3DSSG 模型和 EdgeOriented 模型的复现与改进

李逸凡

摘要

场景图本质上是一种数据的结构化表示方法，可以看做是小型的知识图谱。而场景图生成则是构造这种小型知识图谱的过程，本质上是通过对输入数据的理解识别输入数据中的主体和主体之间的关系。目前场景图生成任务在图像领域大放光彩，但是在 3D 领域的研究进展相对缓慢。因此本次实验是对 3D 场景图生成的经典模型 3DSSG 进行复现，同时也复现了另一篇 EdgeOriented 模型并探究其在 3DSSG 数据集和其自身提出的数据集上的效果。从结果来看，3DSSG 模型和论文中所描述的效果基本一致，达到了复现的要求，EdgeOriented 模型虽结果稍差，但也展现出了其优于基础框架的表现。

关键词：场景图生成；3D 点云

1 引言

场景图本质上是一种数据的结构化表示方法，可以看做是小型的知识图谱，因此可以广泛应用于知识管理、推理、检索、推荐等下游任务。该表示方法是模态无关的，自然语言、视频、语音等数据都可以表示成这种结构类型，可用于多模态信息的融合。3D 场景图在 3D 场景分析任务的基础上，能进一步帮助智能体理解物体之间的关系。3D 场景图生成任务旨在对 3D 室内或者室外场景，在识别物体的同时自动构造物体之间的关系。该课题的研究成果可广泛应用于自动驾驶，3D 场景重建和 3D 场景生成等任务，具有一定的学术价值与应用潜力。同时本文所复现的模型是结构化 3D 场景图生成的基础模型，为后续的研究展开构造了基本框架，方便后续研究的推进。

2 相关工作

2.1 2D 场景图生成

早期 2D 场景图生成工作[1][2][3][4]倾向于分别检测图像中的每一个物体和关系，而忽略了物体和关系之间的联系。因此，近年的 2D 的场景图生成工作专注于探究物体之间视觉背景信息，并可以被大致分为三类，分别是基于卷积神经网络（CNN）、循环神经网络（RNN）、图卷积神经网络（GNN）。

2.1.1 基于卷积神经网络

基于 CNN 的场景图生成使用卷积神经网络来获取局部和全局的视觉特征。然后通过分类任务来预测主体和客体之间的关系。Zhang 等人[5]在 Rel-PN 模型[4]的基础上考虑视觉、空间和语义三种特征类型，并通过相对应的模型来学习它们，和 Rel-PN 不同的地方就在于学习了语义特征，此举取得了更好的效果。LinkNet[6]引入了全局背景编码模块和几何布局编码模块，从整个图像中提取全局背景信息和物体区域之间的空间信息，从而提高算法的性能。Zoom-Net[7]通过使用深度信息传播和局部物体特征与全局关系特征之间的相互作用来识别了复杂的视觉关系。BarCNN[8]认为即便是最先进的特征

提取器中的神经元感受野也是收到限制的，因此利用了一个盒注意力机制，这种机制的使用可以在物体识别阶段使用现有的物体识别模型而不用加入冗余的内容。

2.1.2 基于循环神经网络

基于 RNN 的网络模型会利用 RNN 对背景信息进行编码，例如 Xu[9]利用 RNN 构造图推断，将物体和边的特征进行交互。Zeller[10]采用了双向长短时记忆（LSTM）模块，利用背景信息来完善物体的特征表示。Tang[11]则采用了基于树结构的 LSTM 模块，捕捉了视觉关系的平行关系和分层关系，为场景图生成提供了新的思路。Pannet[12]使用了一个两阶段的关系联系网络，分别抓取了背景信息和关系的对齐特征。Gao 等人[13]提出了一个基于视觉特征机制的分层的循环神经网络，采用层次化的 RNN 来模拟关系三要素，以更有效地处理长期的背景信息和序列信息。Y Teng 等人[14]采用三元组的形式直接推断场景图的内容，消除了过去场景图生成方法中冗余的剪枝过程，并构造伪标签辅助训练。

2.1.3 基于图卷积神经网络

基于图卷积神经网络的模型采用 GNN 对背景信息进行编码，例如 Yang[15]等人修剪原始的场景图为了生成一个稀疏的候选图结构。然后利用基于注意力机制的图卷积神经网络来整合全局背景信息。R Herzig 等人[16]认为在 RNN 和 LSTM 中模型的输入顺序是固定的，但是在实际情况中应当可以做到即使改变了输入的顺序，在相同的特征下模型应当产生相同的输出。因此使用了一个排列组合不变性的结构来预测节点的关系。Lin[21] 等人提出了一个方向感知的信息传递模块，对边缘方向信息进行编码。同时 Yan 等人[17]证明了目前的 GNN 方法是基于同质的场景图，因此提出了异质的场景图生成方法。Lin Xin[18]等人在此基础上利用 Transformer 和负权重图卷积的方式构造了异质场景图生成算法。RU-Net[19]发现目前基于 GNN 的网络会受到节点之间的虚假关联性的负面影响，因此利用图正则化方法构造了一个鲁棒的消息传递机制来对抗节点之间的错误连接。

