**hetero\_GAT\_2Layer**

**Whether the learned embeddings make sense?**

**Plot embedding similarity(learned) vs. miRNA/Gene/Disease similarity (ground-truth)**

1.get embeddings function:get the node learned embeddings of miRNA, gene and disease

def get\_node\_embeddings(self, g, x, mask=None)

node\_embeddings = model.get\_node\_embeddings(hetero\_graph, node\_features)

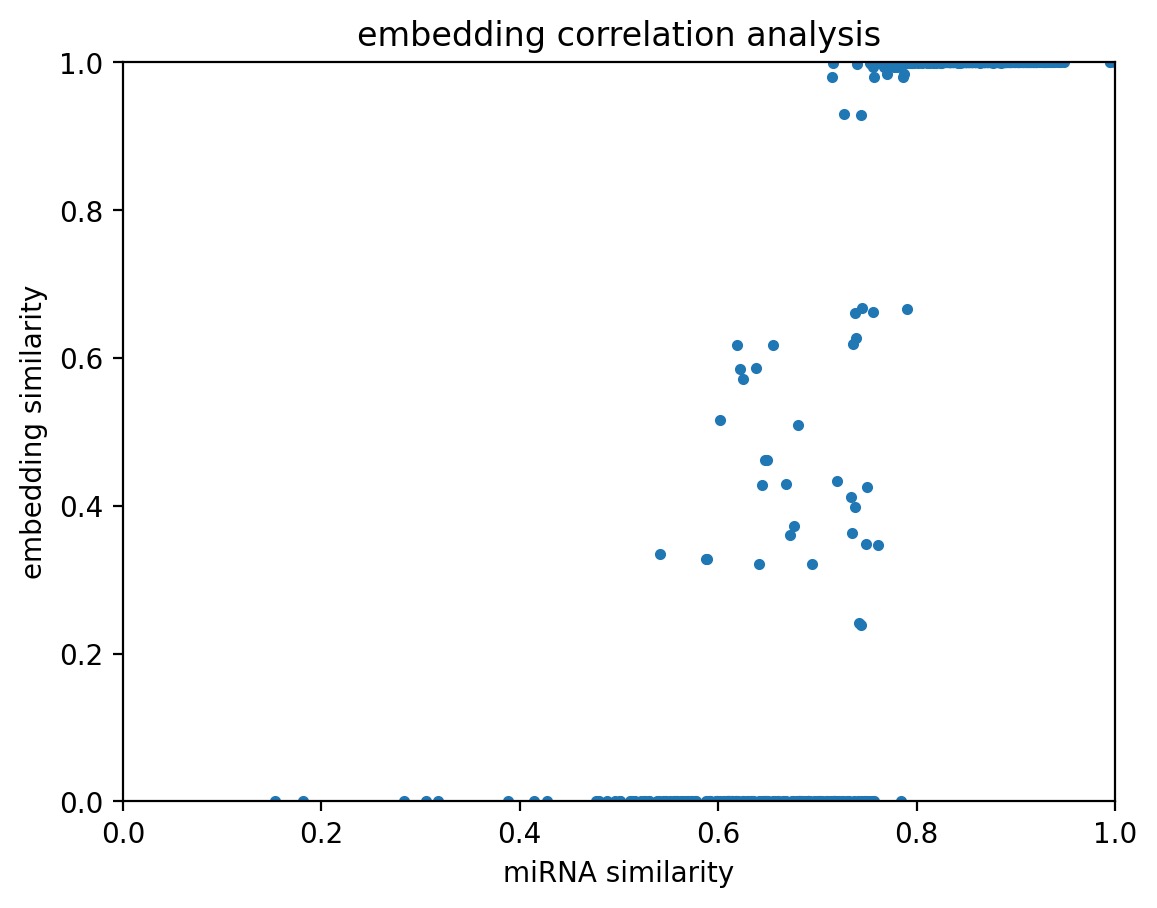
2.Compute similarity between two embeddings

