250103

模型设想

GF全局编码

1. 之前是对DXF文件采取分块的策略，再输入GNNs处理，现在需要对同一份DXF文件的所有块进行整合，并添加位置编码，这里需要设计一个模块来处理(看一些GNNs的论文)，就像子图拼接回原图
2. 对于GF模型的拼接时候引入多头注意力机制

对比设计模块

1. （几何结构，序列语义），（几何结构，序列语义）现在的多模态常见的是从不同数据源得到的数据之间进行对比学习，而我们提出的（几何结构，序列语义）是从同一数据源（同一份DXF文件），但是不同特征提取器得到的特征之间对比学习，也就是多尺度编码之间的对比学习，使用该批次所有可能的匹配对。
2. （几何视图1，几何视图2），以batch为单位进行对比学习，同一份DXF文件的视图与视图之间作为正样本，否则为负样本，如果batch_size=N,则该batch有2N个视图，其中每个视图会计算2N-2个负样本损失，1个正样本损失。
3. （序列视图1，序列视图2）将视图产生部分移到"Sequence Coding Matrix"之后，“DXF Embedding”之前，这依据了（Multimodal Contrastive Training for Visual Representation Learning **CVPR2021**）的原理，视图增强应该是对原始数据进行视图增强，才能增强原始数据，而不是特征变化后再进行视图增强。

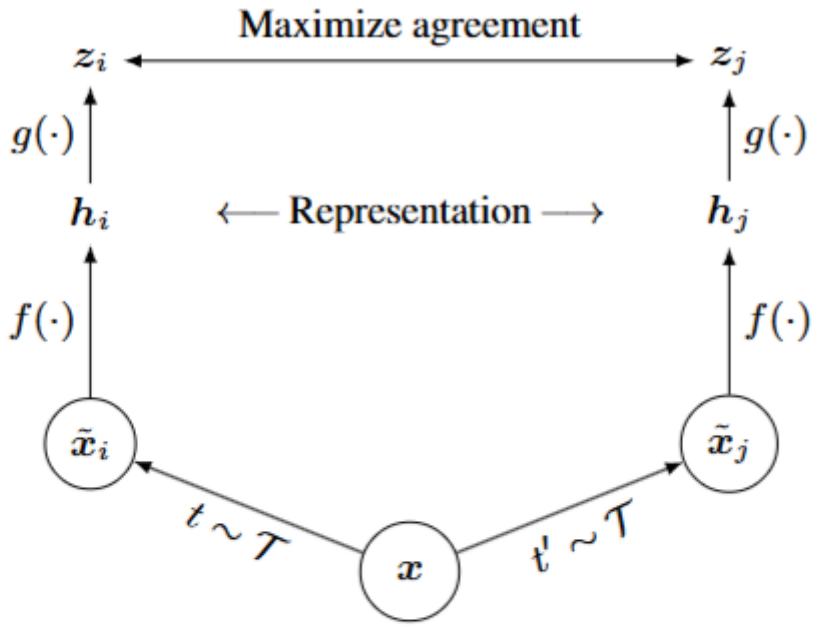


Figure 2. A simple framework for contrastive learning of visua representations. Two separate data augmentation operators ar sampled from the same family of augmentations ($t \sim \mathcal{T}$ an $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlate views. A base encoder network $f(\cdot)$ and a projection head $g($ are trained to maximize agreement using a contrastive loss. Afte training is completed, we throw away the projection head $g(\cdot)$ an use encoder $f(\cdot)$ and representation $\boldsymbol{h}$ for downstream tasks.

