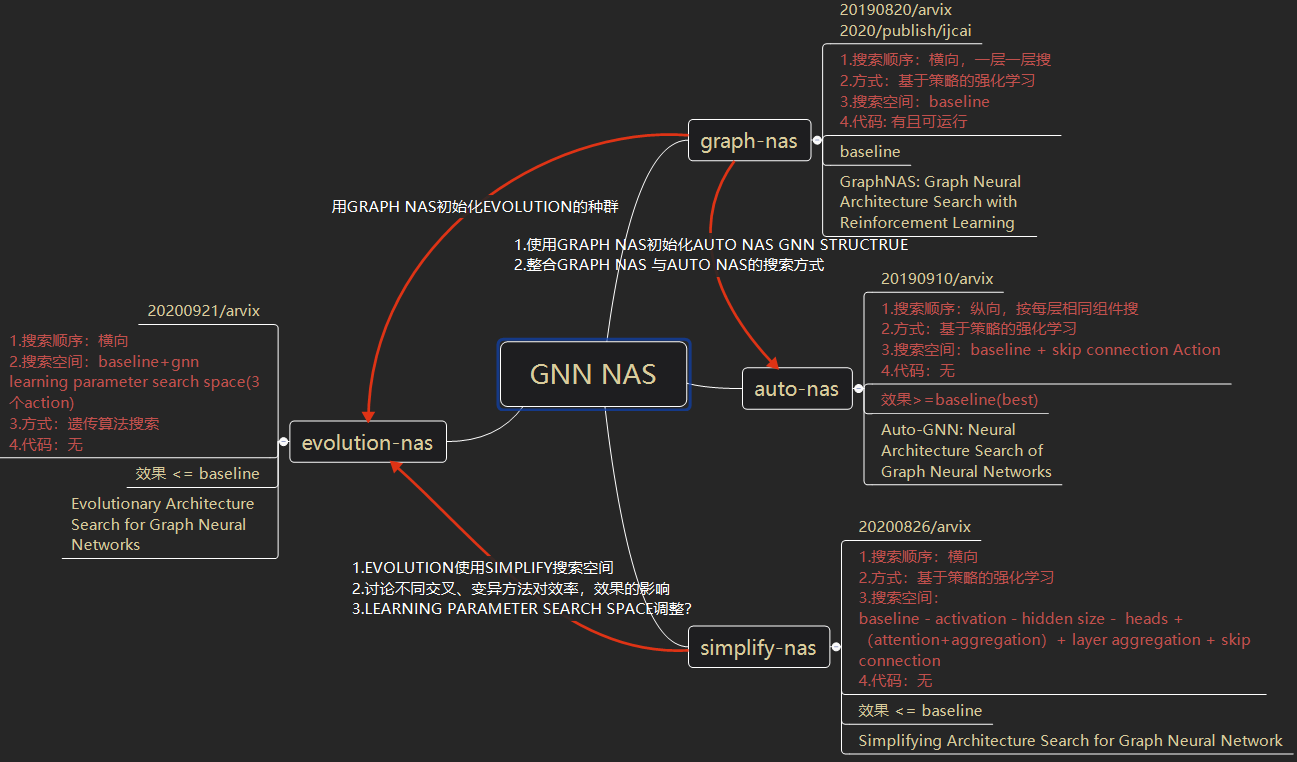


**效果分析：**

**训练policy model时，使用了超过训练集多得多得数据训练policy model，policy model对数据集全局的感知性更强，更能在小规模的训练集中找到每个节点应该聚合邻居的阶数，而其他模型只在训练集的节点中感知整个网络的信息。**