cosine\_similarity

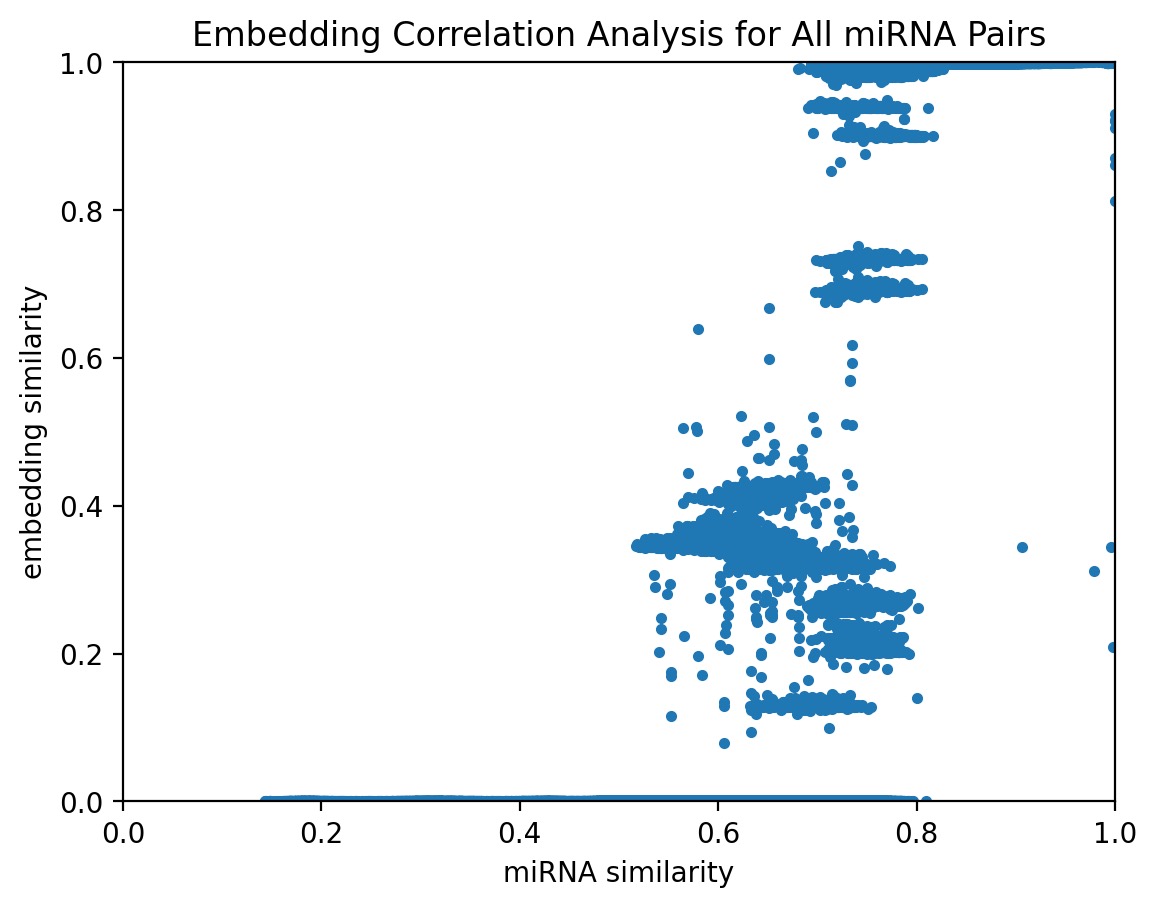
3.Plot embedding similarity vs miRNA/gene/disease similarity

Result: No linear relationship

Method: Random select 700 miRNA pairs



Plot **all miRNA pairs** instead of randomly selecting 700 pairs:



Calculate the correlation between **embedding similarity** and **miRNA similarity:**