损失计算设计

1. 对所有 `i, j ∈ {1, ..., 2N}`，计算投影向量之间的余弦相似度：
   a. sim(i,j) = (z_i^T * z_j) / (||z_i|| * ||z_j||)
   b. 其中 `||z_i||` 和 `||z_j||` 分别是 `z_i` 和 `z_j` 的 L2 范数。
   c. ℓ(i, j) = -log [exp(s_i,j / τ) / Σ [1_{k≠i} * exp(s_i,k / τ)]]
   d. L = (1 / (2N)) * Σ [ℓ(2k-1, 2k) + ℓ(2k, 2k-1)]
2. **（几何结构，序列语义）**的损失计算公式使用InfoNCE损失

公式

对比预训练目标可以表示为：

$$\mathcal{L}_{\text{contrastive}} = -\sum_{i=1}^{N} \log \frac{\exp(\text{sim}(I_i, T_i)/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(I_i, T_j)/\tau)}$$

其中：

- I_i 和 T_i 是第 i 对的图像和文本嵌入。
- $\text{sim}(I_i, T_j)$ 是图像和文本嵌入之间的余弦相似度。
- τ 是温度参数。

该目标鼓励模型最大化匹配图像-文本对的相似度，同时最小化不匹配对的相似度。

3. （几何视图1，几何视图2），（序列视图1，序列视图2）则使用InfoNCE损失
4. 组合的损失计算的参数去实验，找经验值，我们可以看看训练过程，通过参数调整各个损失值之间的大小，达到数量上的平衡

- **总损失**：将模态内和模态间的对比损失结合起来，形成最终的多模态对比训练损失函数：

$$J = \lambda_{ii} J_{ii} + \lambda_{tag} J_{tag} + \lambda_{cc} J_{cc} + \lambda_{ic} J_{ic} + \lambda_{ci} J_{ci} \quad (15)$$

其中 λ_{ii}、λ_{tag}、λ_{cc}、λ_{ic} 和 λ_{ci} 是不同对比损失之间的权衡参数。

- **灵活性**：本文方法不要求所有图像都有标签。对于只有标签或描述的图像，特征学习由 $\lambda_{tag} J_{tag} + \lambda_{ii} J_{ii}$ 或 $\lambda_{ii} J_{ii} + \lambda_{ci} J_{ci} + \lambda_{ic} J_{ic} + \lambda_{cc} J_{cc}$ 指导。

(1) 经验值

- 在论文中，作者提到将 w_p 设置为 **0.1**。这是一个经验值，通常基于初步实验和领域知识确定。
- 选择较小的 w_p 值（如 0.1）是因为重建损失主要用于正则化热图处理器，而不是模型的主要优化目标。

投影层设计

使用投影头网络将表示映射到对比损失空间（在投影空间中定义对比损失比在原始表示空间中更有效，SimCLR 2020 PMLR）

1. 为确保确保模态内和模态间训练方案不相互干扰（Multimodal Contrastive Training for Visual Representation Learning **CVPR2021**），修改DeepDXF的模块内对比学习的投影层为128维MLP，模块间对比学习的投影层为1024维的MLP（这里具体维度是举例子，但是对于对比学习几何特征投影后的维度应该和序列语义投影后的维度一致），在”Transformer Encoder"之后添加两个方向的投影模块
2. GF则在GNNs之后两条线，线路一：添加图池化层+MLP)Contrastive Multi-View Representation Learning on Graphs PRLM 2020)，再进行模态间的对比学习，线路二：在经过图匹配网络完后，添加MLP，并经过MLP投影后再进行模态内的对比学习，MLP可以是2层等

图池化层:
论文使用了一个类似于**Jumping Knowledge Network (JK-Net)**的图池化函数，将每个GCN层的节点表示求和并拼接，然后通过一个单层前馈网络生成图表示:

$$\tilde{h}_g = \sigma\left(\Big\|_{l=1}^{L}\left[\sum_{i=1}^{n}\tilde{h}_i^{(l)}\right]W\right)$$

其中:

- $\tilde{h}_i^{(l)}$ 是第 l 层节点 i 的潜在表示。
- $\|$ 是拼接操作符。
- L 是GCN层数。
- W 是网络参数。

3. 在投影完成后，计算对比损失的点积之前，对所有特征向量进行L2归一化

下游任务（相似度计算）

1. 训练完成后，丢弃投影头网络，仅保留基础编码器网络，用于下游任务（”投影头之前的表示仍然比投影头之后的表示更具信息量，因为投影头可能会移除对下游任务有用的信息” SimCLR）
2. 方案一：对序列编码之间计算相似度，对几何编码之间计算相似度，两者按照训练设置的损失权重各自的占比进行加权得到总相似度

批（层）归一化和残差连接

1. 最后再设计
2. “"批归一化（Batch Normalization，简称 BN）是一种用于深度神经网络的技术，旨在加速训练过程并提高模型的稳定性和性能。它通过对每一层的输入进行归一化处理，使得输入数据的分布更加稳定，从而缓解了训练过程中出现的**内部协变量偏移**（Internal Covariate Shift）问题。""

数据集统一表示

1. 几何结构和序列语义数据打包方案一：两者定义统一的IDX（D={(Ij,cj)}）

2. 负样本对的数量
3. 对于GF模型和DeepDXF模型的数据要统一（因为DeepDXF限制了实体数量最大为4096，造成DXF文件损失，而GF是没有影响的，因此需要对GF模型做筛选）
4. 对每次使用（每个epoch）时候需要随机打散数据
5. 一开始预留的最终评估的数据也需要打散后再来
6. 对于DeepDXF和GF数据，可以像下面一样解包，以及逐个对比计算，我们可以把DeepDXF和GF模型的数据都先加载进来再打包

3. 解包数据的代码实现

- **解包数据 x_i 为 (T_i, I_i) 和可选的 E_i：**
  - 在训练过程中，每个样本 x_i 包含图像 I_i、文本 T_i 和可选的专家热图 E_i。解包数据的代码示例如下：

python 复制

```python
def unpack_batch(batch):
    if len(batch) == 2:
        image, text = batch  # 没有专家热图
        heatmap_mask, text_snippet = None, None
    elif len(batch) == 4:
        image, text, heatmap_mask, text_snippet = batch  # 包含专家热图
    else:
        raise ValueError("Unexpected batch size")
    return image, text, heatmap_mask, text_snippet
```

5. 确保文本嵌入和图像嵌入一一对应

- **嵌入列表的构建：**
  - 在训练过程中，文本嵌入 t_i 和图像嵌入 v_i 需要一一对应。这可以通过将它们的索引保持一致来实现。以下是一个伪代码示例：

python 复制

```python
V = []  # 图像嵌入列表
T = []  # 文本嵌入列表

for i in range(N):  # N 是小批量的大小
    t_i = g(T_i)  # 计算文本嵌入
    v_i = f(I_i)  # 计算图像嵌入
    V.append(v_i)
    T.append(t_i)

    if v_i_lambda exists:  # 如果有混合图像嵌入
        V.append(v_i_lambda)
        T.append(t_i)  # 使用相同的文本嵌入
```

数据增强

1. 数据增强的幅度大点（dropout等设置）
2. GF模型没有使用的数据，应该设置为-1，而不是0
3. DeepDXF的原始数据增强，注意后面是有用SOS填充，对于第一个维度是类型编码，是有实际意义的，不可随便取，并且如果后面维度数字随机，但是第一维是SOS，则是不合理（数据增强不可破坏其合理性），需要找个一个合理的数据增强策略
4. 如果能够找到合适的数据增强策略，那么就将2视图，扩展到4视图（droupout的损失从小到大变化，就如同图像亮度从小到大变化，每一次变化都作为原图的一个视图）
5. 利用 Mixup 策略生成合成样本或者是使用插值的方式产生合成数据，类似于VIT为了适应新的分辨率的图像采用的双线性插值

