中国科学: 信息科学

SCIENTIA SINICA Informationis

论文

以物体关系为启发的高效 3D 物体检测方法

吴宇寰[1], 张达[2], 刘云[1], 张乐[3], 程明明[1*]

1. 南开大学计算机学院, 天津市 300350
2. 加利福尼亚大学圣巴巴拉分校, 加利福尼亚 93106
3. 电子科技大学信息与通信工程学院, 成都市 611731
* 通信作者. E-mail: cmm@nankai.edu.cn

基金资助：国家自然科学基金杰出青年科学基金项目 (编号：62225604)

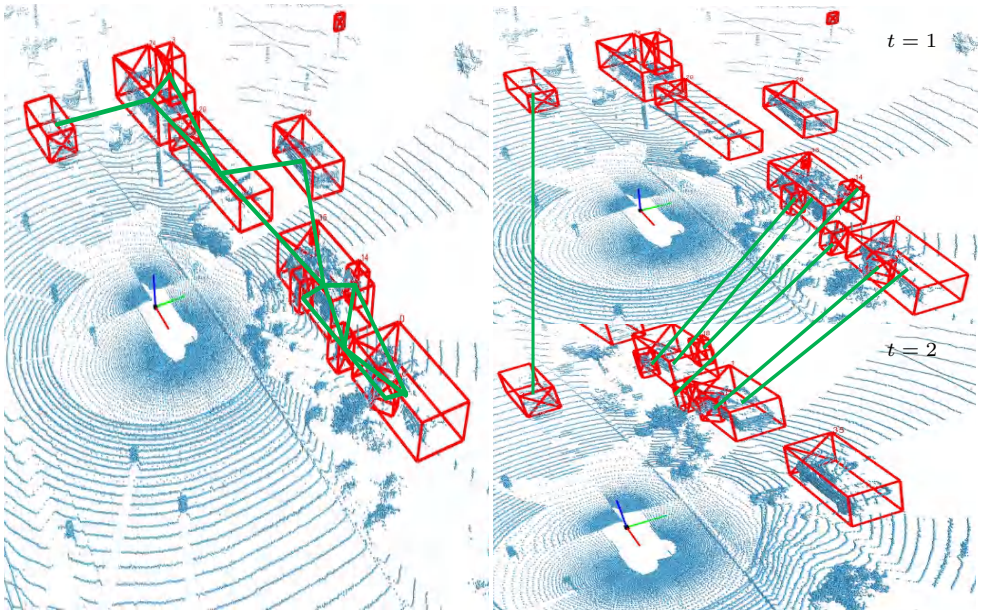
摘要 目前, 基于激光雷达的高效 3D 物体检测框架在利用物体关系方面存在不足, 然而这些关系在空间和时间维度上是自然存在的. 受此启发, 本文提出了一种以物体关系为启发的简单高效的二阶段检测器 Ret3D. 该方法的核心在于利用本文所提出的帧内关系模块和帧间关系模块, 以捕捉空间和时间维度上的关系. 具体来说, 帧内关系模块将当前帧的物体封装为稀疏图, 通过高效的信息传递来优化物体特征. 另一方面, 帧间关系模块动态地密集连接每个物体与其跟踪序列中的其他物体, 并利用这种时间信息, 通过轻量级的 Transformer 网络进一步增强其特征表达能力. 本文在通用的中心基或锚点基检测器的基础上实现了 Ret3D, 并在 Waymo 公开数据集上进行了评估. Ret3D 在额外开销几乎可以忽略不计的情况下, 取得了最佳性能, 在车辆位置检测的一级和二级难度下的 mAPH 指标上分别比近期最强的著名方法高出 2.9% 和 3.2%.

关键词 3D 物体检测, 物体关系, 自动驾驶

1 引言

3D 物体检测的目标是识别大型场景中的车辆、行人、自行车及其他关键特征, 是自动驾驶感知系统的关键模块之一[1]. 在自动驾驶场景下的 3D 物体检测的发展中, 基于激光雷达的方法[2–5] 相较于基于单目或多视角图像的方法[6–8] 具有显著优势, 这是因为激光雷达信号通过点云可以精确地提供远近距离的深度信息. 相对而言, 相机受到二维视角限制, 对远处的感知存在明显不足.

近年来, 2D 物体检测相关研究[9–12] 已证明物体关系能够显著提升检测精度. 而在自动驾驶场景的 3D 物体检测中, 3D 物体的位置和几何特征可以为场景理解和准确的物体识别提供丰富的上下文和结构信息. 然而, 现有的激光雷达 3D 物体检测框架对物体关系的利用仍显不足. 具体而言, 大多数工作仅通过精心设计的卷积神经网络 (Convolutional Neural Networks, CNNs)[4, 13, 14] 或视觉 Transformers[15] 架构隐式地探索物体关系. 还有一些工作[16] 利用长点云序列的方式来提升离线检测性能, 但这显著增加了计算成本. 本文认为, 目前还缺乏一种能够显式且高效利用物体关系来提升 3D 物体检测性能的系统方法.



(a) Intra-frame object relations (b) Inter-frame object relations

图 1 帧内 (a) 及帧间 (b) 关系样例图. 绿色直线代表物体关系. 为简化说明, 该图仅使用长度为 2 的短序列来展示帧间物体关系.

Figure 1 An example of intra-frame (a) and inter-frame (b) object relations. Green lines indicate object relations. For simplicity, only a short sequence with a length of 2 is used to illustrate inter-frame object relations. Best viewed in color.

在实际应用中, 因为激光雷达帧是自然的时间序列, 所以本文考虑帧内和帧间物体关系 (如图 1 所示): 帧内物体关系为当前帧内 3D 物体之间的关系; 帧间物体关系指的是同一 3D 物体在长时间激光雷达序列中不同帧之间的空间和时间关系.

对于帧内物体关系建模来说, 一个简单的方法是将每个物体与同一帧中的所有其他物体进行密集连接. 虽然这种方式可以提升检测性能, 但这种密集连接的图包含大量冗余信息 (例如一辆车辆与远在几十米外的行人几乎没有任何关联), 且不可避免地增加了计算开销. 因此, 利用物体的物理位置等实用的先验信息来构建稀疏图可以有效避免冗余信息, 进一步提高检测效率. 而对于帧间物体关系来说, 物体可能出现在长时间序列的每一帧中. 由于聚合点云和检测结果来处理长序列非常耗时[16], 本文提出利用物体级别的信息 (如位置、朝向和速度) 来建模帧间物体关系, 从而在降低计算成本的同时保持 3D 物体检测的精度.

基于上述观察, 本文引入了一个时空框架, 即以物体关系为启发的 3D 物体检测方法 (Ret3D). 首先, 为了构建帧内物体关系, 先构建出一个稀疏无向图: 在该图中, 每个物体被视为一个节点, 节点之间的空间距离决定是否存在相应的边. 这样的稀疏图能够通过高效的消息传递机制来迭代优化物体的关系和特征. 在此过程中, 由于图的稀疏性和每帧物体特征的低维度, 只需很少的计算开销 (<0.1% 的基线检测器计算量). 最终, 物体的位置可以通过优化后的物体特征进行修正. 对于帧间物体关系, 本文引入了一种基于 Transformer 的检测器, 它通过跟踪的物体序列高效建模每个物体的密集帧间关系, 从而利用不同时间戳的检测结果. 需要强调的是, Ret3D 并非直接从点云中提取目标, 而是在一阶段检测器提供的结果基础上, 通过显式建模物体间的时空关系实现进一步优化.

遵循先前的工作[5, 17], 本文在著名的的 Waymo 公开数据集[1] 上开展了大量实验来验证 Ret3D

的性能. 实验结果表明, 帧内和帧间物体关系对检测性能的提升具有显著作用. 以 CenterPoint [5] 作为基线检测器来说, Ret3D 在一级和二级难度设置下显著优于近期的最佳方法. 因此, 本文关于帧内和帧间物体关系建模的思路为 3D 物体检测提供了新的研究视角, 能够启发未来的研究.