2.2 3D 场景图生成

2.2.1 层次化 3D 场景图

层次化 3D 场景图主要任务是利用二维图像数据分析构造多层次三维场景。李飞飞团队[20]提出了由建筑、房间、物体、相机层构造的场景图，具体通过融合不同全景图像中的相同节点，构造出从多张全景图中生成 3D 场景图的半自动框架。U. Kim[21]同样在图像数据集上进行场景理解，并通过位姿估计和同节点识别构造三维场景图。Shin D[22]则认为 3D 场景信息不止包含目标类别，位置和属性也是非常的重要。因此利用目标检测网络、属性检测网络和关系检测网络三者联合学习场景信息。以上方法主要通过二维场景信息来构造三维场景，本质上并没有直接学习三维场景的特征并且只能已静态的方式构造，无法做到实时构造或者动态构造。

2.2.2 动态场景图

动态场景图是指场景中的物体位置和关系会随时间发生动态改变，因此需要动态的预测目前场景中的物体和物体关系。A. Rosinol[23]利用 Kimera[21]的感知框架的基础上加入了对动态物体的检测和追

踪，实现了动态场景图的构建。但是其只能检测某一个物体（人）在环境中的实时变化，场景的整体信息需要预先生成。同时其在真实环境中其实无法全方位的检测环境中的动态变化，因此在实际运用上有很大困难。

2.2.3 实时场景图

实时场景图构建是指在完成三维扫描的同时实时学习场景信息。Hydra[24]通过机器人在连续室内环境中扫描来实现实时的层次化三维场景图的构建。SceneGraphFusion[25]则以 RGB-D 帧作为输入，在三维测绘的同时，逐步建立全局一致的语义场景图，且不需要预先的场景信息。

2.2.4 结构化场景图

结构化 3D 场景图直接对三维数据进行分析，构造单层的场景结构。Johanna Wald[26]等人在 3RScan 数据集基础上提出了基于点云的多场景的室内场景图数据集 3DSSG，并构造了利用 GCN 进行学习基准模型（benchmark）来分析场景的语义信息。在此基础上，Wu F[27]利用 DGCNN 特征提取网络优化特征提取过程，同时利用图注意力网络来优化节点和边的特征传递过程。Zhang C[28]等人则提出 EdgeGCN 来利用多维边特征进行明确的关系建模，同时探索节点和边之间的交互机制，以实现场景图表示的独立演变。同时，他们还优化了 3DSSG 数据集的标注信息，删除了过于冗余的信息标注，但这也破坏了数据集的多样性。为了引入先验知识，Zhang S[29]等人利用类别标签构造先验知识来抑制场景理解过程中由于视觉外观相似性和其他知觉混乱引起的误差，并通过门控神经单元和 GCN 的网络构架完成节点之间的信息传递。

3 本文方法

本文是对 3DSSG 和 EdgeOriented 两篇论文进行模型复现，因此将分别介绍两篇模型的内容。

3.1 3DSSG 模型

3.1.1 方法概述

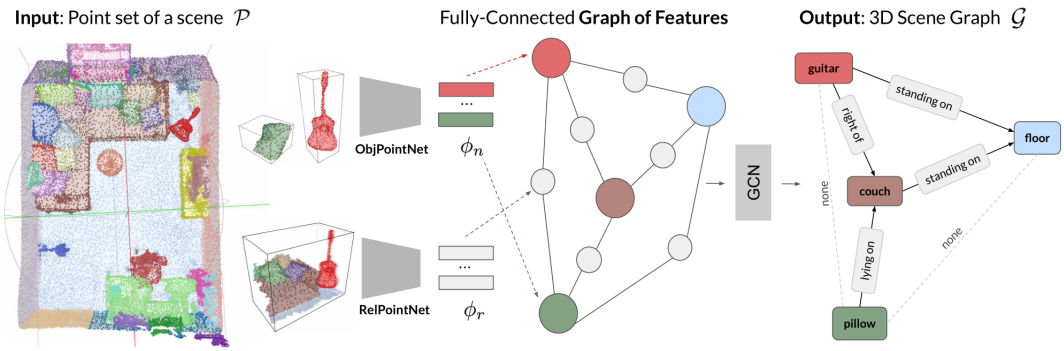

图 1: 3DSSG 模型结构

3DSSG 模型作为 3D 领域场景图生成的基本框架，其模型结构相对简单，主要由特征提取层，特征交互层两层组成。特征提取层负责对点云进行特征预处理，提取点云特征，特征交互层则是根据物体位置信息传递物体之间的特征。模型的结构如图 1所示。