- **Mixup 数据增强**：
  - 由于专家注释的数据量有限，eCLIP 采用 **Mixup** 数据增强策略，通过将原始图像 I_i 和其专家版本 I_i^E 进行线性插值，生成新的合成样本 $I_i^\lambda = \lambda I_i + (1-\lambda) I_i^E$，其中 $\lambda \sim \text{Beta}(\alpha, \alpha)$。
  - 这些专家嵌入形成新的正样本对 (v_i^λ, t_i) 以及相应的负样本对，用于计算 CLIP 的 InfoNCE 损失。

参数设置

1. 对比损失公式的温度参数 τ 设置为0.07
2. SGD优化器，权重衰减为1e-4（具体的优化器，我们可以调整，这不是重点）

   或者是LARS优化器(You et al., 2017)，学习率为4.8（=0.3×批量大小/256）（具体数据需要我们自己再去算），权重衰减为10-6。

   或者是优化器：使用AdamW，权重衰减为0.02。
3. 批量大小[256,8192]，越大越好
4. GNNs编码器和transformer编码器的初始学习率分别为x和y，（前10%epoch（迭代）使用线性预热，学习率使用余弦衰减调度无重启（余弦退火））

   学习率：初始为1e—5，经过2,000次迭代后增加到1e—4，然后通过余弦衰减策略降至1e—5.
5. 先在1个GPU上跑，尝试修改代码，实现分布式训练，但是要注意下述全局批归一化（SimCLR)
6. 调大epochs有益,对比学习很看重长时间训练（[100,1000]),注意前提是批次是随机重采样的
7. 增大transformer和GNNs的层数，注意力头数量等，使用更大更深的网络

训练策略

1. 先不采用十折交叉验证等策略，使用朴素的方式训练，验证，测试

2. 继续使用wandb训练

实验设想

一定要进行大量的实验！！！

消融实验

评估了模型中关键组件的影响

1. 不融合的情况，两条线（GNNs和transformer encoder）各自作为一个模型来测试结果
2. **（几何结构，序列语义）的损失计算公式可以测试下述公式**

1. 图像到描述对比学习

- **查询和键特征**：给定图像-描述对 (I_j, c_j)，使用图像编码器生成查询特征 q_{ic}^j，使用动量文本编码器生成键特征 k_{ic}^j，并将它们映射到共同空间：

$$q_{ic}^j = f_{iq}(I_j^\dagger; \theta_q, \phi_{cq}) \quad (9)$$

$$k_{ic}^j = f_{ck}(c_j^s; \Theta_k, \Phi_{ik}) \quad (10)$$

其中 ϕ_{cq} 和 Φ_{ik} 是不同于 ϕ_{iq} 和 Φ_{ck} 的MLP层参数。

- **对比损失**：在共同空间中，目标是同时最小化 q_{ic} 和正键特征 k_{ic}^+ 之间的距离，并最大化 q_{ic} 与队列中所有其他负键特征之间的距离。图像-描述对比损失定义为：

$$J_{ic} = \sum_{j=1}^{K} [\alpha - q_{ic} \cdot k_{ic}^+ + q_{ic} \cdot k_{ic}^j]_+ \quad (11)$$

其中 α 是边界，$\cdot$ 表示相似度得分，$[x]_+$ 表示 $\max(x, 0)$。

对比实验

1. 对GNNs和transformer encode引入动量编码器，对比实验效果
2. 因为GF和DeepDXF各自最终的损失计算的数据维度不同，前者

[batch_size, max_nodes, perspectives]

后者（经过投影）

(batch_size, 256)