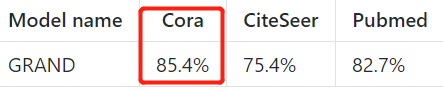


**3.基于阅读文献的思考与小结**



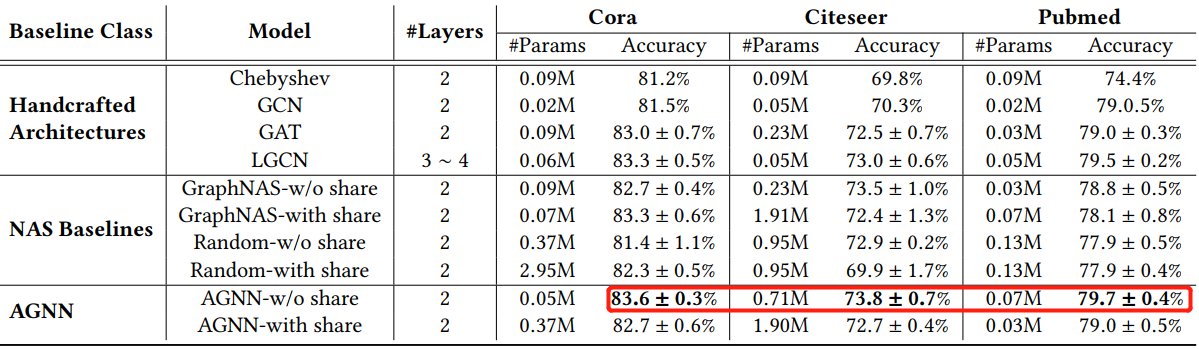
**Graph Random Neural Networks**Wenzheng Feng1∗, Jie Zhang2∗‡, Yuxiao Dong3, Yu Han1, Huanbo Luan1, Qian Xu2, Qiang Yang2,  
Evgeny Kharlamov4, Jie Tang1§  
1 Department of Computer Science and Technology, Tsinghua University  
2WeBank Co., Ltd 3Microsoft Research 4Bosch Center for Artificial Intelligence  
fwz17@mails.tsinghua.edu.cn, {zhangjie.exe, ericdongyx, luanhuanbo}@gmail.com  
yuhanthu@126.com, {qianxu, qiangyang}@webank.com, evgeny.kharlamov@de.bosch.com, [jietang@tsinghua.edu.cn](mailto:jietang@tsinghua.edu.cn)

grand结果：（arvix）

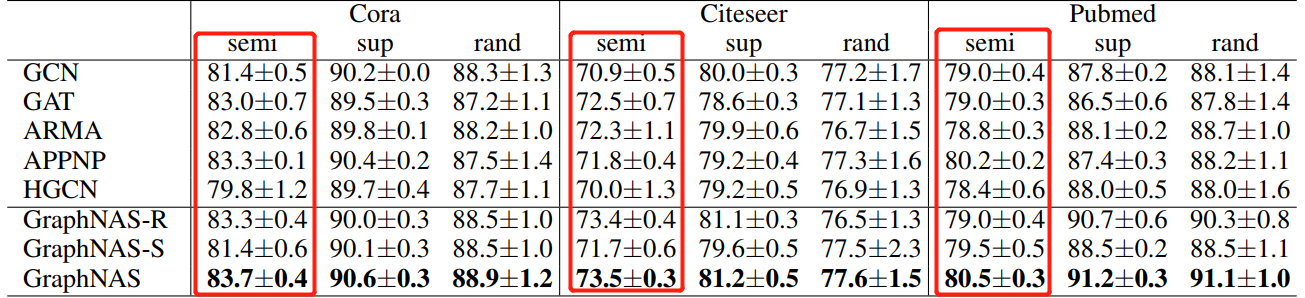




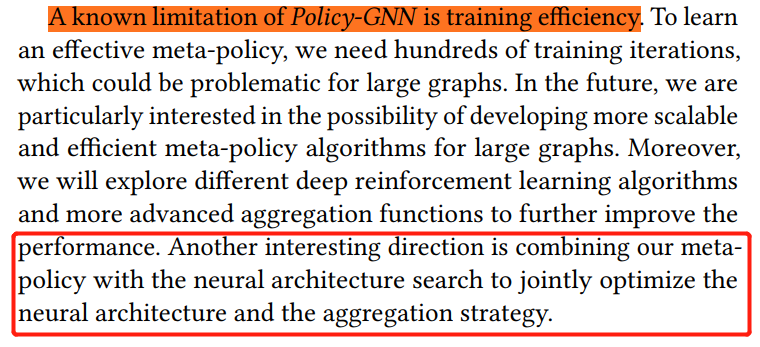
auto-gnn结果(arvix)：



graphnas结果(IJCAI):



