**🧠 GNN推荐系统模块：完整设计思路（以KRAS-G12C为背景）**

**✅ 一、总目标**

构建一个 **异构生物知识图谱（Heterogeneous Biological Graph）**，以基因为中心节点，整合：

* 图结构（基因间调控/通路/共表达）；
* 数值特征（CRISPR得分、文献共现频率、中心性等）；
* 文本特征（基因功能摘要的Transformer编码）；

并用 GNN 模型进行训练，输出：

* P(gene\_i is resistance driver | G) 的推荐概率；
* 推荐结果附带机制类别、路径结构与解释文本。

**🧱 二、图结构设计**

**节点类型（至少）：**

| **类型** | **说明** |
| --- | --- |
| Gene | 基因节点（主节点） |
| Drug | 药物节点（用于可靶性） |
| Pathway | 通路/机制模块节点（用于机制分类/解释） |

**边类型（可选）：**

| **类型** | **来源** | **权重建议** |
| --- | --- | --- |
| Gene–Gene | STRING (>700 分) | 权重按 score |
| Gene–Pathway | Reactome/KEGG | 权重1或按通路重要性 |
| Gene–Drug | OncoKB/DrugBank | 是否可靶 |

**🧬 三、节点特征构建**

**对基因节点输入的 feature vector x\_i 包含：**

| **特征来源** | **维度类型** | **示例值** |
| --- | --- | --- |
| CRISPR得分 | 数值 | -0.71 |
| 文献共现频率 | 数值（TF-IDF） | 0.43 |
| 语义打标标签 | bool / int | 1（有敲除恢复语义） |
| BioBERT摘要嵌入 | 向量（768维） | [0.12, 0.03, ..., 0.98] |
| 是否为靶点 | bool | 1 |
| 中心性/度数 | 数值 | 0.71 |

🔧 这些可以拼接为一个输入向量：x\_i ∈ R^d，供 GNN 使用。

**🧠 四、模型结构选择**

**推荐结构：GraphSAGE or GAT + MLP**

text

复制编辑

Input: Heterogeneous Graph G(V, E), node features x\_i

Step 1: GraphSAGE聚合邻居信息，更新每个基因嵌入 h\_i

h\_i ← Aggregator({h\_j | j ∈ N(i)}) + W·x\_i

Step 2: 拼接自身特征与聚合向量 → 进入 MLP 分类器

Step 3: 输出 y\_i = σ(MLP(h\_i)) → 预测其是否为耐药因子

* 可支持**半监督训练**（部分有标签数据如SHP2/YAP1）
* 或**无监督节点排序训练**（用负样本对比学习）

**🧪 五、训练方式设计**

| **项目** | **内容** |
| --- | --- |
| 标签来源 | 有功能性打标语义支持的基因 = 正例，DepMap无支持为负例 |
| 损失函数 | BCEWithLogitsLoss（二分类） / Ranking Loss |
| 训练方式 | 分层图训练（邻接采样），PyTorch Geometric 或 DGL 实现 |
| 验证指标 | ROC-AUC、Precision@10、命中率（文献/DepMap） |
| 推荐输出 | Top-N gene + 推荐路径图 + 支持语句 + 概率得分 |

**🎯 六、推荐输出结构设计**

| **Gene** | **推荐得分** | **支持路径** | **是否功能证据** | **推荐机制** | **支持文献摘要** |
| --- | --- | --- | --- | --- | --- |
| SHP2 | 0.921 | EGFR → SHP2 → KRAS | ✅ 有 | MAPK反馈激活 | “SHP2 inhibition reversed resistance to sotorasib.” |
| YAP1 | 0.871 | TEAD → YAP1 → KRAS | ✅ 有 | 旁路信号激活 | “YAP1 knockdown restored sensitivity...” |

**🛠️ 七、可选扩展方向（用于后续期刊投稿或模型增强）**

| **模块** | **扩展内容** | **应用收益** |
| --- | --- | --- |
| 🔄 图更新机制 | 随文献加入新基因/路径动态更新 | 系统迭代可持续 |
| 🎯 解释模块 | SHAP / attention 可视化推荐依据 | 增强可解释性 |
| 👥 多癌种图谱 | 添加共现样本组织类型节点 | 泛化与拓展 |
| 🧠 GNN + Transformer 深融合 | 用Transformer输出控制邻居权重 | 多模态神经融合结构 |