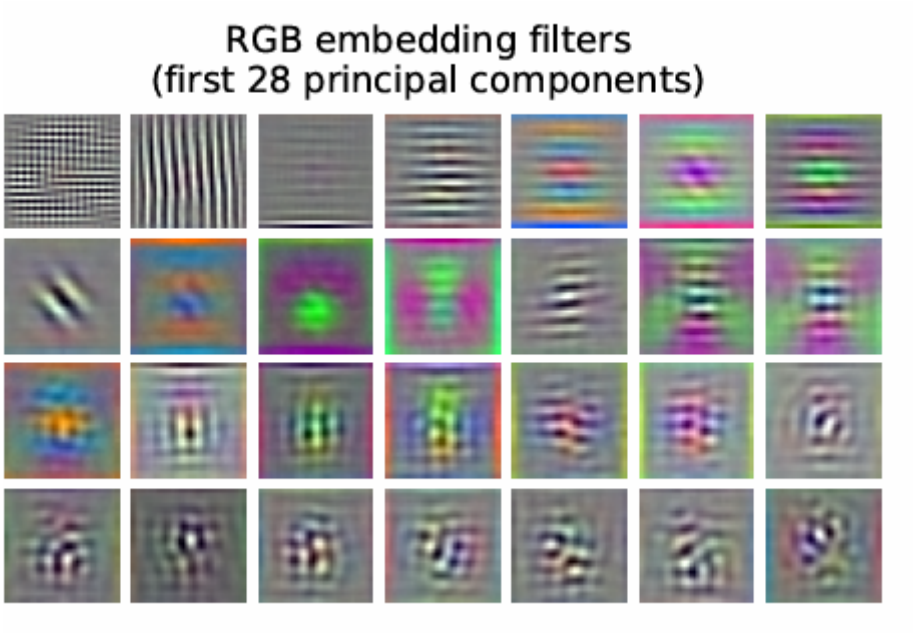
速度上差一个维度，先考虑慢的（也就是统一为GF模型），再考虑快的（统一为DeepDXF模型）对比看效果，因为更多的计算，代表更细致的损失（可以理解为节点之间的计算），效果理论上会更好

课程学习策略

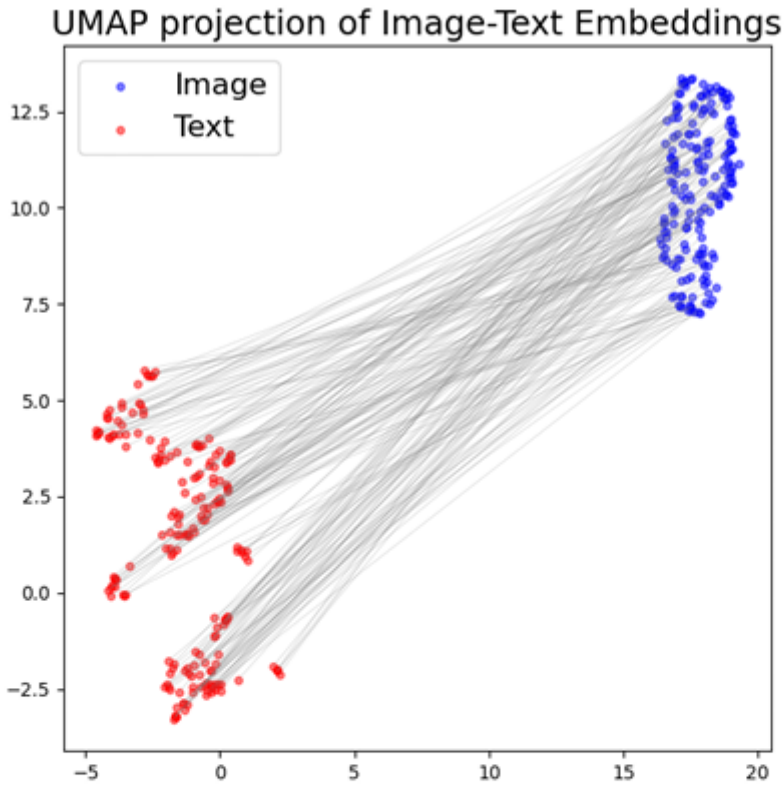
1. 如果合成数据成功得到，采用课程学习训练逐步引入合成数据，对比效果（Improving Medical Multi-modal Contrastive Learning with Expert Annotations ECCV 2024）

成果可视化

1. 嵌入后特征相关的图



2. 这种可视化方法可以学习



(b) Modality Gap. Despite the CLIP contrastive loss aiming at closely aligning image and text embeddings within a shared space, the modalities remain segregated into distinct regions.