**笔记：**

**未来的改进方向**

**1.效率，使用更高效更有效的深度强化学习搜索**

**2.每个节点聚合阶数搜索与GNN结构搜索**

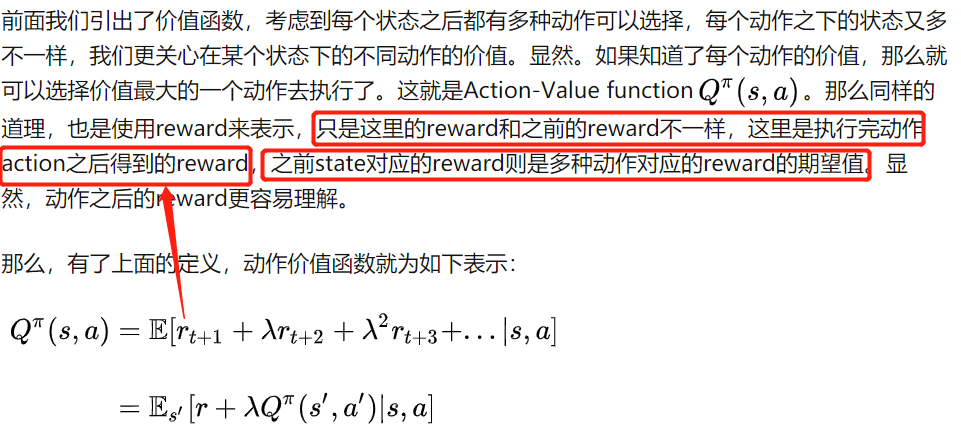
**采取行动的策略方法：epsilion-greedy policy？**

**采样方法:**



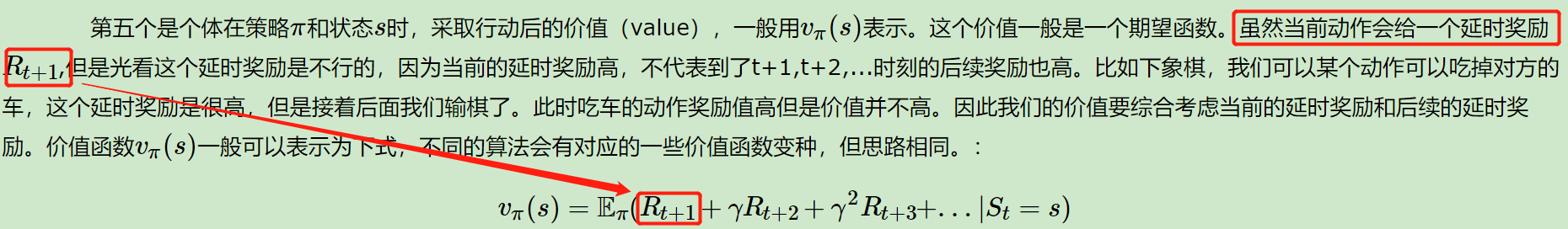
**action-value function，及其reward 定义以下state-value function reward定义有问题：**