**✅ 总结：你该如何执行？**

我可以为你按以下节奏推进：

1. ✅ 生成节点与边表（STRING + KEGG）
2. ✅ 读取 DepMap & 打标数据 → 构建特征向量矩阵
3. ✅ 生成 PyTorch Geometric 的数据加载结构（Data类）
4. ✅ 构建 GraphSAGE + MLP 模型代码
5. ✅ 输出推荐结构与 Top-N 解释表

**1. ✅ 生成节点与边表（STRING + KEGG）**

**【GNN图谱 · 边表构建模板 + 标准填充流程说明】**

**📄 一、边表模板字段说明（你已拿到文件）**

| **字段名** | **类型** | **示例值** | **含义** |
| --- | --- | --- | --- |
| source | string | G1（SHP2） | 起点节点 ID（来自节点表） |
| target | string | G2（KRAS） | 终点节点 ID |
| edge\_type | string | PPI, Pathway\_Member, Drug\_Target | 边的类型（表示连接关系） |
| weight | float/int | 0.89 / 1.0 / 700 | 边的置信度/重要性，可二值也可比例值 |

👉 文件路径（你已拥有）：

* 📄 GNN\_边表模板.csv

**🧠 二、构建边表的标准流程步骤**

你可参考下列流程，分步骤完成真实图谱边数据的构建：

**✅ 步骤 1：明确你的节点类型与连接逻辑**

| **节点类型组合** | **是否构建边？** | **数据来源建议** |
| --- | --- | --- |
| Gene ↔ Gene | ✅ 是 | STRING、BioGRID（PPI）；共表达 |
| Gene ↔ Pathway | ✅ 是 | KEGG/Reactome通路信息 |
| Gene ↔ Drug | ✅ 是 | DrugBank/OncoKB/DGIdb |
| Drug ↔ Pathway | ❌ 可忽略 | 不重要 |
| Gene ↔ Gene（共现） | 可选 | 文献共现 > 阈值也可作为边 |

**✅ 步骤 2：从指定数据库抓取边数据**

| **数据源** | **用途** | **说明** |
| --- | --- | --- |
| [STRING](https://string-db.org/) | Gene–Gene边 | 可设置最低score阈值（如700）下载 |
| [Reactome](https://reactome.org/) / [KEGG](https://www.kegg.jp/) | Gene–Pathway边 | 每个Pathway条目对应参与基因 |
| DrugBank | Gene–Drug边 | 抓取基因-药物靶点信息 |
| [DGIdb](https://www.dgidb.org/) | 补充靶点-药物边 | 用API获取“基因-可靶向药物”对 |
| [OncoKB](https://www.oncokb.org/) | 高可信药物靶点 | 可人工注入关键边如KRAS–Sotorasib |

**✅ 步骤 3：为每类边打标签和权重**

| **边类型** | **标签** | **权重来源/建议** |
| --- | --- | --- |
| STRING互作 | PPI | 用原始score（0–1000） |
| Pathway关系 | Pathway\_Member | 固定为1或重要性赋值 |
| Drug靶向 | Drug\_Target | 固定为1；若是FDA靶点可设为2 |

**✅ 步骤 4：标准格式整理**

将所有边整理为如下格式的 CSV 表：

csv

复制编辑

source,target,edge\_type,weight

G1,G2,PPI,860

G3,G2,Pathway\_Member,1.0

G1,D1,Drug\_Target,1.0

* **注意**：source 和 target 必须出现在节点表的 node\_id 中。

**🛠️ 三、未来图谱边表可拓展的方向**

| **方向** | **拓展方式** |
| --- | --- |
| 引入方向性 | PPI默认无向，但调控可指定方向 Gene\_A → Gene\_B |
| 添加边特征 | edge\_attr 字段，表示药物类型、调控强度等 |
| 图分层构建 | 多层图（上下游、靶向图）可作为 GNN输入增强项 |

**✅ 总结：你当前应做的准备**

1. 根据此模板决定你将纳入哪些节点类型与边类型；
2. 后续按建议填充边表或交由我实现自动拉取；
3. 确保节点表与边表一致，便于后续转换为 GNN 格式（如 PyTorch Geometric 的 Data 对象）；