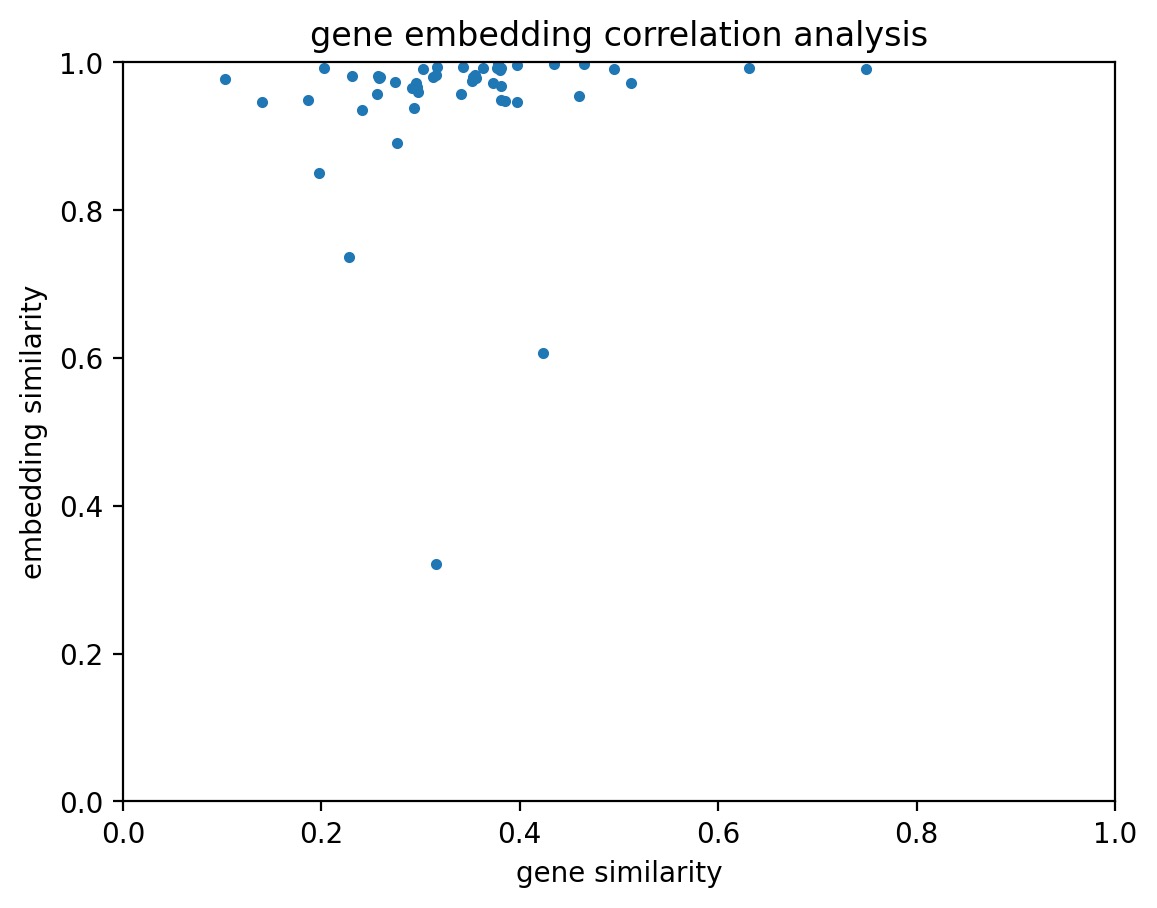
**Pearson Correlation: 0.8366**

**Spearman Correlation: 0.8058**

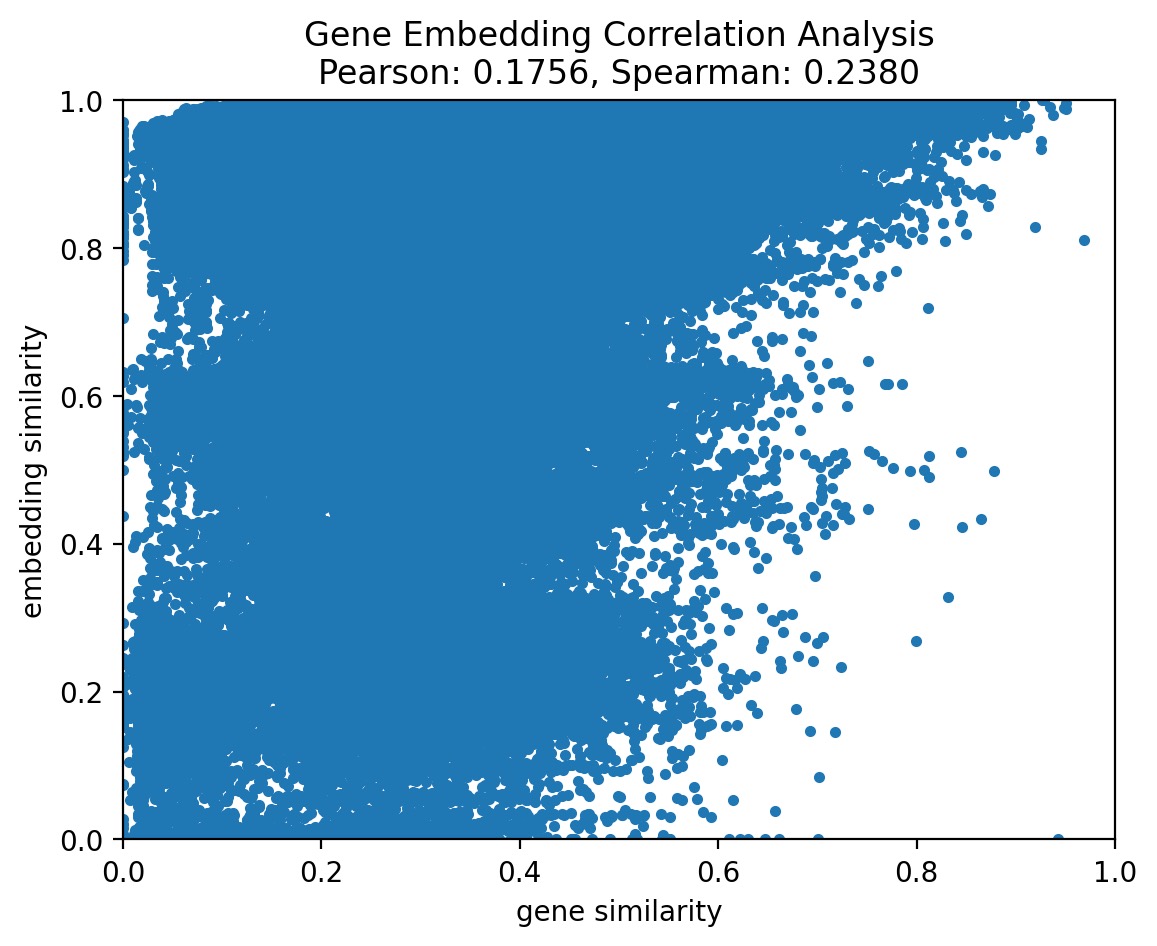