**定义:在状态s下采取策略pi选取动作a的价值**

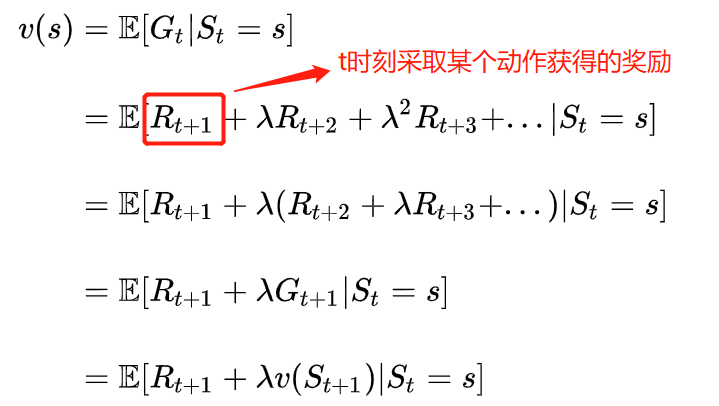




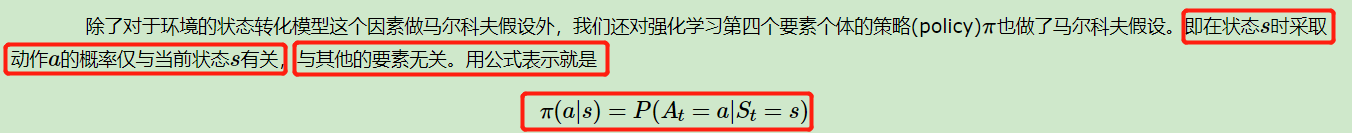
**以下state-value function定义reward正确**



**state-value function，及其reward定义：**

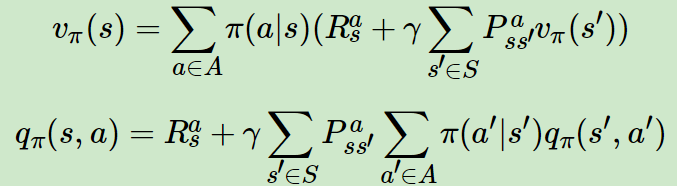


**策略定义：**

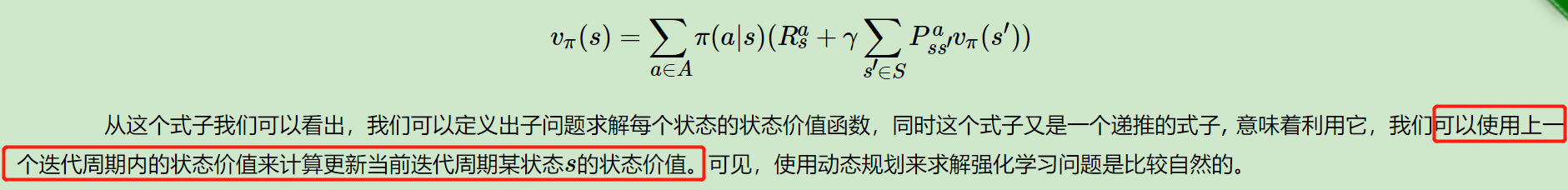




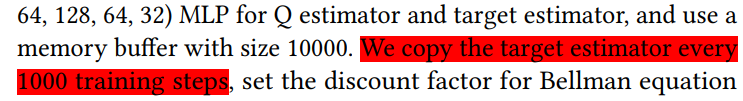
**状态价值函数与动作价值函数递推公式：**

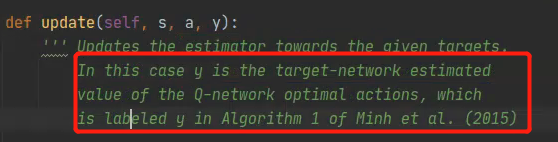


