**1.BC（介数中心性算法，Betweenness Centrality）**

* 1. **原理**：该算法依据节点在网络信息传播时的负载量判断其关键性。计算网络中任意两个节点间的最短路径，若某个节点被最短路径包含的次数越多，就意味着它在信息传播过程中的负载量越大，该节点也就越关键。比如在一个社交网络信息传播场景中，大量信息传播路径都经过某一节点，那么这个节点在信息传播中就极为重要，其介数中心性较高。
  2. **优势**：考虑到了网络的整体结构属性，在研究网络通信问题时应用广泛。通过分析节点介数中心性，能够找出在信息传播过程中起到关键桥梁作用的节点，对理解网络信息流通机制意义重大。
  3. **局限性**：计算复杂度较高。在大规模复杂网络中，计算所有节点的介数中心性需要耗费大量时间和计算资源，这限制了它在大规模网络场景中的应用。

**2.D（度中心性算法，Degree Centrality）**

* 1. **原理**：以节点的度（即与其相连的邻居节点数量）大小来判断节点的重要程度。在网络传播过程中，大度节点能更广泛地与其他节点接触，所以更有可能传播或接收信息。例如在社交网络中，粉丝众多的账号（大度节点），其发布的信息能直接被大量用户看到，影响力相对较大。
  2. **优势**：计算方便快捷，时间复杂度低。在网络边稠密、结构复杂的情况下，能够快速计算出大度节点，在研究网络脆弱性问题时，对保护网络鲁棒性具有重要意义，因为保护大度节点有助于维持网络的连通性和稳定性。
  3. **局限性**：只关注了节点的邻居数目，没有考虑邻居节点的属性。比如在一个社交网络中，虽然某个节点连接的邻居数量多，但这些邻居可能都是影响力较小的普通用户，该节点实际的影响力可能远不如连接少量但极具影响力邻居的节点，这使得度中心性算法在衡量节点关键性时不够全面。

**3.H-index（H 度中心性算法）**

* 1. **原理**：H-index 源于学术领域衡量学者影响力，在复杂网络中用于评估节点重要性。对网络中每个节点，统计其邻居节点度值并排序，找到满足 “有 h 个邻居节点度值不小于 h” 的最大 h 值，此 h 值就是该节点的 H-index 值 。如某节点邻居节点度值分别为 5、4、3、2、1，满足条件的最大 h 值为 3（3 个邻居节点度值不小于 3），则该节点 H-index 值为 3。
  2. **特点**：综合考虑节点邻居的数量和连接强度，一定程度上弥补度中心性仅考虑邻居数量的不足。但它对网络结构的全局特征反映不够充分，受局部连接影响大，在不同结构网络中稳定性有待提升。

**4.K-Shell（K 壳分解算法）**

* 1. **原理**：依据节点在网络中的位置评估关键程度。通过逐层去除网络中度小于等于 K 的节点，将节点划分到不同层次（K-shell 层），处于网络内层（K 值大的层）的节点被认为更关键。如在交通网络中，核心枢纽节点连接众多线路，处于网络内层，对维持网络连通性至关重要；而边缘节点连接少，处于外层，重要性相对较低。
  2. **特点**：时间复杂度低，适用于大型网络关键节点的快速识别。不过，它对网络结构有要求，在星型网络和 BA 无标度网络中适用性不佳，且难以确定各指标最佳权重因子，不同网络结构下划分结果可能差异大。

**5.CC（接近中心性算法，Closeness Centrality）**

* 1. **原理**：通过描述节点到达网络其他节点的速度快慢来衡量节点在网络中的紧密性和重要性。计算公式为*CC*(*i*)=∑*j*=*i*​*dij*​*n*−1​ ，其中*n*是网络节点总数，*dij*​是节点*i*与节点*j*之间的距离。节点的紧密中心性越大，表明其距离其他节点越近，在网络中的中心位置越明显，传播信息的速度越快、范围越广。
  2. **特点**：能有效衡量信息在网络中的流动性，在信息传播相关研究中有重要价值。但计算复杂度较高，在大规模网络计算中成本较大；而且它假设网络中所有节点传播能力相同，未考虑节点自身属性差异，在实际复杂网络应用中有一定局限性。

模型改进：

1. **混合卷积架构：**

* **交替使用GCN和GAT层，兼顾局部结构捕获和注意力加权**
* **GAT层使用多头注意力机制（4头）增强特征表达能力**

1. **多尺度特征融合：**

* **使用Jumping Knowledge Network聚合不同层级的特征**
* **保留各层输出并通过跳跃连接整合多阶邻域信息**

1. **残差连接设计：**

* **每两层添加残差连接，缓解过平滑问题**
* **保留原始特征信息，增强梯度传播**

1. **全局上下文增强：**

* **通过全局池化捕获图级特征**
* **将全局信息广播到各节点作为上下文参考**

1. **增强的MLP结构：**

* **使用深层的残差MLP进行最终预测**
* **引入LayerNorm和LeakyReLU增强稳定性**
* **增加Dropout层防止过拟合**

1. **激活函数优化：**

* **使用LeakyReLU（负斜率0.2）替代普通ReLU**
* **缓解神经元死亡问题，保留更多特征信息**

1. **正则化策略：**

* **各层添加BatchNorm/LayerNorm**
* **双重Dropout策略（卷积层和MLP层）**
* **隐含层维度动态调整，平衡表达能力与过拟合**