**2. ✅ 读取 DepMap & 打标数据 → 构建特征向量矩阵**

**利用 DepMap + 打标信息，构建每个基因的特征向量 x\_i，作为 GNN 输入**

**🧱 输入数据来源说明：**

| **数据类型** | **内容** | **你是否已有** | **我能否执行** |
| --- | --- | --- | --- |
| ✅ 基因列表 | 节点表中的基因名称（如KRAS、SHP2等） | ✔️ 你已有/即将准备 | ✔️ 我可辅助 |
| ✅ DepMap CRISPR数据 | 每个基因的 knockdown 评分（基因效应分数） | 🚫 尚未上传 | ✔️ 我可下载/你也可上传 |
| ✅ 文献共现频率 | 基因在摘要中出现的频次（TF-IDF） | ✔️ 你已抓取1.8万摘要 | ✔️ 可为你提取 |
| ✅ 功能性语义打标 | 基因是否被检测到“敲除恢复敏感性”的句子 | ✔️ 可由我从摘要打标 | ✔️ 模块已构建，可执行 |
| ✅ 是否可靶 | 基因是否是 DrugBank/OncoKB 靶点 | 🚫 你尚未上传 | ✔️ 我可查找构建 |
| ✅ 文本语义嵌入 | BioBERT 向量（可选） | 🚫 你未请求 | ✔️ 可调用预训练模型处理 |

**📘 特征向量示意设计：**

我们将构建如下结构的矩阵：

| **Gene** | **CRISPR\_score** | **TFIDF\_score** | **Has\_semantic\_hit** | **Drug\_target** | **[BERT\_embedding...]** |
| --- | --- | --- | --- | --- | --- |
| SHP2 | -0.71 | 0.57 | 1 | 1 | [0.12, ..., 0.98] |
| AURKA | -0.31 | 0.29 | 0 | 0 | [...] |

→ 用于构建：x\_i ∈ R^d 的特征向量

**✅ 3.生成 PyTorch Geometric 的数据加载结构（Data类）**

**目标：为KRAS-G12C推荐系统构建标准 torch\_geometric.data.Data 输入结构**

你未来只需：

* 准备节点特征矩阵 x
* 节点标签向量（可选）y
* 图结构（边索引）edge\_index
* 其他辅助数据（如边特征、节点类型等）

即可直接运行 GNN 模型（如 GCN / GraphSAGE / GAT 等）

**🧱 一、PyG Data 对象核心结构**

python

复制编辑

from torch\_geometric.data import Data

import torch

# 示例：构建一个简单图结构

x = torch.tensor([[1.0, 0.0], [0.0, 1.0], [1.0, 1.0]], dtype=torch.float) # 3个节点，2维特征

edge\_index = torch.tensor([[0, 1, 2, 0],

[1, 0, 0, 2]], dtype=torch.long) # 4条边（无向）

y = torch.tensor([0, 1, 0], dtype=torch.long) # 可选：节点标签（监督训练时使用）

data = Data(x=x, edge\_index=edge\_index, y=y)

**🔍 对应到你的KRAS-G12C推荐系统：**

| **模块** | **数据类型** | **来源** | **是否必须** |
| --- | --- | --- | --- |
| x | 节点特征矩阵 | TF-IDF + CRISPR + 语义标注等 | ✅ 必须 |
| edge\_index | 图结构边 | 来自 STRING / KEGG | ✅ 必须 |
| y | 节点标签（0/1） | 文献打标 / OpenTargets | ⚠️ 可选（监督任务需要） |
| edge\_attr | 边权重 | STRING得分等 | ⚠️ 可选 |
| node\_type | 节点类型（Gene/Drug/Pathway） | 来自节点表 | ⚠️ 可选，用于异构图 |

**✅构建 GraphSAGE + MLP 模型代码（示例）**

**GraphSAGE + MLP 推荐模型代码（KRAS-G12C推荐系统框架）**

python

复制编辑

import torch

import torch.nn.functional as F

from torch\_geometric.nn import SAGEConv

from torch\_geometric.data import Data

# === 模型结构 ===

class GNNRecommendation(torch.nn.Module):

def \_\_init\_\_(self, input\_dim, hidden\_dim, output\_dim):

super(GNNRecommendation, self).\_\_init\_\_()

self.conv1 = SAGEConv(input\_dim, hidden\_dim)

self.conv2 = SAGEConv(hidden\_dim, hidden\_dim)

self.mlp = torch.nn.Sequential(

torch.nn.Linear(hidden\_dim, hidden\_dim),

torch.nn.ReLU(),

torch.nn.Linear(hidden\_dim, output\_dim)