**状态价值函数的更新：**



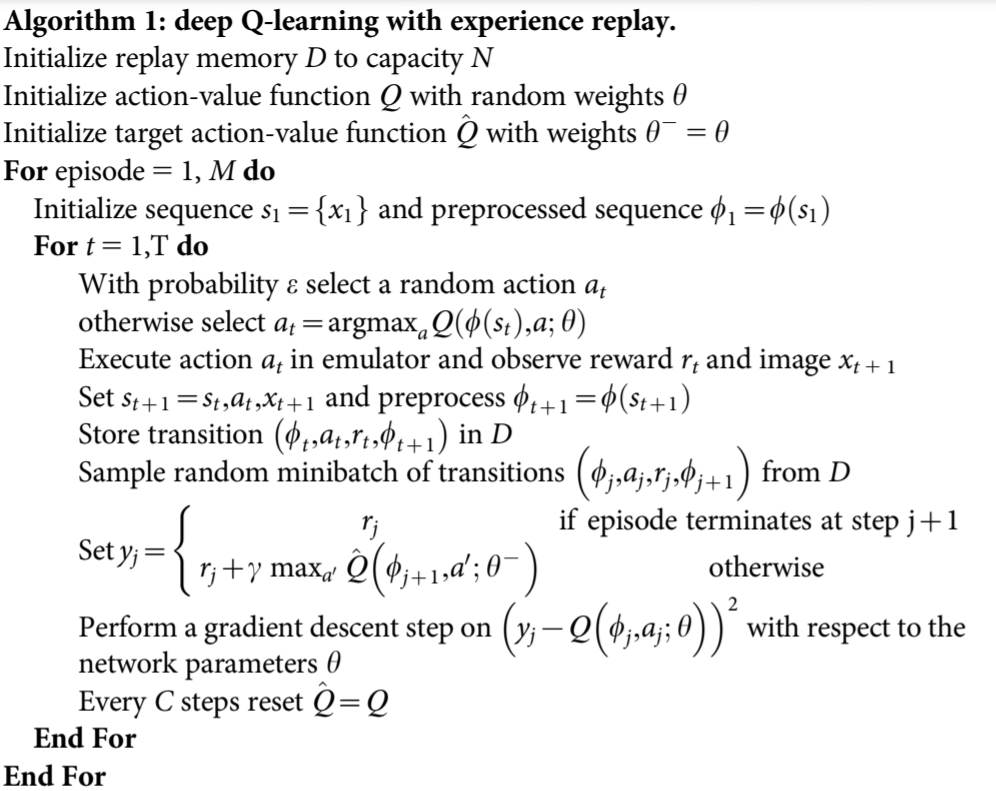
**Q\_estimator与target\_estimator的关系**





**计算y的算法**

**Human-level control through deep reinforcement learning**



**每次sample 节点的 batch\_size数=119**

**一次取119个节点送入q\_net**

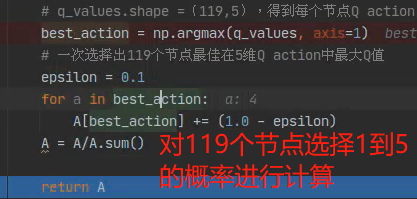
**对于citeseer数据集只使用前120个节点的标签数据进行训练**



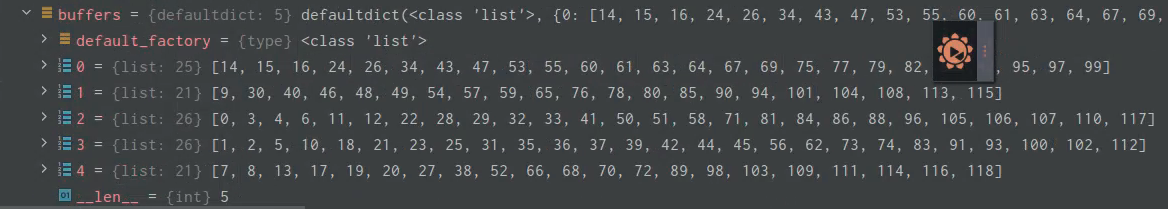