总体来说, 本文贡献总结如下:

- 利用显式且高效地学习帧内和帧间物体关系的方式, 大幅提升了自动驾驶中的高效 3D 物体检测的精度.
- 提出了帧内关系模块 (Intra-Frame Relation Module, IntraRM), 通过消息传递在同一激光雷达帧内构建稀疏物体图来优化每个物体的特征.
- 提出了帧间关系模块 (Inter-Frame Relation Module, InterRM), 通过轻量的 Transformer 网络动态连接每个物体在物体序列中的帧间关系来进一步优化物体特征.

2 相关工作

2.1 基于激光雷达的 3D 物体检测

3D 物体检测是自动驾驶感知系统中的核心问题[18–20]. 与只有 4 个自由度的 2D 物体检测不同, 3D 物体检测至少有 7 个自由度, 包括 3D 中心位置 (x, y, z)、长、宽、高和朝向角. 因此, 3D 物体检测需要非常敏感的深度先验, 而这可以通过激光雷达信号轻松获取. 因此, 近年来基于激光雷达的 3D 物体检测[2–5] 表现优于其他方法[6–8].

通常在自动驾驶场景下, 每帧激光雷达数据包含数十万个点[1, 18]. 直接在点云中搜索 3D 物体是非常具有挑战性的, 因为点云稀疏、不规则且无序. 常见的做法是将不规则的点云转换为规则网格的特征. 例如, Vote3Deep[21] 通过基于特征的投票机制将点云转换为规则的体素, 以实现实时处理速度. VoxelNet[22] 进一步提出通过 PointNet[23] 获取更全面的体素特征, 并使用3D 卷积进行特征提取. SECOND[2] 提出采用 3D 稀疏卷积, 大幅提升了骨干特征的提取速度. 其他一些工作[3, 24] 则通过将点云在鸟瞰视图 (bird eye view, BEV) 下的固定编码或 PointNet[23] 转换为柱状特征, 然后利用更快的 2D CNN 进行特征提取. 受[25, 26] 的启发, PillarOD[27] 和 CenterPoint[5] 引入了无锚框 (anchor-free) 的 3D 物体检测框架, 取代了传统的基于锚框的预测机制, 采用柱状中心点进行无锚框预测. HEDNet[28] 提出了一种新型的分层编码-解码架构, 用于在稀疏点云中捕获长距离依赖关系, 并解决了现有方法中信息交换受限的问题. FSD[29] 提出了完全稀疏的检测框架, 减少了大量冗余计算, 大幅提升了算法的高效性. 在 FSD[29] 的基础上, VoxelNeXt[30] 在所有模块中都采用了稀疏的体素特征预测, 显著地提升了 3D 物体检测与跟踪的性能.

2.2 二阶段 3D 物体检测

最近, 二阶段 3D 物体检测器因其强大的兼容性和在感兴趣区域 (region of interest, RoI) 上的精细化处理而变得流行. 许多方法[4, 5, 17, 31–34] 将 2D R-CNN 风格的框架[35] 适配于鸟瞰图中的 3D 物体检测. 首先, 通过区域建议网络 (region proposal network, RPN) 生成物体候选区域. 然后, 额外的回归头独立地对每个候选区域进行打分和校正.

然而, 直接应用上述策略并不理想, 因为鸟瞰图特征本身也很稀疏, 在此过程中信息损失是不

可避免的. 因此, 许多方法[4, 31, 32, 34, 36, 37] 利用点云或体素特征来获取更丰富的空间信息. 例如, Point R-CNN[32] 利用了每个感兴趣区域中的原始点特征. PV-RCNN[4] 进一步提出了点——体素的抽象集合, 通过对每个感兴趣区域中的点和体素特征进行编码, 丰富了空间信息. LiDAR R-CNN[34] 利用初始点云来优化检测结果, 即将检测区域内及其周围的点云输入到 PointNet[23] 中. 二阶段 CenterPoint[5] 检测器仅使用每个物体中的 5 个 BEV 特征点进一步优化了检测结果.

2.3 用于 3D 感知的图网络

由于图网络在几何特征提取中的有效性, 其在 3D 感知任务中非常流行. 近年来, 许多工作利用图网络进一步增强点云[38, 39]、体素[40]、区域[41, 42] 或鸟瞰图[43] 特征. 例如, Wang 等人[38] 提出了用于室内 3D 分析的动态图网络, 其中每个点被视为一个节点, 并且在每次图迭代中动态更新图. Zarzar 等人[41] 提出了一个基于图网络的物体检测器, 通过建议框的图来过滤低置信度的建议框. Shi 等人[44] 提出了 Point GNN, 它将初始点云编码为一个大型稀疏图, 并通过自动注册机制来同时检测多个物体. Wang 等人[43] 利用基于查询的图网络进一步增强稀疏鸟瞰图特征.

从另一个角度来看, 本文提出了基于稀疏图网络的帧内关系模块来建模同一激光雷达帧内的物体关系. 该模块通过图网络显式学习物体关系特征, 并优化每个物体的检测结果. 所提出的帧内关系模块由于物体特征的简单表示和物体关系图的稀疏性, 在计算上非常高效. 帧内关系模块可以轻松集成到现代高效 3D 物体检测器中.

2.4 用于 3D 物体检测的 Transformer

Transformer 最初是自然语言处理 (natural langauage processing, NLP) 领域的主导工具, 因为其擅长通过多头自注意力 (multi-head self-attention, MHSA) 进行全局关系建模. 由于全局关系对于视觉任务同样重要, Carion 等人[45] 提出了 DETR, 将 Transformer 适配于 2D 物体检测, 大大简化了检测流程. 受 DETR[45] 的启发, 许多工作[46–51] 将 Transformer 适配于视觉任务, 证明其在大多数视觉任务 (如图像分类[46]、物体检测[52] 和语义分割[49]) 中能够超越 CNNs[53, 54]. 近年来, 最近 Transformer 也在 3D 物体检测任务中取得了巨大成功. 例如, Voxel Transformer[15] 引入了一个用于体素特征提取的 3D 稀疏 Transformer, 取代了传统的 3D 稀疏 CNN. CT3D[37] 提取每个建议框中的点特征, 然后单独使用基于通道的 Transformer 进行建议框优化. 基于 PointNet++[23], Liu 等人[55] 引入了一个无锚框框架, 通过 Transformer 进行类似于 DETR[45] 的集合到集合的框预测. SST[56] 提出了使用单步长稀疏 Transformer 进行 3D 物体检测, 为后续工作奠定了基础. DSVT[57] 在 SST[56] 的基础上, 通过引入动态稀疏窗口注意力机制, 进一步提升了检测性能. FlatFormer[58] 则通过扁平化窗口注意力机制, 显著减少了结构化和填充的计算开销, 在保持性能的同时大幅降低了计算复杂度. Chen 等人[59] 提出了针对难检测实例的探测机制, 使得检测方法更聚焦于难以检测的实例.

与其他方法不同, 本文提出了帧间关系模块, 该模块通过跟踪的物体序列高效提取帧间物体关系. 每个检测到的物体通过其历史位置、尺寸和运动的强先验进行优化.