3.1.2 特征提取模块

给定场景 S 的点集 P 和类不可知的实例分割 M，场景图预测网络（SGPN）的目标是生成图 $G=(N,R)$，包含场景中的对象 N 以及他们的关系 R。该模型的学习方法基于场景图预测中的通用框架，其中涉及到每个节点 ϕ_n 和边缘 ϕ_r 的视觉特征提取。该模型使用两种 PointNet 体系结构来提取 ϕ_n 和 ϕ_r，我们将其称为 ObjPointNet 和 RelPointNet，首先对于实例 i 的组成点云 $\mathcal{P}_\rangle$，通过以下公式获得：

$$P_i = \{\delta_{m_k i} \odot p_k\}_{k=1,|P|} \tag{1}$$

δ 表示克罗内克函数，它的输入为 m_k, i，m_k 表示实例 i 的点云掩码，k 表示场景编号（在场景集合 $|\mathcal{P}|$ 中），这一步的目的就是确定节点的点云集合。然后将 $\mathcal{P}_\rangle$ 输入 ObjPointNet。然后用 3D bounding box 的方法提取出物体对 $\mathcal{P}_{\rangle,|}$ 的点云集合，其中 $\mathcal{B}$ 表示 bounding box，这和上面提取单个物体的方法按理是相同的。

$$P_{ij} = \{p_k | p_k \in (\mathcal{B}^i \bigcup \mathcal{B}^j)\}_{k=1,|P|} \tag{2}$$

具体实施时分别提取单个实例的点云和两个相邻实例的点云送入 ObjPointNet 和 RelPointNet 中。

3.1.3 消息传递模块——GCN

我们将提取的特征以关系三元组（subject，predicate，object）的形式排列在图结构中 $(\phi_{s,ij}, \phi_{p,ij}, \phi_{o,ij})$，其中 ϕ_n 占据主语/宾语单位，而边缘特征 ϕ_r 占据谓语单位。接下来，采用图卷积网络（GCN）处理获得的三元组。GCN 的每个消息传递层均包含两个步骤。首先，将每个三元组输入至 MLP$g_1(\cdot)$ 中以进行信息传播。

$$(\psi_{s,ij}^{(l)}, \phi_{p,ij}^{(l+1)}, \psi_{o,ij}^{(l)}) = g_1(\phi_{s,ij}^{(l)}, \phi_{p,ij}^{(l)}, \phi_{o,ij}^{(l)}) \tag{3}$$

ψ 表示处理后的特征，s, p, o 分别表示 subject、predicate、object。之后，对于某个节点，在聚合步骤中，将来自该节点的所有有效连接的信号求平均，生成的节点特征将输入至另一个 MLP $g_2(\cdot)$ 中。

$$\rho_i^{(l)} = \frac{1}{|R_{i,s}| + |R_{i,o}|}(\sum_{j \in R_s} \psi_{s,ij}^{(l)} + \sum_{j \in R_o} \psi_{o,ij}^{(l)}) \tag{4}$$

$$\phi_i^{(l+1)} = \phi_i^{(l)} + g_2(\rho_i^{(l)}) \tag{5}$$

3.1.4 损失函数

模型的损失函数包含分类损失 L_{obj} 以及谓词分类损失 L_{pred}。

$$\mathcal{L}_{total} = \lambda_{obj}\mathcal{L}_{obj} + \mathcal{L}_{pred} \tag{6}$$

因为两个实体之间可能存在多个可行关系来描述它们之间的关系。例如一个椅子可以 front of 另一个椅子，也可以 same as 另一个椅子。因此 $\mathcal{L}_{\sqrt{\nabla}|\Gamma}$ 被定义为每一个类的二元交叉熵。为了解决 class imbalance 的问题，两个损失项都使用 focal loss 的格式

$$\mathcal{L}_{total} = -\alpha_t(1-p_t)^\gamma log p_t \tag{7}$$

其中 p_t 表示预测的值，γ 是超参数，α_t 表示在计算多类损失（$\mathcal{L}_{\text{CE}}$）时为类别的标准化的频率倒数，在计算 per-class loss（$\mathcal{L}_{\text{focal}}$）为固定的有边/无边的值。