Seems meaningful

Result: No linear relationship

Method: Random select 50 gene pairs



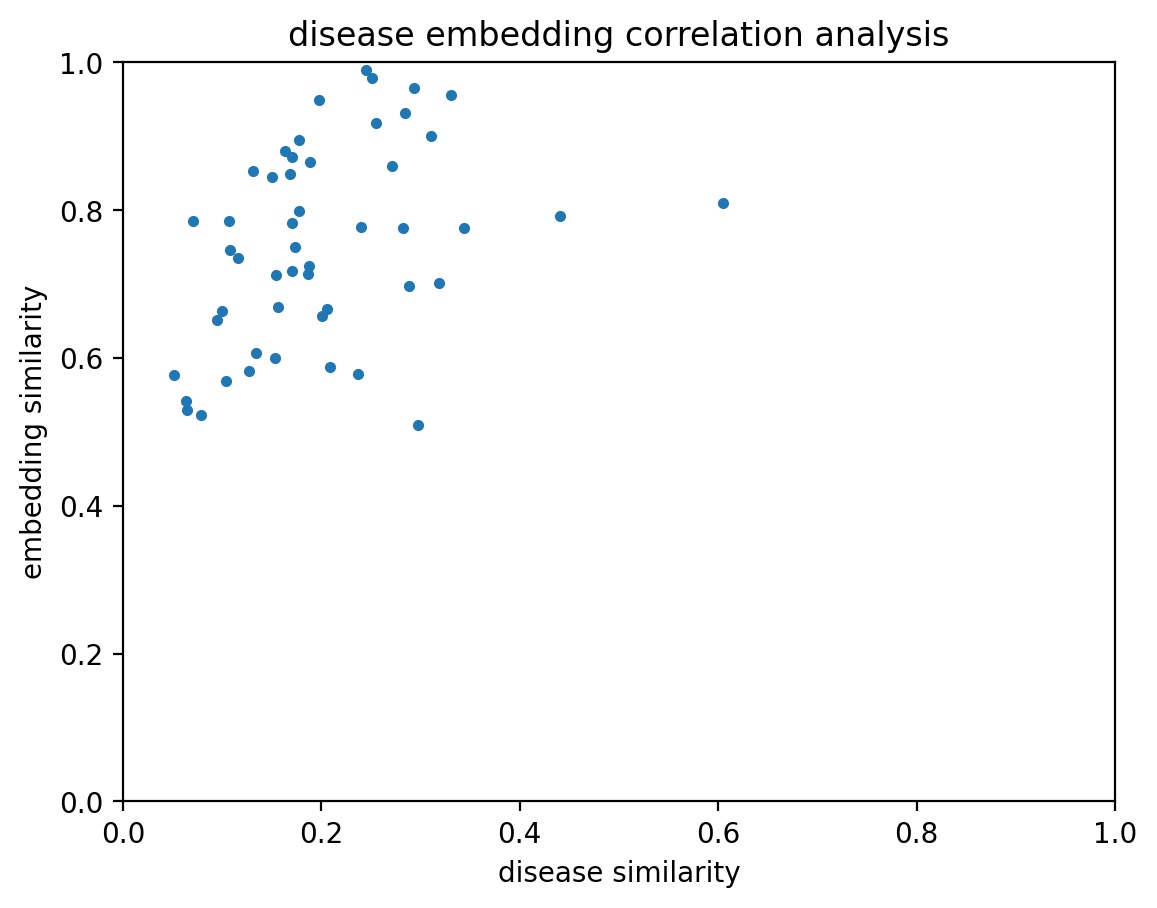
Plot **all gene pairs (giant data points):**



Still unexpected outcome, consistent with gene clustering shape by tSNE embedding plots