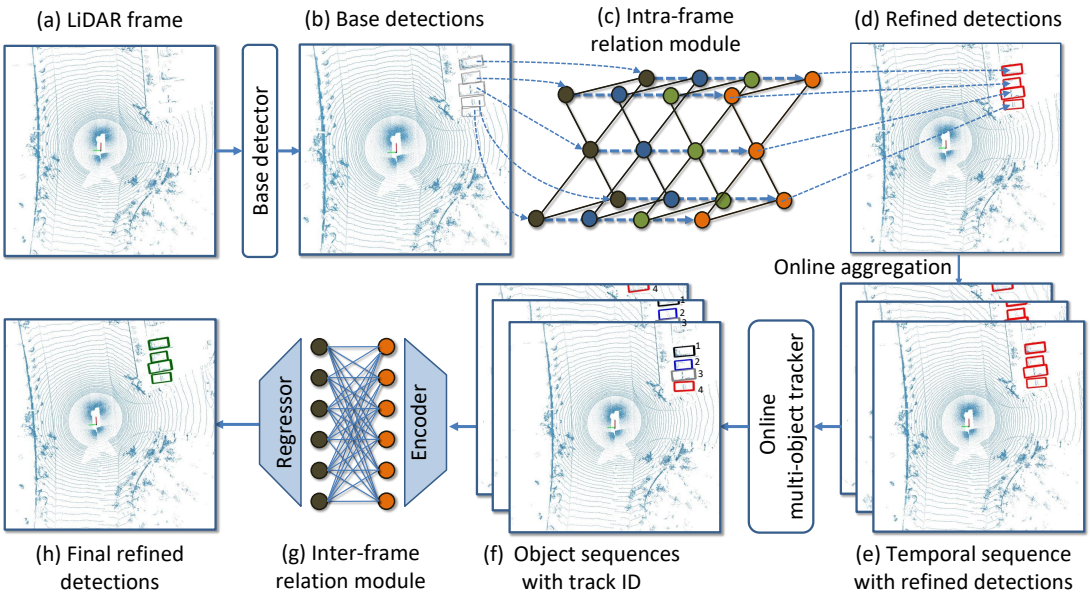


图 2 Ret3D 的整体流程图. Ret3D 是一个二阶段检测器, 用于高效地优化一阶段检测器的检测结果. 它由两个部分组成, 分别是帧内关系 (c) 和帧间关系 (g) 模块, 用于通过帧内和帧间物体关系优化检测结果.

Figure 2 The pipeline of Ret3D. Ret3D is a two-stage detector that refines the detection results of one-stage detectors efficiently. Ret3D consists of two parts, IntraRM (c) and InterRM (g), for refining detection results using intra-frame and inter-frame object relations, respectively.

2.5 用于物体关系建模的物体检测方法

近年来, 物体关系建模成为 3D 物体检测和感知领域的重要研究方向. 例如, 在点云检测中, Object DGCNN[38] 提出了一种基于动态图的信息传递机制以优化 3D 目标检测; 3D-MAN[60] 通过多帧注意力机制有效聚合来自不同时间帧的特征; MPPNet[61] 利用原始点云和检测结果提取出代理点, 实现多帧特征编码和交互, 从而有效地处理长序列点云. 在通用图像感知领域, SELSA[62] 提出了序列级别语义聚合模块, 通过全局跨帧建模提升了视频物体检测的鲁棒性. 这些方法为高效建模物体关系提供了坚实的基础. 本文借鉴上述工作, 提出了帧内和帧间的稀疏图网络模块, 用于在 3D 感知领域高效建模物体关系. 具体而言, 帧内关系模块显式建模同一帧内的物体关系以优化检测结果; 帧间关系模块则通过历史轨迹追踪, 利用位置、尺寸及运动信息的强先验提升检测精度. 本文方法不仅结构简单且计算高效, 可轻松集成到现代 3D 目标检测器中, 并显著提升检测性能.

3 以物体关系为启发的二阶段检测器

该部分将介绍 Ret3D 的整体流程, 详见第 3.1 节. Ret3D 包含两个高效模块: 帧内关系模块 (帧内关系模块) 和帧间关系模块 (帧间关系模块), 分别在第 3.2 节和第 3.3 节中详细介绍. 最后, 本文在第 3.4 节中对 Ret3D 的每个模块的时间复杂度进行了分析.

3.1 整体流程

本文在图 2 中展示了 Ret3D 的整体流程. 给定点云 $\mathcal{P} = \{p_1, p_2, ..., p_M\}$, 其中包含 M 个点, 单阶段基线检测器将点云转换为规则的体素 (voxel) 或柱状体 (pillar) 用于进一步处理. 为了保持通用性并简化说明, 假设点云已被转换为规则体素.

单阶段基线检测器. 首先利用一个 3D 骨干网络从体素中提取规则的地图视图特征 $\mathcal{B} \in \mathbb{R}^{C \times H \times W}$, 其中 H 和 W 由初始体素大小和 3D 骨干网络的步幅决定, C 是地图视图特征 $\mathcal{B}$ 的通道数. 获取地图视图特征 $\mathcal{B}$ 后, 回归头会预测一组检测结果 $\mathcal{D}$, 其中包含每个检测到的物体的中心、大小、朝向和速度. 定义检测结果 $\mathcal{D}$ 为每个检测物体的基本特征. 同时, 根据所有检测结果的中心位置, 在地图视图特征 $\mathcal{B}$ 上裁剪出每个检测物体的特征向量 $\mathcal{O}$.

帧内关系模块. 为了建模帧内物体关系, 本文提出了帧内关系模块, 如图 2 (c) 所示. 在该模块中, 本文从单阶段基线检测器收集所有的基本特征. 此外, 为了有效建模每个物体, 本文还提取了每个物体的鸟瞰图特征向量, 该特征的鸟瞰图位置与每个检测物体的中心对应. 根据每个物体的空间位置构建一个稀疏图, 然后通过消息传递机制在图上优化物体的特征, 进一步得到优化后的检测结果. 详细信息请参考第 3.2 节.

帧间关系模块. 从另一个角度, 如图 2 (g) 所示, 帧间关系模块用于建模帧间物体关系, 通过跟踪的物体序列优化检测结果. 为了与实际应用保持一致, 仅使用历史帧而忽略未来帧. 给定物体序列, 利用一个轻量的 Transformer 进行特征提取, 它在捕捉全局关系方面高效且强大, 最终为每个物体回归出优化后的检测结果. 第 3.3 节将介绍该模块的更多细节.

可扩展性和适用性. Ret3D 框架具有良好的可扩展性和适用性: 1) 帧内和帧间关系模块均采用了模块化设计, 可以灵活地与不同的一阶段检测器集成. 此外, 这些模块也可以独立使用或组合使用, 为不同应用场景提供灵活的选择. 2) 本文提出的物体关系建模方法不依赖于特定的检测器架构或特征提取方式, 可以广泛应用于各类 3D 检测任务. 例如, 该方法可以扩展到室内场景理解、机器人导航等领域. 3) 通过稀疏图结构和轻量级 Transformer 的设计, 本方法在保持高性能的同时具有较低的计算开销, 适合实际应用场景的部署.

3.2 帧内关系模块

现有的二阶段方法仅通过点云[4, 37]、体素[40] 或鸟瞰图[5] 特征单独优化检测结果, 它们并没有考虑物体之间的关系, 而帧内物体关系对于优化检测结果也非常重要. 因此, 本文提出了一种高效的解决方案, 即新的帧内关系模块 (帧内关系模块). 通过帧内关系模块, 物体被连接为一个稀疏图, 因为密集图会包含大量冗余信息并引入较大的计算成本. 这样的定义确保了图中的边主要用于捕捉局部相关性, 而远距离物体由于物理上通常缺乏直接关联, 故不会被连接. 另一方面, 稀疏图仅包含重要的边, 允许图网络通过高效的消息传递优化每个节点特征. 该架构如图 2 (b)-(d) 所示.

对于每帧激光雷达输入, 基线检测器会先检测出 n 个物体, 其基本特征为 $\mathcal{D} = \{d_1, d_2, ..., d_n\}$, 对应的地图视图特征向量为 $\mathcal{O} = \{o_1, o_2, ..., o_n\}$. 在这些检测到的物体上构建一个稀疏图 $G = (V, E)$, 其中节点集 $V = \{v_1, v_2, ..., v_n\}$ 包含所有检测物体, 边集 E 连接节点对.