3.2 EdgeOriented 模型

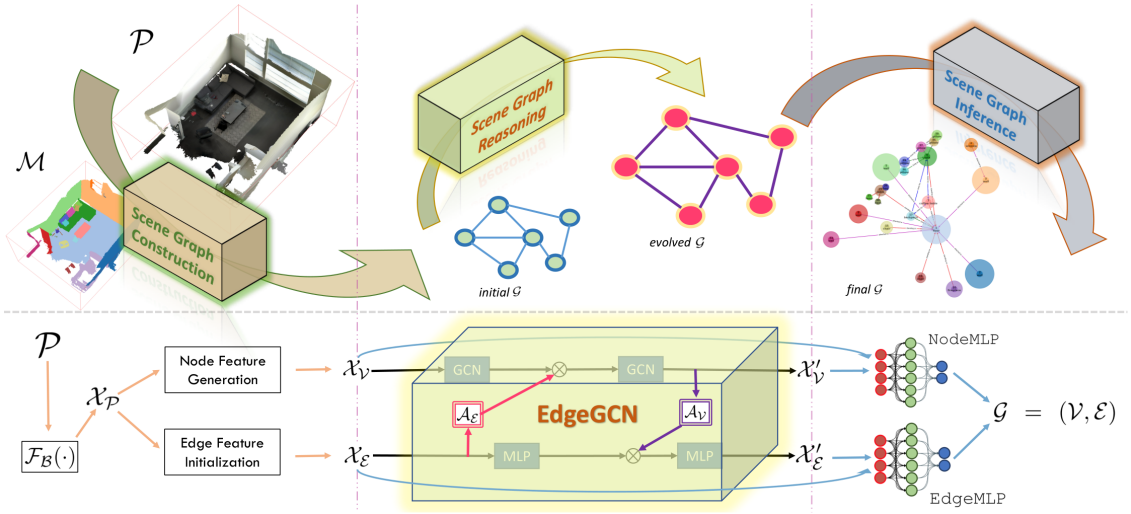

图 2: EdgeOriented 模型结构

3.2.1 方法概述

该模型将场景图生成分为了三个阶段：分别是场景图构造、推理和推断。构造阶段，删除了 3DSSG 中冗余的双特征提取函数，采用单特征提取函数简化模型；推理阶段，提出了基于边的图卷积网络（EdgeGCN），以利用多维度的边特征进行明确的关系建模，同时探索节点和边缘之间的两个相关双重互动机制，以实现场景图表示的独立演化。同时该论文改进了 3DSSG 数据集，通过对物体和边重新标签的方法减少了物体和边的类别，提出了 3DSSG-O27R16，并探究了其在该数据集上的效果。

3.2.2 特征提取模块

本文和 3DSSG 中使用两个分开的 backbone 来获取独立的物体和特定关系特征不同，本文通过共享一个单一的 backbone 来减少场景理解中的冗余，表示为 $\mathcal{F}_{\mathcal{B}}(\cdot)$ 来从 $\mathcal{P} \in \mathcal{R}^{\mathcal{N}\times\mathcal{C}_{\text{in}}}$ 中获取点层面的特征 $\mathcal{X}_{\mathcal{P}} \in \mathcal{R}^{\mathcal{N}\times\mathcal{C}_{\text{out}}}$，其中 $\mathcal{C}_{\text{in}}$ 和 $\mathcal{C}_{\text{out}}$ 表示输入点相应的特征维度以及经过特征获取后的点的特征维度。$\mathcal{X}_{\mathcal{P}}$ 被进一步传播，以方便对图中 m 个节点和 m^2 个边的表征进行初始建模。$\mathcal{X}_{\mathcal{P}}$ 分别被表示为 $\mathcal{X}_{\mathcal{V}} \in \mathcal{R}^{\mathcal{N}\times\mathcal{C}_{\text{node}}}$ 和 $\mathcal{X}_{\mathcal{E}} \in \mathcal{R}^{\updownarrow\times\updownarrow\times\mathcal{C}_{\text{edge}}}$。这里的 $\mathcal{F}_{\mathcal{B}}(\cdot)$ 在实际使用时用的是 PointNet 模型和 DGCNN 模型进行特征提取。

节点特征生成 对称池化函数 $\}(\cdot)$，和 class-agnostic 的 point-to-instance 指标 $\mathcal{M} \in \{\infty, ..., \updownarrow\}^{\mathcal{N}}$ 一起从点的特征 $\mathcal{X}_{\mathcal{P}}$ 来生成场景图中每一个物体 i 实例特征 $\mathcal{X}_{\mathcal{V}_i} \in \mathcal{R}^{\infty\times\mathcal{C}_{\text{node}}}$，这个过程可以表示为

$$\mathcal{X}_{v_i} = g(\{\delta(\mathcal{M}_k, i)\}_{k=1,...,\mathcal{N}}) \tag{8}$$

其中 $\delta(\cdot,\cdot)$ 表示换标函数，具体为括号内二者相等为 1，不同为 0，这里应该起到掩码的作用，用于表示属于结点 $\mathcal{V}_i$ 的点云中的点。最终可以生成所有节点的特征 $\mathcal{X}_{\mathcal{V}}$

边特征生成 和别的模型将对象间的结构关系重新表述为特殊类型的节点不同，该模型的框架会将这种信息表示为多维的边 $\mathcal{X}_{\mathcal{E}}$，$\mathcal{X}_{\mathcal{E}_{(\rangle,|)}} = (\mathcal{X}_{\mathcal{V}_\rangle} + (\mathcal{X}_{\mathcal{V}_|} - \mathcal{X}_{\mathcal{V}_\rangle}))$ 用于表示初始的边特征。

3.2.3 EdgeGCN

以前的场景图工作大多将边缘预测作为从节点表征学习中获得的副产品，这可能会忽视了节点和边的在 SGG 中潜在能力。然而本模型会将节点和边同等对待，作为成对的联合表示。因此，我们给每个人分配一个专属的学习分支，并研究图推理技术对其特征表示的增强。