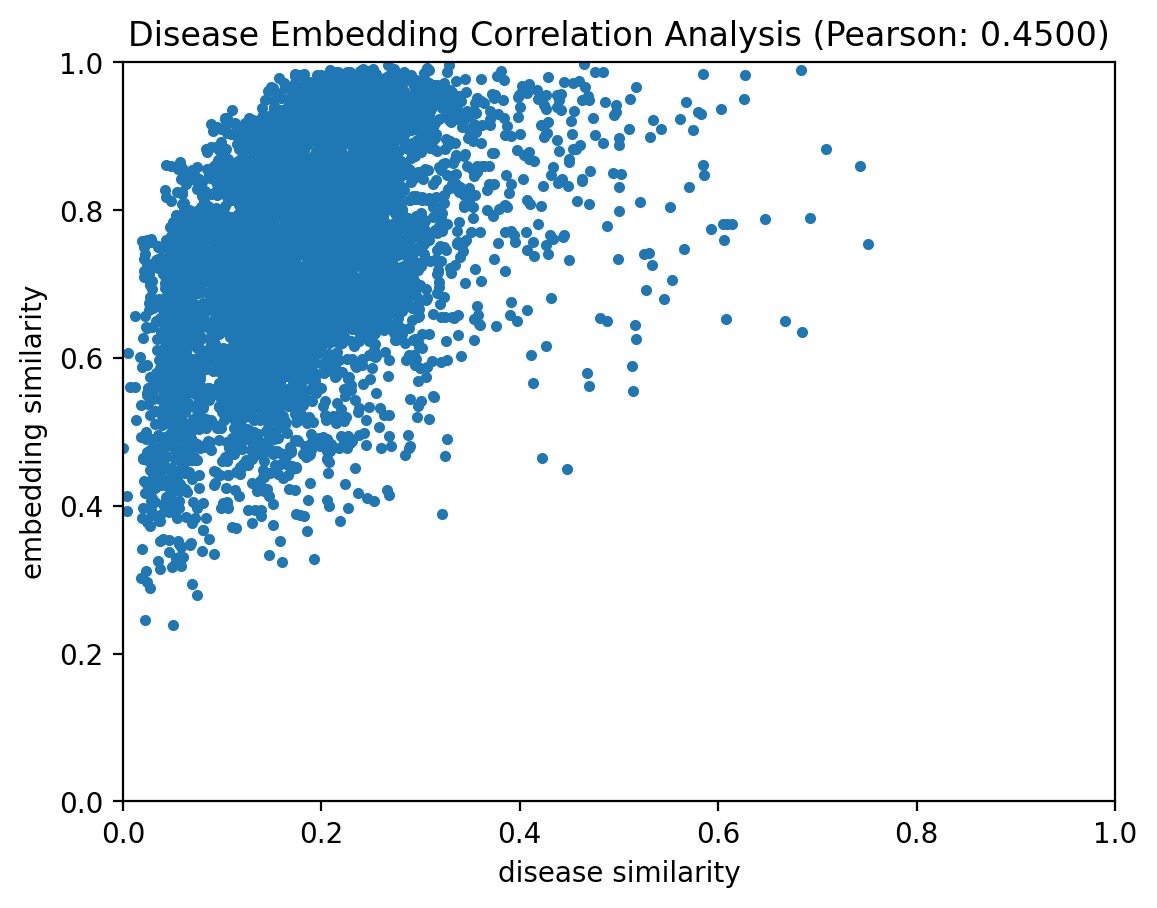
Result: weak linear relationship

Method: Random select 50 disease pairs



Plot **all disease pairs：**

Pearson Correlation: 0.4500, p-value: 0.0000



卡点：

1.model是针对某个edge-type来预测该edge的score

验证make sense under certain edge-type eg. ('miRNA', 'associate\_with', 'disease')

2. Epoch =150（loss最小且稳定）取embeddings

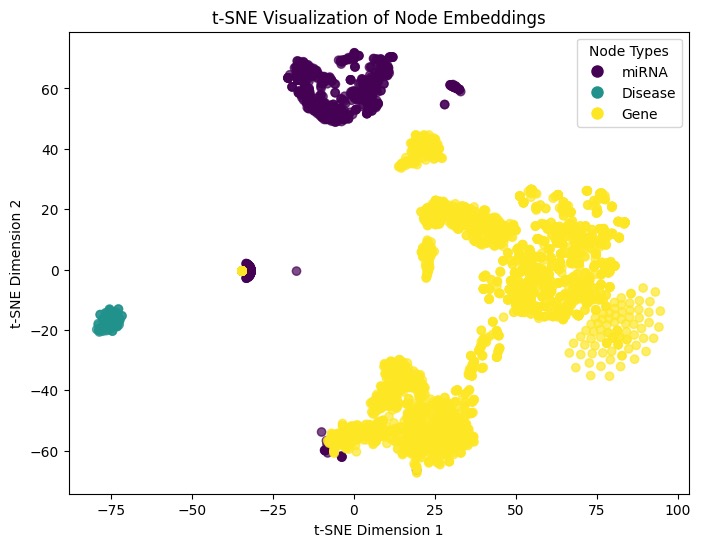
**tSNE/PCA:Plot embeddings**

所有不同颜色label不同node type，看不同的node type是不是cluster在一起eg.看miRNA是不是成一个cluster，disease是不是成一个cluster

The goal is to reduce the high-dimensional embeddings into 2D or 3D space and color-code them based on node types, to see if they form distinct clusters.