本文采用动态图网络[38] 作为消息传递的机制来迭代图中的节点特征, 具体如式 (2) 和式 (3) 所示. 假定节点 v_i 和 v_j 的初始特征分别为 x_i^0 和 x_j^0, 它们可以通过在检测结果和对应的地图视图特征向量的通道上的拼接来表示:

$$x_i^0 = \text{Concat}(d_i, o_i)\mathbf{W}^x, x_j^0 = \text{Concat}(d_j, o_j)\mathbf{W}^x, \tag{1}$$

其中 $x_i^0, x_j^0 \in \mathbb{R}^{C_x}$, $\mathbf{W}^x \in \mathbb{R}^{(C+T)\times C_x}$ 可以看作是线性层的权重, Concat 代表几个特征在通道上的拼接结果. T 是每个物体的基本特征长度. C_x 是每个节点的编码特征长度. 节点 v_i 和 v_j 之间的边特征 e_{ij} 可以计算为:

$$e_{ij}^0 = \mathcal{H}(x_j^0 - x_i^0, x_i^0), \tag{2}$$

其中 $\mathcal{H}(\cdot)$ 是提取边特征的非线性转换函数. 本文移除了距离超过 r 米的节点之间的边, r 设置为 2 米. 关于这个参数的消融实验见第 5 节.

获取所有边特征后, 应用通道最大池化来更新每个节点的特征:

$$x_i^1 = \max_{j\in\vartheta} e_{ij}^0, \tag{3}$$

其中 ϑ 是与第 i 个节点相连接的节点的索引集, x_i^1 表示该节点在一次更新后的优化特征, max 为最大池化操作.

整个过程是迭代进行的, 即迭代运行上述操作 m 次来获取每个节点的特征 x_i^m. 最终, 节点特征 x_i^{final} 是所有隐藏节点特征的通道拼接结果:

$$x_i^{final} = \text{Concat}(x_i^1, x_i^2, ..., x_i^m). \tag{4}$$

所有最终节点特征将被输入到回归头内. 该回归头由三个线性层组成, 分别预测优化后的物体位置、朝向和类别标签. 帧内关系模块的最终损失函数定义如下:

$$\mathcal{L}_{intra} = \mathcal{L}_{cls} + \lambda_1\mathcal{L}_{reg} + \lambda_2\mathcal{L}_{dir}, \tag{5}$$

其中 $\mathcal{L}_{cls}$ 是焦点损失[63], $\mathcal{L}_{reg}$ 和 $\mathcal{L}_{dir}$ 是用于回归物体位置和朝向的平滑 L1 损失[35]. λ_1 和 λ_2 为平衡权重.

3.3 帧间关系模块

最近一些方法[16, 64] 针对离线 3D 物体检测进行了研究, 旨在提高服务器端的自动标注精度. 这些方法虽然通过聚合长时间序列的点云提升了性能, 但也带来了巨大的计算开销, 因此不适用于在线应用 (如实时自动驾驶). 为了解决这个问题, 本方法设计了一种高效的帧间关系模块.

给定一个长时间序列的激光雷达点云数据, 单阶段基线检测器与帧内关系模块结合可以为每个单帧激光雷达数据提供优化的检测结果. 在此基础上, 该模块进一步通过[5] 中采用的跟踪器跟踪每个物体, 获取物体序列 $\mathbb{S} = \{\mathcal{S}_1, \mathcal{S}_2 ..., \mathcal{S}_n\}$. 由于在跟踪序列中的同一物体具有高度相关性, 直观的方法是通过密集连接的图建模帧间关系. 然而, 直接在这种密集图上使用图网络进行消息传递会带来大量额外的开销, 如第 3.4 节讨论所述. 因此, 本方法转而使用 Transformer, 因其在建模全局关系方面表现出色[46, 65].

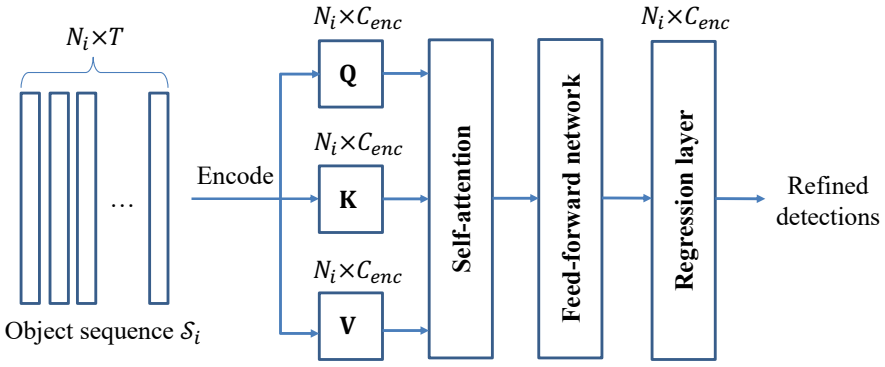


图 3　帧间关系模块的详细结构.　给定每个物体序列, 它通过一个轻量化的 Transformer 对其特征进行提取并优化.

Figure 3　The detailed structure of InterRM. Given the tracked object sequence for each object, we perform lightweight transformer-based feature extraction for individual refinement.

对于每个物体序列 $\mathcal{S}_i$, 其长度为 N_i, 其序列特征为 $\{d_{i,-t_{N_i-1}}, d_{i,-t_{N_i-2}}, ..., d_{i,-t_1}, d_i\}$, 这些特征是单阶段基线检测器的预测结果, 其中 $\{-t_{N_i-1}, -t_{N_i-2}, ..., -t_1\}$ 表示过去的时间戳. N_i 是由跟踪结果动态决定的. 在物体序列 $\mathcal{S}_i$ 中, 来自过去时间戳的位置信息已被投影到当前帧的坐标系中. 该模块先对序列特征进行通道拼接, 并使用线性层对特征进行编码:

$$\mathbf{D} = \mathrm{Concat}(d_{i,-t_{N_i-1}}, d_{i,-t_{N_i-2}}, ..., d_{i,-t_1}, d_i)^{\mathbf{T}}\mathbf{W}^e, \tag{6}$$

其中 $\mathbf{W}^e$ 是线性层的权重, 输出通道数为 C_{enc}, $\mathbf{D} \in \mathbb{R}^{N_i \times C_{enc}}$ 为编码后的特征, N_i 为物体的跟踪帧数.

假设自注意力机制中的查询、键和值分别表示为 $\mathbf{Q}$、$\mathbf{K}$ 和 $\mathbf{V}$, 它们可以通过下式计算:

$$(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = (\mathbf{D}\mathbf{W}^q, \mathbf{D}\mathbf{W}^k, \mathbf{D}\mathbf{W}^v), \tag{7}$$

其中 $\mathbf{W}^q$、$\mathbf{W}^k$ 和 $\mathbf{W}^v$ 分别为计算查询 (query)、键 (key) 和值 (value) 的线性层的权重. 然后可以计算自注意力 $\mathbf{A}$:

$$\mathbf{A} = \mathrm{Softmax}(\frac{\mathbf{Q} \times \mathbf{K}^{\mathbf{T}}}{\sqrt{C_K}}) \times \mathbf{V}, \tag{8}$$

其中 C_K 为键的通道数, $\frac{1}{\sqrt{C_K}}$ 为自注意力的缩放因子. 这里简略了多头注意力机制的概念. 接下来, 前馈网络 (Feed Forward Network, FFN) 学习残差表示以增强自注意力 $\mathbf{A}$:

$$\mathbf{A}' = \mathrm{FFN}(\mathbf{A} + \mathbf{D}) + \mathbf{A} + \mathbf{D}, \tag{9}$$

其中 FFN 由两个线性层组成, 扩展率为 2, $\mathbf{A}'$ 是优化后的特征. 最后, 本模块应用回归层来预测物体的位置和朝向, 并选择平滑 L1 损失[35] 作为回归损失函数, 它也是帧间关系模块的总体损失 $\mathcal{L}_{inter}$. 帧间关系模块的详细结构如图 3 所示.

3.4　时间复杂度

接下来的内容为分析帧内关系模块和帧间关系模块的时间复杂度.

帧内关系模块. 包括回归头在内, 帧内关系模块的时间复杂度为 $O(C_x^2 n)$. 由于每帧激光雷达通常包含几十个物体, 帧内关系模块仅需要很少的计算开销 (小于 0.1G FLOPs).