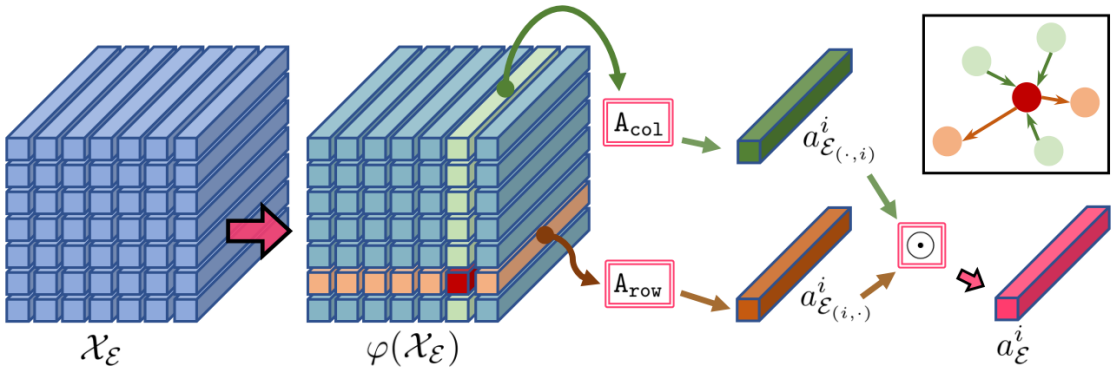

图 3: 成对边注意力

用于节点更新的成对边缘注意力机制 该注意力机制的目标是学习到 $\mathcal{A}_{\mathcal{E}} \in \mathcal{R}^{\Updownarrow \times \mathcal{C}'_{\wr\sqcap}}$，该 mask 是根据边特征 $\mathcal{X}_{\mathcal{E}}$ 决定的。为了使用上 $\mathcal{X}_{\mathcal{E}}$ 中的方向信息，我们在计算其边迭代向量 $\dashv^{\rangle}_{\mathcal{E}} \in \mathcal{R}^{\Updownarrow \times \mathcal{C}'_{\wr\sqcap}}$ 时会同时考虑其作为 source 和 target 的情况。当 $\mathcal{V}_\rangle$ 作为源头和目标时，边迭代特征会被分表表示成 $\dashv^{\rangle}_{\mathcal{E}_{(\rangle,\cdot)}}$ 和 $\dashv^{\rangle}_{\mathcal{E}_{(\cdot,\square\rangle)}}$

$$\dashv^{\rangle}_{\mathcal{E}_{(\rangle,\cdot)}} = \mathbf{A}_{row}(\{W^T_{\varphi}\mathcal{X}_{\mathcal{E}_{(i,k)}} | \forall \mathcal{V}_k\}) \tag{9}$$

$$\dashv^{\rangle}_{\mathcal{E}_{(\rangle,\cdot)}} = \mathbf{A}_{col}(\{W^T_{\varphi}\mathcal{X}_{\mathcal{E}_{(k,i)}} | \forall \mathcal{V}_k\}) \tag{10}$$

其中 $\mathcal{W}_{\varphi} \in \mathcal{R}^{\mathcal{C}_{\urcorner\sqcap\urcorner} \times \mathcal{C}'_{\wr\sqcap}}$ 是一个可以训练的转换矩阵，用于将 $\mathcal{X}_{\mathcal{E}_{(\cdot,\cdot)}} \in \mathcal{R}^{\Updownarrow \times \mathcal{C}_{\urcorner\sqcap\urcorner}}$ 的特征维度映射到 C'_{node}，然后 $A_{row}(\cdot)$ 和 $A_{col}(\cdot)$ 表示在每一个特征维度上的信息聚合函数。因此边迭代向量 $\dashv^{\rangle}_{\mathcal{E}} \in \mathcal{R}^{\Updownarrow \times \mathcal{C}'_{\wr\sqcap}}$ 最终的值可以表示 $\dashv^{i}_{\mathcal{E}} = \sigma(\dashv^{i}_{\mathcal{E}_{(\rangle,\cdot)}} \odot \dashv^{i}_{\mathcal{E}_{(\cdot,\rangle)}})$，其中 $\odot$ 表示点乘，σ 表示 sigmoid 激活函数。之后就可以得到场景图场景图 $\mathcal{G}$ 上的邻接矩阵 $\mathcal{A}_{\mathcal{G}}$，因此场景图结点特征表示 $\mathcal{X}'_{\mathcal{V}}$ 为：

$$\mathcal{X}'_{\mathcal{V}} = f(\hat{A}_{\mathcal{G}}(f(\hat{A}_{\mathcal{G}}\mathcal{X}_{\mathcal{V}}W_{G1}) \odot A_{\mathcal{E}})W_{G2}) \tag{11}$$