### **Step-by-Step Approach:**

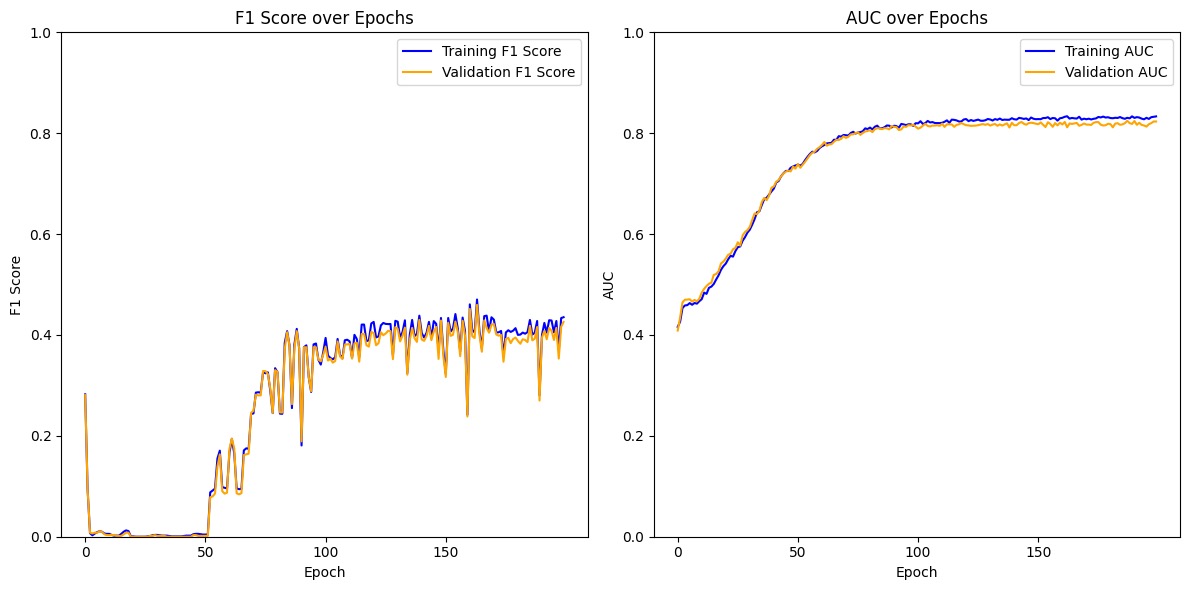
1. **Collect Embeddings for All Node Types**: Gather the embeddings for miRNA, disease, and gene into a single matrix.
2. **Label the Node Types**: Assign labels to the embeddings based on their node types (e.g., 0 for miRNA, 1 for disease, 2 for gene).
3. **Run t-SNE**: Apply t-SNE to reduce the dimensionality of the embeddings.
4. **Plot the Embeddings**: Use matplotlib to create a scatter plot, coloring the points based on their labels.



**Performance Evaluation: Change Accuracy to F1 score/AUC**

Background: the accuracy reach the peak at the early stage of epochs =10

* The fact that the F1 score does not go much higher (e.g., around 0.8-1.0) suggests that the model is still struggling with recall or precision, meaning that it might be missing a lot of positive cases or is generating a lot of false positives.
* The **AUC** increases steadily until around **epoch 100**, where it starts to stabilize around **0.85** for both training and validation. AUC close to 1.0 indicates good separability between the classes, and **0.85** suggests the model is performing fairly well in distinguishing between the positive and negative classes.

****

**Intermediate epoch embedding Check**

Check intermediate epoch embedding to see if the model is overfitting (check every 20 epoch after 50 epoch for embedding similarity)

Results: no overfitting problems

pics:hetero\_GAT -Intermediate embedding.ipynb

但上次跑出来的embedding similarity图跟这次的不一样

**Add node features**

<https://chatgpt.com/share/67185a59-4b28-800e-9078-556864ea2d1e>

1.找到miRNA对于sequence 2. 通过 seq2vector转换成node feature

miRNA: seq2vector

gene: DNABERT

* DNABERT: [link](https://academic.oup.com/bioinformatics/article/37/15/2112/6128680) 拿人类的基因组pre-trained 过后的BERT model

disease: DE (differentiated expressed) gene jaacard index / HPO

**hetero\_GCN\_2Layer**

**GCN**

1:15’虽然accuracy很高，但模型没学到东西（loss没有随epoch变化）

step1:换成F1 score

step2:画embedding

**Future Direction**

1. - Meaningful Initial node features 根据哪两个node之间有连接 模型也可以学到很多东西所以可能影响不大

- similarity计算方法改变

2. Evaluation Index:改成F1(>80%)，AUC

换成别的loss

3. Data imbalance problem

4. Baseline model比较