帧间关系模块. 帧内关系模块使用跟踪序列对每个物体进行优化. 对于每个物体序列, 假设序列的平均长度为 N, 自注意力层和线性层的时间复杂度分别为 $O(C_{enc}N^2)$ 和 $O(C_{enc}^2 N)$. 帧间关系模块的总体时间复杂度为 $O(n(C_{enc}^2 N + C_{enc}N^2))$. 另外, 对于具有 $\frac{N(N-1)}{2}$ 条边的稠密图, 使用仅一次迭代的图网络时间复杂度为 $O(nC_{enc}^2 N^2)$, 其计算量约为 64.9G FLOPs, 是使用 Transformer 的 20 倍. 因此, 本方法的设置在效率方面具有显著优势.

3.5 可扩展性和适用性

4 实验

4.1 实验设置

实现细节. 本文使用 PyTorch 库[66] 实现了 Ret3D 网络. 由于 Ret3D 非常高效, 整个训练过程仅在单个 RTX 2080Ti GPU 上进行. 网络在训练时使用 AdamW[67] 优化器, 权重衰减为 0.01. 两个著名的一阶段通用检测器, SECOND[2] 和 CenterPoint[5] 作为基线检测器. 两者都使用稀疏 3D 卷积作为骨干进行特征提取, 体素大小为 $\{0.1m, 0.1m, 0.15m\}$. 本文使用知名开源工具箱 OpenPCDet[68] 重新实现了 SECOND[2], 对于 CenterPoint[5] 则直接使用了作者提供的两帧输入的预训练模型. 帧内关系模块的学习率设置为 3×10^{-4}, 每个小批量 (mini-batch) 包含 16 帧激光雷达数据. 帧间关系模块的学习率相同, 每个小批量包含 64 个物体序列. 两个模块都进行 10 万次迭代训练. 在帧内关系模块中, 过去一帧和当前帧的检测结果会聚合在同一张稀疏图中用来预测物体速度. 损失平衡权重 λ_1 和 λ_2 分别设置为 2.0 和 0.2. EdgeConv[38] 是式 (2) 的非线性转换函数. 帧间关系模块使用了[5] 中的基础跟踪算法, 平均物体序列长度 N 为 100, C_{enc} 设置为 256. 计算自注意力 $\mathbf{A}$ 时使用了 16 个头 式 (8) , 以捕捉多样化的注意力.

数据集. 所有实验都在 Waymo 公开数据集[1] 上进行. 与仅标注了 15K 激光雷达帧的 KITTI[18] 数据集相比, Waymo 公开数据集包含 798 个训练序列和 202 个验证序列, 分别拥有 158K 和 40K 标注的激光雷达帧. 每个序列的采样频率为 10Hz. Waymo 公开数据集包含 1200 万个 3D 检测框, 数据量是 KITTI 数据集的 150 倍. 前者的场景也更加复杂, 因此挑战性更高. 本文在其训练集中训练了所提的 Ret3D 框架, 并在验证集上进行评估. 检测范围在 x 和 y 轴 $[-75.2m, 75.2m]$, 在 z 轴为 $[-2m, 4m]$, 因此基线检测器的输入体素数量为 $41 \times 1504 \times 1504$.

评估指标. 根据 Waymo 公开数据集[1] 的建议, 本文采用 mAPH 即加权朝向精度的平均精度作为主要评估指标. 关于 mAPH 的更多细节请参考原始论文[1]. mAP 则作为参考指标. 按照 Waymo 公开数据集[1] 的建议, 车辆、行人和骑行者类别的 IoU 阈值分别设置为 0.7、0.5 和 0.5, 并在一级、二级难度 (LEVEL_1, LEVEL_2) 下分别进行评估. 其中, 二级难度没有移除点数量少于 5 的检测框[1], 所以其难度更大.

表 1 与现有车辆位置检测方法的对比结果. 该表为在 **Waymo** 公开数据集的验证集上的评估结果[1]. **PV-RCNN*** 为使用中心检测头[5] 在 **OpenPCDet** 工具箱中重新实现的 **PV-RCNN**. 本文在一级和二级难度设置下报告了结果, **Ret3D** 相比近期强竞争对手取得了显著提升. 多帧方法的名称以 "**_nf**" 结尾, 表示直接使用 n 帧进行检测.

Table 1 Comparison with state-of-the-art methods for vehicle detection. Results are evaluated on the Waymo Open Dataset (WOD) validation set[1]. Results of PV-RCNN* are re-implemented PV-RCNN with center-based detection head[5] by the OpenPCDet toolbox. Names of multi-frame methods are ended with "_nf", marking that n frames are directly used for detection.

Setting	Method	mAPH	mAP
LEVEL_1	PointPillars[3]	62.8	63.3
	LaserNet[69]	50.1	52.1
	PV-RCNN[4]	-	70.3
	PV-RCNN*[4]	78.0	77.5
	PillarOD[27]	-	69.8
	RCD[13]	69.6	69.2
	CVCNet[70]	-	65.2
	Voxel R-CNN[17]	-	75.6
	PVGNet[14]	-	74.0
	LiDAR R-CNN[34]	75.5	76.0
	RangeDet[71]	-	72.9
	CT3D[37]	-	76.3
	VoTR-TSD[15]	74.3	75.0
	3D-MAN_16f[60]	74.5	74.0
	VoxelNeXt[30]	77.7	78.2
	RSN_3f[72]	78.1	78.4
	CenterPoint_2f[5]	74.4	74.9
	Ret3D_2f (Ours)	**81.0**	**81.6**
LEVEL_2	PointPillars[3]	55.1	55.6
	PV-RCNN[4]	63.7	64.2
	Voxel R-CNN[17]	-	66.6
	LiDAR R-CNN[34]	67.9	68.3
	VoTR-TSD[15]	65.3	65.9
	CT3D[37]	-	69.0
	3D-MAN_16f[60]	67.6	67.1
	RSN_3f[72]	69.1	69.5
	FocalFormer3D[59]	67.6	68.1
	VoxelNeXt[30]	69.4	69.9
	CenterPoint_2f[5]	69.7	70.2
	Ret3D_2f (Ours)	**72.9**	**73.4**

4.2 评估结果

车辆位置检测. 本文将所提出的 Ret3D 框架与过去三年中发表的 14 种著名方法进行对比. 对比结果如表 1 所示. 在一级难度设置下, Ret3D 在 mAPH 和 mAP 上分别比最强的竞争对手提高了 2.9% 和 3.2%. 在更具挑战性的二级难度设置下, Ret3D 在 mAPH 和 mAP 上都提高了 3.2%. 这些

表 2　与现有行人位置检测方法的对比结果. 该表为在 Waymo 公开数据集的验证集上的评估结果[1]. 多帧方法的名称以 "_nf" 结尾, 表示直接使用 n 帧进行检测.

Table 2 Comparison with state-of-the-art methods for pedestrian detection. Results are evaluated on the WOD validation set[1]. Names of multi-frame methods are ended with "_nf", marking that n frames are directly used for detection.

Settings	Methods	mAPH	mAP
LEVEL_1	PointPillars[3]	56.1	70.0
	PillarOD[27]		72.5
	MVF [73]	-	65.3
	PVGNet[14]	-	69.5
	PointAugmenting[74]	-	75.4
	RangeDet[71]	-	75.9
	CenterPoint_2f[5]	75.1	78.3
	Ret3D (Ours)	**79.7**	**82.8**
LEVEL_2	PointPillars[3]	51.1	63.8
	PointAugmenting[74]	-	70.6
	FocalFormer3D[59]	66.4	72.4
	VoxelNeXt[30]	68.6	73.5
	CenterPoint_2f[5]	70.3	73.3
	Ret3D_2f (Ours)	**71.9**	**74.9**

表 3　帧内和帧间关系模块的效果. "*" 表示其计算成本是在检测到 50 个目标的情况下计算的. 结果在二级难度的设置下使用 Waymo 公开数据集的验证集[1] 测试得出的.

Table 3 Effect of IntraRM and InterRM. "*" indicates that the computational cost is computed with 50 detected objects. Results are tested on the WOD validation set[1] under the LEVEL_2 setting.