其中 $\mathcal{A}_{\mathcal{E}}$ 表示边驱动的交互 score，f 表示非线性激活函数，$\widehat{A}_{\mathcal{G}} = \mathcal{A}_{\mathcal{G}} + I$，$W_{G1} \in \mathcal{R}^{\mathcal{C}_{\wr\sqcap} \times \mathcal{C}'_{\wr\sqcap}}$，$W_{G2} \in \mathcal{R}^{\mathcal{C}'_{\wr\sqcap} \times \mathcal{C}_{\wr\sqcap}}$，因此特征维度变换为 $\mathcal{C}_{\wr\sqcap} \to \mathcal{C}'_{\wr\sqcap} \to \mathcal{C}_{\wr\sqcap}$。这里可以和 GCN 的公式做对比，发现差别主要是点乘上 $\mathcal{A}_{\mathcal{E}}$ 其它部分都相同。也就是说在原本的 GCN 的基础上加上了边的特征信息。在实际使用时，$\mathcal{C}_{\wr\sqcap} = 2 \times C'_{node} = 256$

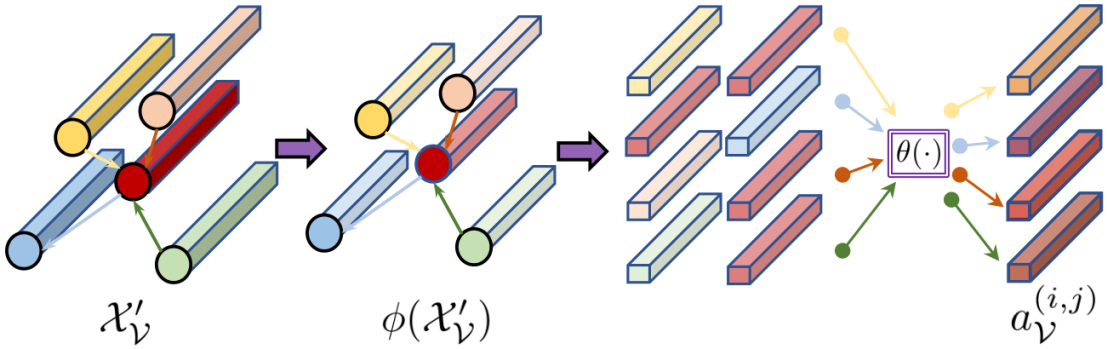

图 4: 成对边注意力

用于边更新的成对节点注意力机制 和成对边注意力机制中相同，本模块的目标是学习到点作为 source 和 target 时的多维注意力 mask$\mathcal{A}_{\mathcal{V}} \in \mathcal{R}^{\updownarrow\times\updownarrow\times C'_{\upharpoonright\upharpoondown\upharpoonright}}$，因此对于给定的边 $\mathcal{E}_{(),|)}$，最终产生一个边缘节点互动的驱动得分 $a_{\mathcal{V}}^{(i,j)} \in R^{1\times1\times C'_{edge}}$, 双加号 $++$ 文中是根据边进行对应特征拼接。

$$a_{\mathcal{V}}^{(i,j)} = \sigma(W_\theta^T f(W_\phi^T \mathcal{X}'_{V_i} + + W_\phi^T \mathcal{X}'_{V_j})) \tag{12}$$

之后便可以将节点特征和边特征相融合来提高边特征的表示能力

$$\mathcal{X}'_{\mathcal{E}} = f(W_{FC2}^T(f(W_{FC1}^T \mathcal{X}_{\mathcal{E}}) \odot \mathcal{A}_{\mathcal{V}})) \tag{13}$$

边特征可由该方法进行学习，其中 $\mathcal{A}_{\mathcal{V}}$ 表示的就是上面学习到的 attention mask，最终同样让边的特征维度变化为：$\mathcal{C}_{\upharpoonright\upharpoondown\upharpoonright} \to \mathcal{C}'_{\upharpoonright\upharpoondown\upharpoonright} \to \mathcal{C}_{\upharpoonright\upharpoondown\upharpoonright}$

对于上一阶段的输出节点特征 $\mathcal{X}'_{\mathcal{V}}$ 和边特征 $\mathcal{X}'_{\mathcal{E}}$，使用 NodeMLP 和 EdgeMLP，这两个 MLP 都是由两层全连接层组成，但是参数不共享。损失函数使用的是 muti-class cross entropy $\mathcal{L}_{\backslash\upharpoonright} + \mathcal{L}_{\upharpoonright\upharpoondown\upharpoonright}$

4 复现细节

4.1 与已有开源代码对比

本次复现的两个模型——3DSSG 和 EdgeOriented 模型，其中 3DSSG 模型官方没有给出源码，仅给出了数据集介绍。因此复现过程参照了一位网友的复现代码。但是，该代码存在代码运行不通、结果不收敛等问题，最后经过排查发现其对于损失函数的复现存在问题，且涵盖一部分代码错误。经过调试和修改之后，将原本的单卡训练提升为多卡训练，且模型可以正常收敛并基本达到论文水平。同时，原本的模型中采用的是 PointNet 特征提取函数，为轻量化模型，也为了更好的模型效果，将原本的 PointNet 替换成 PointNext，最终发现其对于模型效果也有一定的提升。