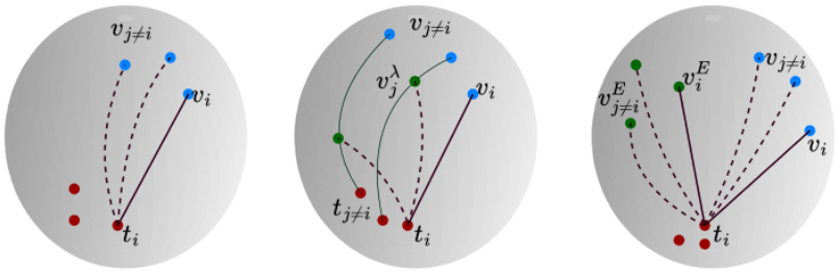
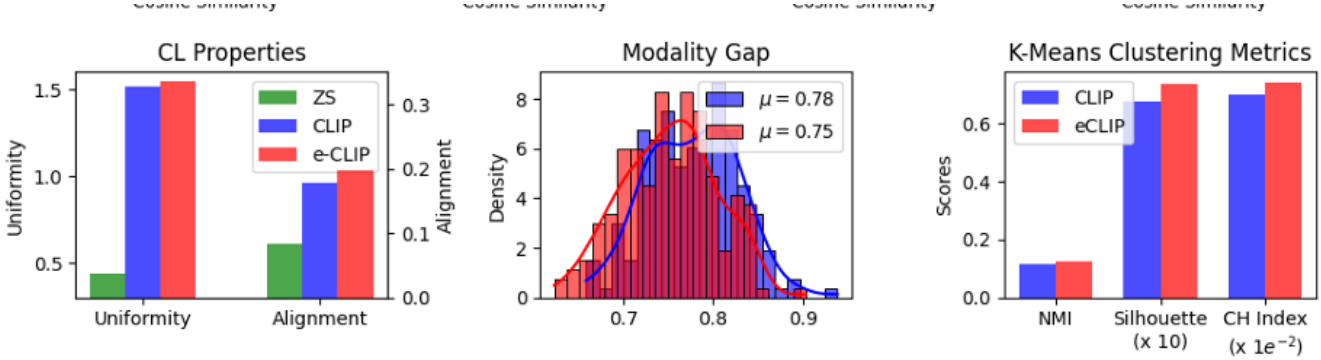


Fig. 3: Comparing eCLIP with m^2-mixup [33]. *(left)* Standard CLIP showing image-text positive pairs (v_i, t_i) (solid line), while the other image embeddings serve as negative pairs (dashed line). *(center)* the m^2-mixup creates negative pairs (v_j^λ, t_i) via interpolation between embeddings along the geodesic. *(right)* eCLIP adds expert image embedding, v_i^E, in addition to v_i for text t_i, forming additional positive and negative pairs