)

def forward(self, x, edge\_index):

# GNN聚合

x = self.conv1(x, edge\_index)

x = F.relu(x)

x = self.conv2(x, edge\_index)

x = F.relu(x)

# MLP分类

out = self.mlp(x)

return out

**✅ 示例训练流程（Binary Classification）**

python

复制编辑

def train(model, data, optimizer, criterion):

model.train()

optimizer.zero\_grad()

out = model(data.x, data.edge\_index)

loss = criterion(out[data.train\_mask], data.y[data.train\_mask])

loss.backward()

optimizer.step()

return loss.item()

def test(model, data, mask):

model.eval()

with torch.no\_grad():

out = model(data.x, data.edge\_index)

pred = out.argmax(dim=1)

correct = (pred[mask] == data.y[mask]).sum()

acc = correct / mask.sum()

return acc.item()

**📦 Data 对象要求（需你准备）**

| **属性** | **含义** | **示例形状** |
| --- | --- | --- |
| x | 节点特征矩阵 | [N, d]，如 [37, 10] |
| edge\_index | 边索引（COO） | [2, E] |
| y | 节点标签（0/1） | [N] |
| train\_mask, val\_mask, test\_mask | 训练/验证/测试集布尔掩码 | [N] |

**🧠 输出结果解释**

* model(x, edge\_index) 将返回 [N, 2] 维的 logits
* argmax 表示每个节点是否是“耐药因子”候选的预测分类
* 若你使用 output\_dim=1，可替换为 sigmoid 输出概率（推荐值）

**✍️ 你现在可以怎么做：**

1. 准备好 Data 数据对象（可用我前面的 PyG 构建模板）；
2. 决定是否为监督任务（是否提供 y）；
3. 训练 + 输出推荐得分（或直接排序输出 top-N 节点）；

**✅ 5.输出推荐结构与 Top-N 解释表设计**

**🧱 一、推荐结构组成（每个候选基因）**

| **字段名** | **含义** | **来源** |
| --- | --- | --- |
| Gene | 基因名 | 节点名 |
| Score | GNN输出推荐分数（或分类概率） | 模型预测 |
| Rank | 排名（按得分降序） | 排序后生成 |
| Function Evidence | 是否命中“敲除恢复敏感性”语义 | 文献语义打标 |
| Drug Target | 是否已有靶点药物 | DrugBank / OncoKB |
| Pathway Cluster | 所属机制通路 | 图结构推理 or Reactome注释 |
| Supporting Edge | 推荐路径上的关键连接 | GNN message path (可选) |
| Supporting Sentence | 文献中出现的支持语句 | 已识别语义证据 |
| PMID | 支持文献编号 | PubMed数据 |

**📘 二、示例推荐输出表（Top-N）**

| **Rank** | **Gene** | **Score** | **Function Evidence** | **Drug Target** | **Pathway Cluster** | **Support Sentence** | **PMID** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | SHP2 | 0.921 | ✅ 是 | ✅ 是 | MAPK Feedback Loop | “Knockdown of SHP2 re-sensitized cells...” | 37252163 |
| 2 | YAP1 | 0.889 | ✅ 是 | ❌ 否 | Hippo-Bypass | “Loss of YAP1 restored sensitivity…” | 36543217 |
| 3 | AURKA | 0.783 | ❌ 否 | ✅ 是 | Cell Cycle Escape | — | — |

**🧠 三、构建方式建议（自动生成逻辑）**

1. 运行模型，获取每个基因的 score = sigmoid(model(x))
2. 将得分排序，输出 Top-N（可设定阈值）
3. 联合：
   * 语义打标（bool）
   * 靶点药物（bool）
   * 支持语句（打标模块输出）
   * PMID（打标模块输出）

合并为结构化 .csv 或 .xlsx 文件

**📦 输出表用途**

| **场景** | **用途** |
| --- | --- |
| 🔬 生物学分析 | 快速判断推荐是否有功能/药物/通路支持 |
| 📈 模型评估 | 看 Top-K 命中率是否高（文献支持是否充分） |
| 📝 论文图表 | 推荐表格 + 路径结构可作为主图 |
| 🧠 交互界面 | 可用于未来部署前端筛选与结果展示（如 SkywalkR） |