No.	Methods	# Params	# FLOPs	Overall		Vehicle		Pedestrian		Cyclist	
				mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP
1	CenterPoint[5]	7.8M	127.7G	68.2	69.9	67.3	67.8	67.5	71.0	69.9	70.8
2	No.1 + Two-stage[5]	1.5M	0.1G*	70.3	71.7	69.7	70.2	70.3	73.3	70.9	71.7
3	No.1 + IntraRM (Ours)	1.4M	0.1G*	71.1	72.5	70.9	71.4	70.9	73.8	71.6	72.4
4	No.3 + InterRM (Ours)	0.7M	3.2G*	**72.3**	**73.8**	**72.9**	**73.4**	**71.9**	**74.9**	**72.2**	**73.0**
-	*Improvement*	-	-	**+4.1**	**+3.9**	**+5.6**	**+5.6**	**+4.4**	**+3.9**	**+2.3**	**+2.2**
5	SECOND[2]	7.8M	124.0G	59.3	64.7	65.1	65.7	53.5	62.8	59.3	65.6
6	No.5 + IntraRM (Ours)	1.4M	0.1G*	63.6	67.0	66.8	67.4	57.6	65.9	66.4	67.6
7	No.6 + InterRM (Ours)	0.7M	3.2G*	64.8	68.2	69.2	69.8	58.6	66.8	66.7	67.9
-	*Improvement*	-	-	**+5.5**	**+3.5**	**+4.1**	**+4.1**	**+5.1**	**+4.0**	**+7.4**	**+2.3**

显著的改进表明, 物体关系对于车辆位置检测的提升非常有帮助.

行人位置检测. 行人位置检测对于自动驾驶同样非常重要. 本文仍然将 Ret3D 与近三年报告了行人位置检测结果的著名方法进行比较. 对比结果如表 2所示. Ret3D 在一级和二级难度设置下的 mAPH 相对过去最强的方法提升了 4.6% 和 1.6%, 这显示了 Ret3D 在行人位置检测中的优势.

表 4 帧间关系模块中位置编码模式的讨论. “PE” 代表位置编码. 该实验在 Waymo 公开数据集的验证集下进行, 并采用二级难度设置进行评估.

Table 4 Positional encoding in InterRM. “PE” denotes the positional encoding. Results are tested on the WOD validation set[1] under the LEVEL_2 setting.

PE Settings	Vehicle		Pedestrian		Cyclist	
	mAPH	mAP	mAPH	mAP	mAPH	mAP
Baseline	70.9	71.4	70.9	73.8	71.6	72.4
+ Implicit	**72.9**	**73.4**	**71.9**	**74.9**	72.2	73.0
++Temporal	72.7	73.1	71.8	74.7	**72.6**	**73.4**
++Spatial	71.9	72.4	71.4	74.3	72.0	72.7

4.3 模型分析

帧内关系模块与帧间关系模块的影响. 正如在第 1 节中描述的, 本文探索了如何显式利用帧内和帧间物体关系来提高高效 3D 物体检测器的性能. 为此, 本文提出了 Ret3D 框架, 该框架由两个模块组成, 即帧内关系模块和帧间关系模块. 为了验证所提出的帧内关系模块和帧间关系模块的有效性, 本文进行了消融实验, 结果如表 3 所示. 本文以基于中心点预测的 CenterPoint[5] 和基于锚框的 SECOND[2] 检测器为基线检测器进行实验. 本文还将 Ret3D 与二阶段的 CenterPoint[5] (该方法独立优化每个检测结果) 进行对比. 实验表明, 帧内关系模块相比基线检测器有显著的提升 (在 mAPH 上提升了 2.9%, 在 mAP 上提升了 2.6%), 这显示了利用帧内物体关系的优越性. 用帧内关系模块替换二阶段 CenterPoint 带来了 0.8% 的 mAPH 和 mAP 提升, 表明帧内关系比独立优化每个物体更重要. 在帧内关系模块基础上加入帧间关系模块可以进一步显著提升性能 (mAPH 提升了 1.2%, mAP 提升了 1.3%). 总体而言, Ret3D 相比 CenterPoint[5] 在二级难度的设置下分别提升了 4.1% mAPH 和 3.9% mAP. 表 3 中的结果还显示, Ret3D 相比 SECOND[2] 框架在 mAPH 和 mAP 上分别提升了 5.6% 和 3.5%. 以上分析表明, 帧内和帧间物体关系对于提升自动驾驶中高效 3D 物体检测性能都非常重要.

为了进行效率分析, 本文在表 3 中列出了网络参数和计算复杂度 (FLOPs). 该表结果显示两个基线检测器的计算成本都超过 100G FLOPs (No. 1, 5). 二阶段的 CenterPoint[5] (No. 2) 引入了非常小的网络复杂度和可忽略的计算成本 (0.1G FLOPs), 因为它仅基于鸟瞰图特征向量进行单独优化. 帧内关系模块引入了与二阶段 CenterPoint[5] 相当的网络复杂度和计算成本, 同时显式利用了帧内物体关系. 帧间关系模块的网络参数比帧内关系模块和二阶段 CenterPoint[5] 更少, 但帧间关系模块需要更多的计算成本. 然而, 帧间关系模块的计算成本仍然远低于基线检测器, 保持了 3D 物体检测的高效性.

处理速度. 在高效的 3D 物体检测中, 运行速度是一个关键因素. 本文测试了 Ret3D 的运行速度, 所有实验均在单个 RTX 2080Ti 显卡上进行. 基于 RTX 2080Ti 显卡, 一阶段 CenterPoint 的每帧耗时约为 130ms. 帧内模块的每帧耗时约为 2ms, 而帧间模块的每帧耗时约为 10ms, 两个模块的总耗时约为一阶段检测器的 1/10. 该实验表明, 尽管引入了额外的计算模块, Ret3D 仍然保持了较高的运行效率.

表 5 半径 r 的设置讨论. 该实验在 Waymo 公开数据集的验证集下进行, 并采用二级难度设置进行评估.

Table 5 The radius setting in IntraRM. Results are tested on the WOD validation set[1] under the LEVEL_2 setting.

Radius	Vehicle		Pedestrian		Cyclist	
	mAPH	mAP	mAPH	mAP	mAPH	mAP
-	65.1	65.7	53.5	62.8	59.3	65.6
1.0m	65.9	66.5	57.6	65.7	65.1	66.4
2.0m	**66.8**	**67.4**	**57.6**	**65.9**	**66.4**	**67.6**
4.0m	**66.8**	**67.4**	**57.6**	65.8	66.3	67.5

表 6 稀疏图迭代次数 m 的设置讨论. 该实验在 Waymo 公开数据集的验证集下进行, 并采用二级难度设置进行评估.

Table 6 Number of iterations m for graph update. Results are tested on the WOD validation set[1] under the LEVEL_2 setting.

# Iters	Vehicle		Pedestrian		Cyclist	
	mAPH	mAP	mAPH	mAP	mAPH	mAP
-	65.1	65.7	53.5	62.8	59.3	65.6
1	66.6	67.2	57.4	65.6	65.6	66.9
2	66.7	67.3	57.4	65.7	65.9	67.1
4	**66.8**	**67.4**	**57.6**	**65.9**	**66.4**	**67.6**
6	**66.8**	**67.4**	57.5	**65.9**	66.3	67.5

位置编码在帧间关系模块中的作用. 一般来说, Transformer[46, 65] 倾向于接受具有位置编码的输入. 默认情况下, 本文使用线性层对检测结果进行编码 (见式 (6)), 这可以看作是隐式的时空位置编码. 这里, 本文外部添加了两种类型的显式编码, 即时间编码或空间编码, 分别通过在编码特征 **D** 上直接添加时间或空间项来实现. 结果如表 4 所示. 该实验的基线为已添加帧内关系模块的 CenterPoint[5] 方法. 可以发现发现, 使用时间编码会略微降低车辆和行人类别的性能, 而对于骑行者类别, 性能会略有提升. 空间编码会显著降低所有三个类别的检测精度. 因此, 在实践中本文不添加显式位置编码.