**对119个节点特征向量做归一化后送入Q\_net中**

**一个batch\_size的nodes使用同一张action 概率表**



**buffers数据结构**

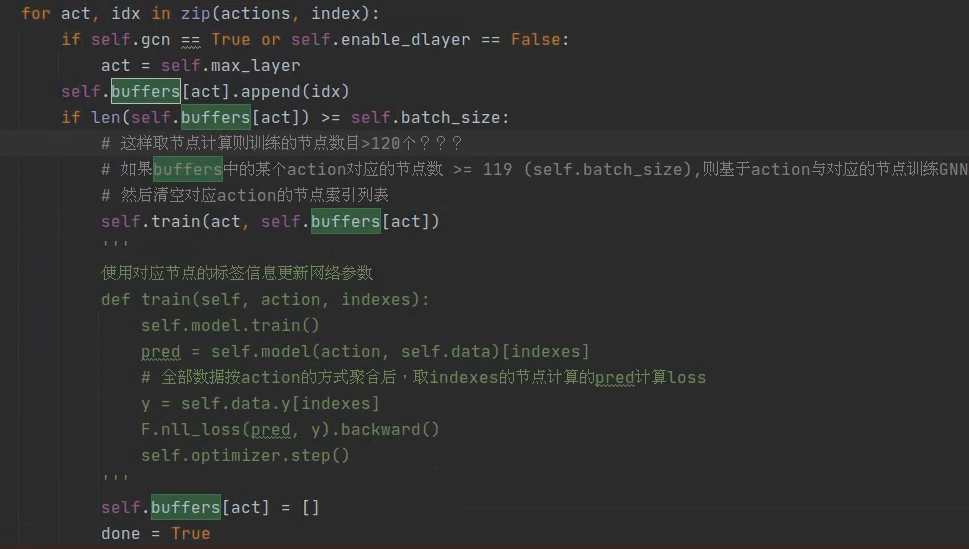


**基于已选节点的action随机在k-hop范围内选择节点，返回节点的索引**

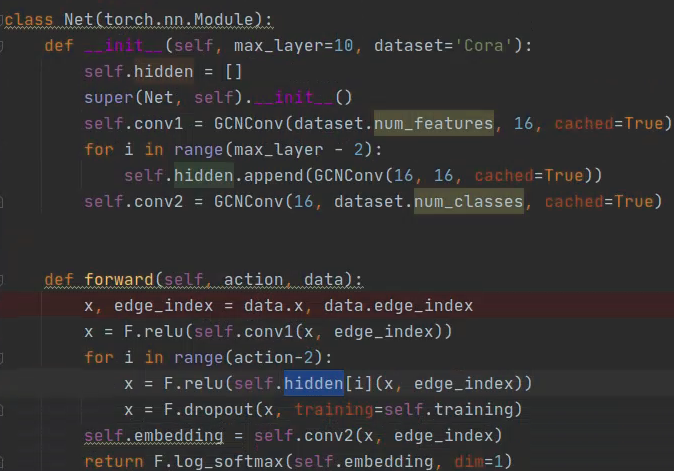




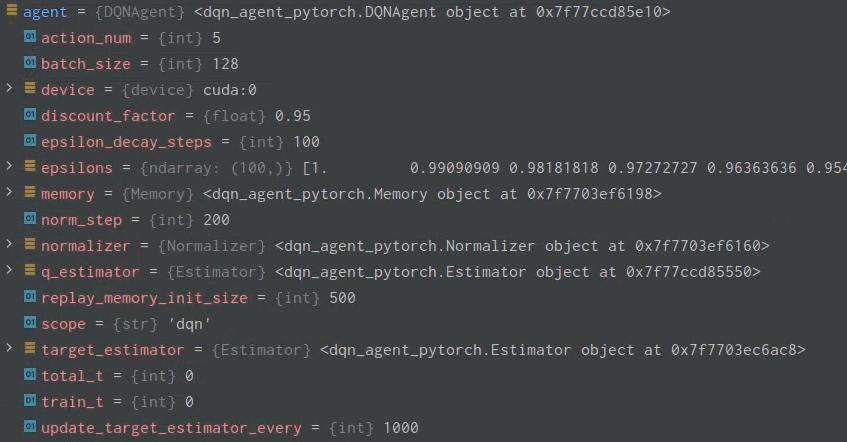
**训练：**

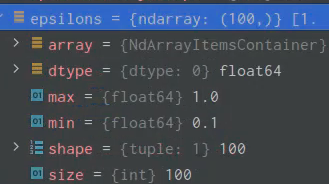


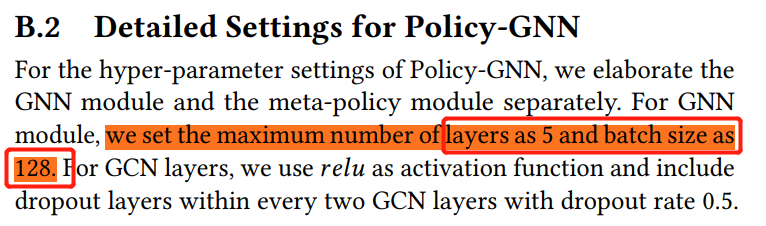


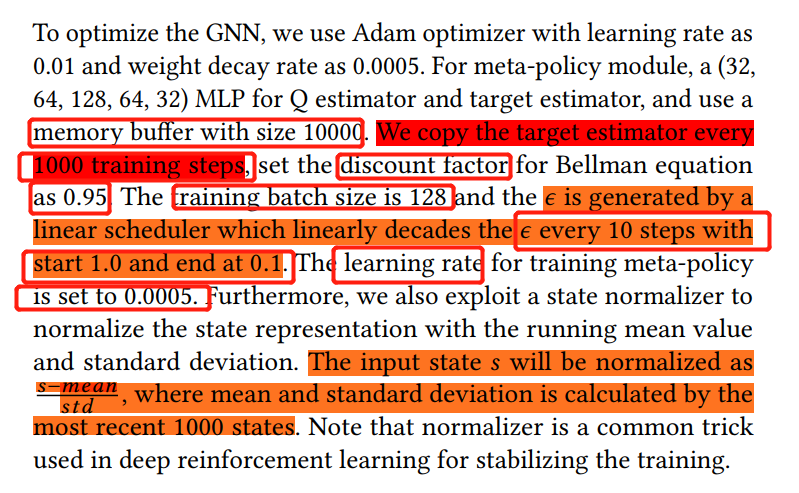