EdgeOriented 同样没有完整的官方源码，仅仅提供了一部分核心代码，因此从数据预处理到训练函数搭建、损失函数的撰写都是本人完成。同时还给模型加入了可视化函数来方便观察结果。

4.2 实验环境搭建

采用的是 pytorch 的模型框架，主要的实验环境如下：opencv-python trimesh tensorboardX easydict tqdm h5py matplotlib numpy plyfile torch==1.10.0 tensorboard

4.3 创新点

本次实验的创新点创新点相对较少，主要是对现有工作的复现。创新之处主要在更换了模型的特征提取网络，由原本的 PointNet 网络更换为 PointNext 网络，模型的特征提取效果有一定的提升。同时，由于模型细节文中并没有很好的介绍，因此加入了残差网络的结构来有效的防止模型的过拟合。

5 实验结果分析

5.1 3DSSG 模型结果

下表是在 3DSSG 数据集上两个实验的实验结果，由于该领域是全新的领域，在指标计算上尚未有前人工作可以借鉴。加之文章并没有给出具体的指标计算方法，因此只能参照二维场景图生成的指标计算模式。最后根据文章内容可以分为三个下游任务，分别是三元组预测（relationship prediction）、物体类别预测（object class prediction）和关系预测（predicate prediction）。其中三元组预测是仅给定点云和实例分割，联合预测物体的类别和关系；物体类别预测则是给定物体的实例分割，预测物体的类别；关系预测则是给定物体的类别标签和分割，最后预测物体之间的关系。R@k 则是表示在前 k 个有把握的关系中预测正确的关系的比率，但是由于并没有给出官方源码，所以对于如何使用物体类别标签仍然存在问题，因此最后计算的指标相较于原文有一定的差别。最后的复现结果如下图。

	Relationship Prediction		Object Class Prediction		Predicate Prediction	
Method	R@50	R@100	R@5	R@10	R@3	R@5
3DSSG-GT	40	66	68	78	89	93
3DSSG	53.51	54.66	64.10	72.14	73.21	73.38
EdgeOriented	60.23	76.88	41.38	55.63	78.53	85.74

表 1: **3DSSG 数据集结果**

由表中数据可知，本次复现在没有调参的情况下基本达到了原文的水平，同时测试了 EdgeOriented 模型在 3DSSG 数据集上的表现，证明了其相较于基础模型的有效性，并展现了其在三元组预测上的出色表现。

5.2 EdgeOriented 模型结果

原文中 EdgeOriented 模型并没有在 3DSSG 数据集上进行测试，而是提出了一个新的数据集，该数据集和 3DSSG 数据集一样基于 3RScan 数据集产生，但节点类别数量和边的类别数量都要远少于 3RScan。最后在该数据集上产生的结果如下：

	Relationship Prediction		Object Class Prediction		Predicate Prediction	
Method	R@50	R@100	R@5	R@10	F1@3	F1@5
EdgeOriented-GT	39.91	48.68	90.70	97.58	78.88	90.86
EdgeOriented	18.84	32.07	47.61	72.56	70.42	84.03

表 2: **EdgeOriented 数据集结果**

从结果来看，模型结果并不是很好。分析原因如下：首先是因为新的数据集是按照每一个场景进行划分，一个场景对应一个点云。而场景间物体的数量和关系的数量存在较大的区别（最少的只有三到四个物体，但是最多的有 103 个物体）。但在预处理的采样的过程中，每一个场景都会被采样到同样数量的点，这会导致一些物体的点数较少，可能会损失大量的特征。

其次是模型本身具有良好的性能，但因为指标计算有误导致呈现出的模型效果不好。在计算三元组的 R@50 和 R@100 中，初始的计算指标的公式是：对于 R@N，从所有预测的三元组中取出 N 个三元组，判断每一个三元组是否属于真实的 Triplet，然后用属于的值除以总的真实 Triplet 的数量。这

种指标计算方法对于 3DSSG 上是可行的。因为 3DSSG 将大场景划分成了一个个小场景，每一个场景最多 9 个物体，72 个关系，同时两个物体之间可能存在多个关系类型，且都有标注（eg：人和单车之间可能会同时有 person ride bike 和 person on bike）。但是在 EdgeOriented 中，并没有对场景进行划分，这就导致有一些场景非常的大，此时真实的 Triplet 数量会远远大于 100，因此如果仍然用除以真实 Triplet 的数量的方法会导致 Recall 值过低，即便模型预测效果好结果也会很低。并且该数据集中同一对主客体之间并不存在多种 relation，因此会对结果产生一定的影响。

6 总结与展望

本次实验的不足之处主要在复现的模型相对简单，且缺少创新性的改进。这也和目前 3D 场景图生成仍然处于起步阶段有关，从数据集到模型都相对较少有关。同时从复现结果来看，离论文结果尚有一定差距，这主要是模型未调参以及研究方向处于起步阶段，对模型细节不清晰导致的。但是 3D 场景图生成具有巨大的进步空间，未来可以参考 2D 场景图生成的内容对 3D 方面的模型进行优化创新，同时也可以根据 3D 数据结构的特性进一步探究更符合 3D 数据结构的模型。

参考文献

[1] ZHAN Y, YU J, YU T, et al. On exploring undetermined relationships for visual relationship detection [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 5128-5137.