帧间模块中网络模块的选择. 本文比较了 Transformer 和图网络在帧间模块上的效果. 实验以一阶 CenterPoint 检测器堆叠帧内模块优化为基础. 结果表明, 使用图网络的帧间模块在基线检测器上的提升仅为 0.5% mAPH, 而使用 Transformer 的提升为 1.2% mAPH. 此外, 图网络引入的额外计算量为 64.9GFLOPs, 而 Tranformer 引入的计算量仅为 3.2GFLOPs.

帧内关系模块中的半径 r 设置. 在帧内关系模块中, 本文构建的图只连接相邻的节点对. 对于每个节点 v_i, 本文只连接与其空间中心在以 v_i 为中心、半径为 r 米的圆范围内的其他节点. 理想情况下, r 越大, 构建的图越稠密, 性能也越好, 但这会增加计算成本. 基于本文的观察, 本文在帧内关系模块中对不同的 r 值进行了实验. 结果如表 5 所示, 本文使用 SECOND[2] 作为基线. 本文发现, $r \geqslant 1.0$m 已经足以用于行人和车辆位置检测. 对于骑行者检测, r 越大性能越好. 然而, 本文发现当 $r = 4$m 时, 性能略有下降, 并且图中的边数量增加了 3 倍. 考虑到效果与效率之间的平衡, 本文最终选择 $r = 2$m.

稀疏图的迭代次数 m. 由于该图的更新过程可以是迭代的, 本文可以通过多次迭代优化图中的节点特征. 本文在表 6 中展示了不同迭代次数对性能的影响. 实验基线为 SECOND[2]. 可以发现, 不同的 m 值对车辆和行人位置检测的性能影响不大. 当 $m = 4$ 时, 在车辆位置检测上取得了最佳效果, 并在行人位置检测中表现良好. 对于骑行者位置检测, $m = 4$ 的效果最好. 因此, 本文最终选择 $m = 4$ 进行图更新.

5 结论

为了显式地利用帧内和帧间物体关系来提升 3D 物体检测的性能的同时保持高效性, 本文提出了一种简单高效的框架——Ret3D. 它采用了二阶段的检测方式, 通过帧内和帧间关系模块来优化单阶段检测器的检测结果. 实验结果证明了这两种物体关系对提升检测性能的重要性. 在 Waymo 公开数据集的验证集上, Ret3D 取得了显著的性能提升. 以车辆位置检测为例, Ret3D 分别在一级和二级难度的 mAPH 上相对近期最强的著名方法提高了 2.9% 和 3.2%.

参考文献

1 Sun P, Kretzschmar H, Dotiwalla X, et al. Scalability in perception for autonomous driving: Waymo open dataset. In: IEEE Conf. Comput. Vis. Pattern Recog., 2020. 2446–2454

2 Yan Y, Mao Y, Li B. SECOND: Sparsely embedded convolutional detection. Sensors, 2018, 18: 3337

3 Lang A H, Vora S, Caesar H, et al. Pointpillars: Fast encoders for object detection from point clouds. In: IEEE Conf. Comput. Vis. Pattern Recog., 2019. 12697–12705

4 Shi S, Guo C, Jiang L, et al. PV-RCNN: Point-voxel feature set abstraction for 3D object detection. In: IEEE Conf. Comput. Vis. Pattern Recog., 2020. 10529–10538

5 Yin T, Zhou X, Krahenbuhl P. Center-based 3D object detection and tracking. In: IEEE Conf. Comput. Vis. Pattern Recog., 2021. 11784–11793

6 Simonelli A, Bulo S R, Porzi L, et al. Disentangling monocular 3D object detection. In: Int. Conf. Comput. Vis., 2019. 1991–1999

7 Chen Y, Tai L, Sun K, et al. Monopair: Monocular 3D object detection using pairwise spatial relationships. In: IEEE Conf. Comput. Vis. Pattern Recog., 2020. 12093–12102

8 Reading C, Harakeh A, Chae J, et al. Categorical depth distribution network for monocular 3D object detection. In: IEEE Conf. Comput. Vis. Pattern Recog., 2021. 8555–8564

9 Tu Z. Auto-context and its application to high-level vision tasks. In: IEEE Conf. Comput. Vis. Pattern Recog., 2008. 1–8

10 Galleguillos C, Belongie S. Context based object categorization: A critical survey. CVIU, 2010, 114: 712–722

11 Mottaghi R, Chen X, Liu X, et al. The role of context for object detection and semantic segmentation in the wild. In: IEEE Conf. Comput. Vis. Pattern Recog., 2014. 891–898

12 Hu H, Gu J, Zhang Z, et al. Relation networks for object detection. In: IEEE Conf. Comput. Vis. Pattern Recog., 2018. 3588–3597

13 Bewley A, Sun P, Mensink T, et al. Range conditioned dilated convolutions for scale invariant 3D object detection. In: Conference on Robot Learning, 2020

14 Miao Z, Chen J, Pan H, et al. PVGNet: A bottom-up one-stage 3D object detector with integrated multi-level features. In: IEEE Conf. Comput. Vis. Pattern Recog., 2021. 3279–3288

15 Mao J, Xue Y, Niu M, et al. Voxel transformer for 3D object detection. In: Int. Conf. Comput. Vis., 2021. 3164–3173

16 Qi C R, Zhou Y, Najibi M, et al. Offboard 3D object detection from point cloud sequences. In: IEEE Conf. Comput. Vis. Pattern Recog., 2021. 6134–6144

17 Deng J, Shi S, Li P, et al. Voxel R-CNN: Towards high performance voxel-based 3D object detection. In: AAAI, 2021

18 Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite. In: IEEE Conf. Comput. Vis. Pattern Recog.. IEEE2012. 3354–3361

19 Liang M, Yang B, Chen Y, et al. Multi-task multi-sensor fusion for 3D object detection. In: IEEE Conf. Comput. Vis. Pattern Recog., 2019. 7345–7353

20 Yang Z, Sun Y, Liu S, et al. 3DSSD: Point-based 3D single stage object detector. In: IEEE Conf. Comput. Vis. Pattern Recog., 2020. 11040–11048

21 Engelcke M, Rao D, Wang D Z, et al. Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks. In: IEEE ICRA, 2017

22 Zhou Y, Tuzel O. VoxelNet: End-to-end learning for point cloud based 3D object detection. In: IEEE Conf. Comput. Vis. Pattern Recog., 2018. 4490–4499

23 Qi C R, Su H, Mo K, et al. PointNet: Deep learning on point sets for 3D classification and segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog., 2017

24 Yang B, Luo W, Urtasun R. Pixor: Real-time 3D object detection from point clouds. In: IEEE Conf. Comput. Vis. Pattern Recog., 2018. 7652–7660

25 Duan K, Bai S, Xie L, et al. Centernet: Keypoint triplets for object detection. In: Int. Conf. Comput. Vis., 2019. 6569–6578

26 Tian Z, Shen C, Chen H, et al. FCOS: A simple and strong anchor-free object detector. IEEE Trans. Pattern Anal. Mach. Intell., 2020

27 Wang Y, Fathi A, Kundu A, et al. Pillar-based object detection for autonomous driving. In: Eur. Conf. Comput. Vis.. Springer2020. 18–34

28 Zhang G, Junnan C, Gao G, et al. Hednet: A hierarchical encoder-decoder network for 3d object detection in point clouds. Adv. Neural Inform. Process. Syst., 2023, 36

29 Fan L, Wang F, Wang N, et al. Fully sparse 3d object detection. Adv. Neural Inform. Process. Syst., 2022, 35: 351–363

30 Chen Y, Liu J, Zhang X, et al. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In: IEEE Conf. Comput. Vis. Pattern Recog., 2023. 21674–21683

31 Chen Y, Liu S, Shen X, et al. Fast point R-CNN. In: Int. Conf. Comput. Vis., 2019. 9775–9784

32 Shi S, Wang X, Li H. PointRCNN: 3D object proposal generation and detection from point cloud. In: IEEE Conf. Comput. Vis. Pattern Recog., 2019. 770–779

33 Shi S, Wang Z, Shi J, et al. From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. IEEE Trans. Pattern Anal. Mach. Intell., 2020