3. 一致性和对齐性的图和指标分析，肯定可以做实验

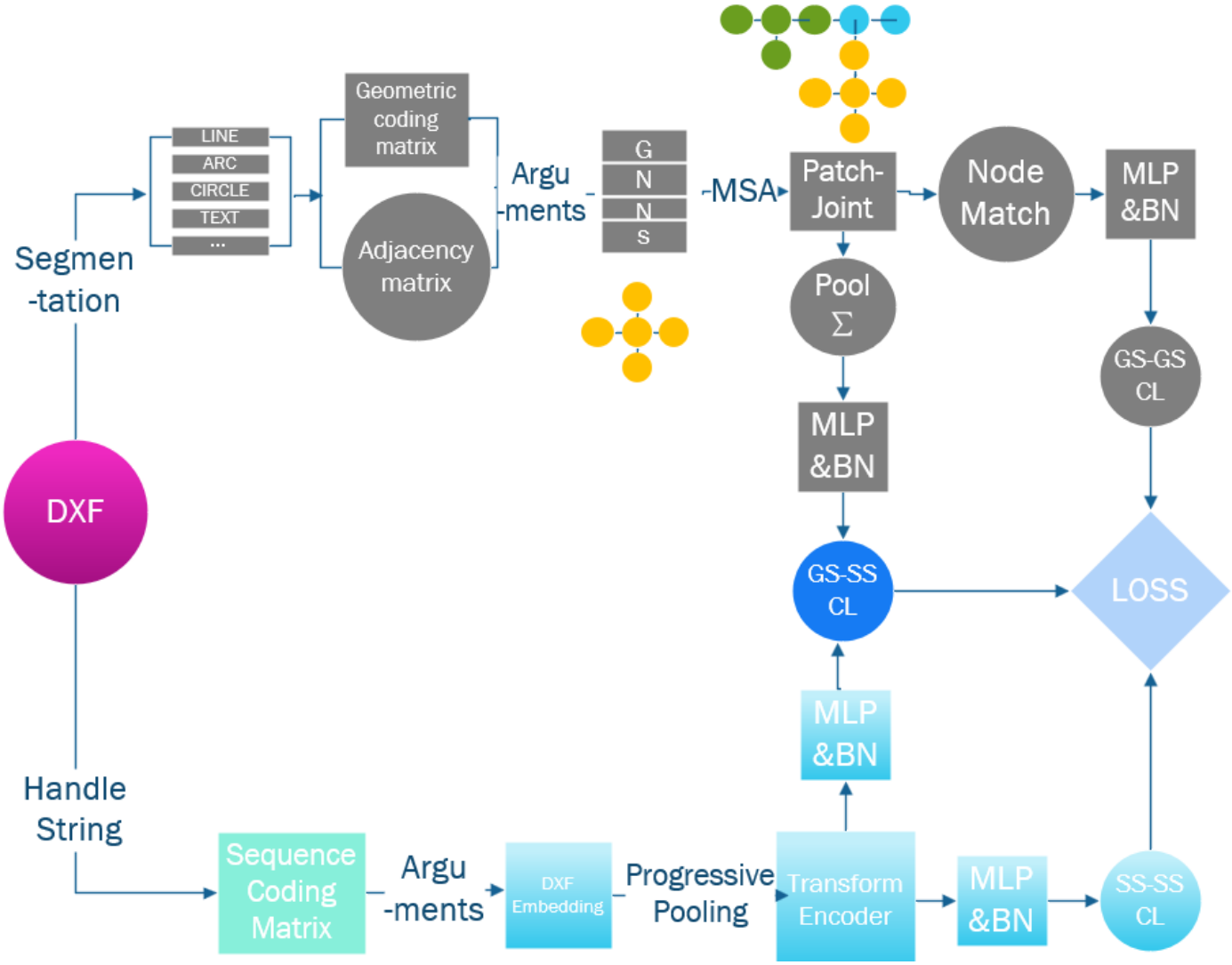


4. 超球体必画

数据设想

1. 把Zero-shot的能力作为实验评估，根据我们的模型，从网上找风格不同的公开的DXF数据进行测试实验,或者说这是对比实验，这样我们提出的框架就具有通用性了
2. 使用CAD论文相关的数据集，自己再处理成DXF，CAD-GPT: Synthesising CAD Construction Sequence with Spatial Reasoning-Enhanced Multimodal LLMs AAAI 2025

模型图


DXF
Segmen
-tation
LINE
ARC
CIRCLE
TEXT
...
Geometric coding matrix
Adjacency matrix
Argu
-ments
G
N
N
S
-MSA
Patch-Joint
Node Match
MLP &BN
Pool Σ
GS-GS CL
MLP &BN
GS-SS CL
LOSS
MLP &BN
Handle String
Sequence Coding Matrix
Argu
-ments
DXF Embedding
Progressive Pooling
Transform Encoder
MLP &BN
SS-SS CL