[2] ZHAN Y, YU J, YU T, et al. Multi-task compositional network for visual relationship detection[J]. International Journal of Computer Vision, 2020, 128(8): 2146-2165.

[3] ZHANG H, KYAW Z, CHANG S F, et al. Visual translation embedding network for visual relation detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 5532-5540.

[4] ZHANG J, ELHOSEINY M, COHEN S, et al. Relationship proposal networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 5678-5686.

[5] ZHANG J, SHIH K, TAO A, et al. An interpretable model for scene graph generation[J]. arXiv preprint arXiv:1811.09543, 2018.

[6] WOO S, KIM D, CHO D, et al. Linknet: Relational embedding for scene graph[J]. Advances in neural information processing systems, 2018, 31.

[7] YIN G, SHENG L, LIU B, et al. Zoom-net: Mining deep feature interactions for visual relationship recognition[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 322-338.

[8] KOLESNIKOV A, KUZNETSOVA A, LAMPERT C, et al. Detecting visual relationships using box attention[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019: 0–0.

[9] XU D, ZHU Y, CHOY C B, et al. Scene graph generation by iterative message passing[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 5410-5419.

[10] ZELLERS R, YATSKAR M, THOMSON S, et al. Neural motifs: Scene graph parsing with global context [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 5831-5840.

[11] TANG K, ZHANG H, WU B, et al. Learning to compose dynamic tree structures for visual contexts[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 6619-6628.

[12] CHEN Y, WANG Y, ZHANG Y, et al. Panet: A context based predicate association network for scene graph generation[C]//2019 IEEE International Conference on Multimedia and Expo (ICME). 2019: 508-513.

[13] GAO W, ZHU Y, ZHANG W, et al. A hierarchical recurrent approach to predict scene graphs from a visual-attention-oriented perspective[J]. Computational Intelligence, 2019, 35(3): 496-516.

[14] TENG Y, WANG L. Structured sparse r-cnn for direct scene graph generation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 19437-19446.

[15] YANG J, LU J, LEE S, et al. Graph r-cnn for scene graph generation[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 670-685.

[16] HERZIG R, RABOH M, CHECHIK G, et al. Mapping images to scene graphs with permutation-invariant structured prediction[J]. Advances in Neural Information Processing Systems, 2018, 31.

[17] YAN Y, HASHEMI M, SWERSKY K, et al. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks[J]. arXiv preprint arXiv:2102.06462, 2021.

[18] LIN X, DING C, ZHAN Y, et al. HL-Net: Heterophily Learning Network for Scene Graph Generation [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 19476-19485.

[19] LECLERC S, SMISTAD E, GRENIER T, et al. RU-Net: A refining segmentation network for 2D echocardiography[C]//2019 IEEE International Ultrasonics Symposium (IUS). 2019: 1160-1163.

[20] ARMENI I, HE Z Y, GWAK J, et al. 3d scene graph: A structure for unified semantics, 3d space, and camera[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 5664-5673.

[21] KIM U H, PARK J M, SONG T J, et al. 3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents[J]. IEEE transactions on cybernetics, 2019, 50(12): 4921-4933.

[22] SHIN D, KIM I. Deep neural network-based scene graph generation for 3d simulated indoor environments[J]. KIPS Transactions on Software and Data Engineering, 2019, 8(5): 205-212.

[23] ROSINOL A, GUPTA A, ABATE M, et al. 3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans[J]. arXiv preprint arXiv:2002.06289, 2020.

[24] HUGHES N, CHANG Y, CARLONE L. Hydra: a real-time spatial perception system for 3d scene graph construction and optimization[J]., 2022.

[25] WU S C, WALD J, TATENO K, et al. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 7515-7525.

[26] WALD J, DHAMO H, NAVAB N, et al. Learning 3d semantic scene graphs from 3d indoor reconstructions[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 3961-3970.

[27] WU F, YAN F, SHI W, et al. 3D scene graph prediction from point clouds[J]. Virtual Reality & Intelligent Hardware, 2022, 4(1): 76-88.

[28] ZHANG C, YU J, SONG Y, et al. Exploiting edge-oriented reasoning for 3d point-based scene graph analysis[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 9705-9715.

[29] ZHANG S, HAO A, QIN H, et al. Knowledge-inspired 3D Scene Graph Prediction in Point Cloud[J]. Advances in Neural Information Processing Systems, 2021, 34: 18620-18632.