**agent参数设置**





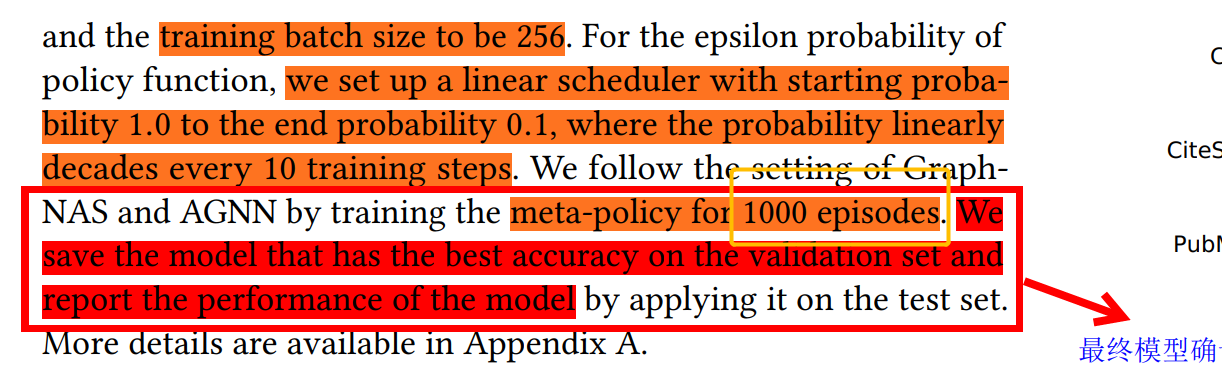




**policy training batch:**

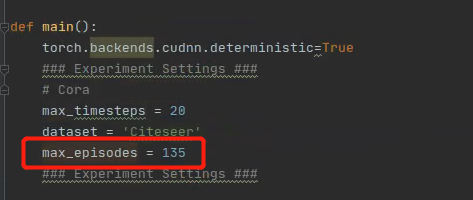


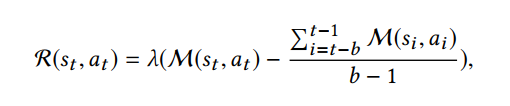
训练policy 1000 epochs



gcn参数：



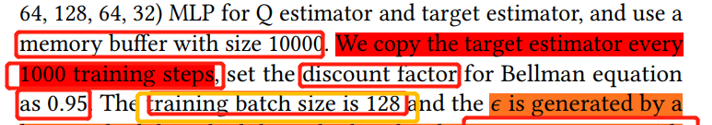




lamda:超参决定reward信号强度，lamda=100 在代码中

矛盾点：





代码用的是128

问题点

~~1.参数设置问题：~~论文中提出的超参在代码中一致

2.~~reward计算问题：~~利用的公式和论文一致

3.~~random模式与Q-learning开关模式：~~代码有开关