34 Li Z, Wang F, Wang N. LiDAR R-CNN: An efficient and universal 3D object detector. In: IEEE Conf. Comput. Vis. Pattern Recog., 2021. 7546–7555

35 Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell., 2016, 39: 1137–1149

36 Yang Z, Sun Y, Liu S, et al. STD: Sparse-to-dense 3D object detector for point cloud. In: Int. Conf. Comput. Vis., 2019. 1951–1960

37 Sheng H, Cai S, Liu Y, et al. Improving 3D object detection with channel-wise transformer. In: Int. Conf. Comput. Vis., 2021. 2743–2752

38 Wang Y, Sun Y, Liu Z, et al. Dynamic graph CNN for learning on point clouds. ACM Trans. Graph., 2019, 38: 1–12

39 Najibi M, Lai G, Kundu A, et al. DOPS: Learning to detect 3D objects and predict their 3D shapes. In: IEEE Conf. Comput. Vis. Pattern Recog., 2020. 11913–11922

40 He Q, Wang Z, Zeng H, et al. SVGA-Net: Sparse voxel-graph attention network for 3D object detection from point clouds. arXiv preprint arXiv:2006.04043, 2020

41 Zarzar J, Giancola S, Ghanem B. PointRGCN: Graph convolution networks for 3d vehicles detection refinement. arXiv preprint arXiv:1911.12236, 2019

42 Feng M, Gilani S Z, Wang Y, et al. Relation graph network for 3D object detection in point clouds. IEEE Trans. Image Process., 2021, 30: 92–107

43 Wang Y, Solomon J. Object DGCNN: 3D object detection using dynamic graphs. In: Adv. Neural Inform. Process. Syst., 2021

44 Shi W, Rajkumar R. Point-GNN: Graph neural network for 3D object detection in a point cloud. In: IEEE Conf. Comput. Vis. Pattern Recog., 2020. 1711–1719

45 Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers. In: Eur. Conf. Comput. Vis.. Springer2020. 213–229

46 Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In: Int. Conf. Learn. Represent., 2021

47 Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows. In: Int. Conf. Comput. Vis., 2021

48 Wang W, Xie E, Li X, et al. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Int. Conf. Comput. Vis., 2021

49 Chen H, Wang Y, Guo T, et al. Pre-trained image processing transformer. In: IEEE Conf. Comput. Vis. Pattern Recog., 2021. 12299–12310

50 Wu Y H, Liu Y, Zhan X, et al. P2T: Pyramid pooling transformer for scene understanding. arXiv preprint arXiv:2106.12011, 2021

51 Liu Y, Sun G, Qiu Y, et al. Transformer in convolutional neural networks. arXiv preprint arXiv:2106.03180, 2021

52 Zhu X, Su W, Lu L, et al. Deformable DETR: Deformable transformers for end-to-end object detection. In: Int. Conf. Learn. Represent., 2021

53 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: IEEE Conf. Comput. Vis. Pattern Recog., 2016. 770–778

54 Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks. In: IEEE Conf. Comput. Vis. Pattern Recog., 2017. 4700–4708

55 Liu Z, Zhang Z, Cao Y, et al. Group-free 3D object detection via transformers. In: Int. Conf. Comput. Vis., 2021

56 Fan L, Pang Z, Zhang T, et al. Embracing single stride 3d object detector with sparse transformer. In: IEEE Conf. Comput. Vis. Pattern Recog., 2022. 8458–8468

57 Wang H, Shi C, Shi S, et al. DSVT: Dynamic sparse voxel transformer with rotated sets. In: IEEE Conf. Comput. Vis. Pattern Recog., 2023. 13520–13529

58 Liu Z, Yang X, Tang H, et al. Flatformer: Flattened window attention for efficient point cloud transformer. In: IEEE Conf. Comput. Vis. Pattern Recog., 2023. 1200–1211

59 Chen Y, Yu Z, Chen Y, et al. Focalformer3d: focusing on hard instance for 3d object detection. In: Int. Conf. Comput. Vis., 2023. 8394–8405

60 Yang Z, Zhou Y, Chen Z, et al. 3D-MAN: 3d multi-frame attention network for object detection. In: IEEE Conf. Comput. Vis. Pattern Recog., 2021. 1863–1872

61 Chen X, Shi S, Zhu B, et al. MPPNet: Multi-frame feature intertwining with proxy points for 3d temporal object detection. In: Eur. Conf. Comput. Vis.. Springer2022. 680–697

62 Wu H, Chen Y, Wang N, et al. Sequence level semantics aggregation for video object detection. In: Int. Conf. Comput. Vis., 2019. 9217–9225

63 Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell., 2020, 42: 318–327

64 Zakharov S, Kehl W, Bhargava A, et al. Autolabeling 3D objects with differentiable rendering of SDF shape priors. In: IEEE Conf. Comput. Vis. Pattern Recog., 2020. 12224–12233

65 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Adv. Neural Inform. Process. Syst., 2017. 6000–6010

66 Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library. Adv. Neural Inform. Process. Syst., 2019, 32: 8026–8037

67 Loshchilov I, Hutter F. Decoupled weight decay regularization. In: Int. Conf. Learn. Represent., 2017

68 Team O D. OpenPCDet: An open-source toolbox for 3D object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020

69 Meyer G P, Laddha A, Kee E, et al. LaserNet: An efficient probabilistic 3D object detector for autonomous driving. In: IEEE Conf. Comput. Vis. Pattern Recog., 2019. 12677–12686

70 Chen Q, Sun L, Cheung E, et al. Every view counts: Cross-view consistency in 3D object detection with hybrid-cylindrical-spherical voxelization. In: Adv. Neural Inform. Process. Syst., 2020

71 Fan L, Xiong X, Wang F, et al. RangeDet: In defense of range view for LiDAR-based 3D object detection. In: Int. Conf. Comput. Vis., 2021. 2918–2927

72 Sun P, Wang W, Chai Y, et al. RSN: range sparse net for efficient, accurate lidar 3d object detection. CoRR, 2021, abs/2106.13365. URL https://arxiv.org/abs/2106.13365

73 Zhou Y, Sun P, Zhang Y, et al. End-to-end multi-view fusion for 3D object detection in LiDAR point clouds. In: Conference on Robot Learning, 2020. 923–932

74 Wang C, Ma C, Zhu M, et al. PointAugmenting: Cross-modal augmentation for 3D object detection. In: IEEE Conf. Comput. Vis. Pattern Recog., 2021. 11794–11803

Ret3D: Rethinking Object Relations for Efficient 3D Object Detection

Yu-Huan Wu[1], Da Zhang[2], Yun Liu[1], Le Zhang[3] & Ming-Ming Cheng[1*]

1. *College of Computer Science, Nankai University, Tianjin* 300350*, China*;
2. *University of California Santa Barbara, California* 93106*, USA*;
3. *School of Information and Communication Engineering, UESTC, Chengdu* 611731*, China*
* Corresponding author. E-mail: cmm@nankai.edu.cn

Abstract Current efficient LiDAR-based detection frameworks are lacking in exploiting object relations, which naturally present in both spatial and temporal manners. To this end, we introduce a simple, efficient, and effective two-stage detector, termed as Ret3D. At the core of Ret3D is the utilization of novel intra-frame and inter-frame relation modules to capture the spatial and temporal relations accordingly. More Specifically, intra-frame relation module (InterRM) encapsulates the intra-frame objects into a sparse graph and thus allows us to refine the object features through efficient message passing. On the other hand, inter-frame relation module (IntraRM) densely connects each object in its corresponding tracked sequences dynamically, and leverages such temporal information to further enhance its representations efficiently through a lightweight transformer network. We instantiate our novel designs of IntraRM and InterRM with general center-based or anchor-based detectors and evaluate them on Waymo Open Dataset (WOD). With negligible extra overhead, Ret3D achieves the state-of-the-art performance, being 2.9% and 3.2% higher than the recent competitor in terms of the LEVEL_1 and LEVEL_2 mAPH metrics on vehicle detection, respectively.

Keywords 3D Object Detection, Object Relations, Autonomous Driving