基于原始点云网格自注意力机制的三维目标检测方法

鲁斌[1,2]，孙洋[1,2]，杨振宇[1,2]
（1. 华北电力大学计算机系，河北 保定 071003；2. 复杂能源系统智能计算教育部工程研究中心，河北 保定 071003）

摘 要：为了增强感兴趣区域（RoI）的特征表达，包括空间网格特征编码模块和软回归损失，提出了一种基于原始点云网格自注意力机制的三维目标检测方法 GT3D。网格特征编码模块用于通过自注意力机制对点的局部特征和空间特征进行有效加权，充分考虑点云之间的几何关系，以提供更准确的特征表达；软回归损失用于改善数据标注过程中由于标注不准确而产生的回归歧义问题。将所提方法在公开的三维目标检测数据集 KITTI 上进行实验。结果表明，所提方法相比其他已公开的基于点云的三维目标检测方法检测准确率提升明显，并提交了 KITTI 官方测试集进行公开测试，对简单、中等和困难 3 个难度等级的汽车检测准确率分别达到 91.45%、82.76%和 79.74%。

关键词：三维目标检测；点云；自注意力机制；空间坐标编码；软回归损失
中图分类号：TP391.4
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2023189

Grid self-attention mechanism 3D object detection method based on raw point cloud

LU Bin[1,2], SUN Yang[1,2], YANG Zhenyu[1,2]
1. School of Control and Compute Engineering, North China Electric Power University, Baoding 071003, China
2. Engineering Research Center of Intelligent Computing for Complex Energy Systems, Ministry of Education, Baoding 071003, China

Abstract: To enhance the feature representation of region of interest (RoI), which incorporated a spatial context encoding module and soft regression loss, a grid self-attention mechanism 3D object detection method based on raw point cloud, named GT3D, was proposed. The spatial context encoding module was designed to effectively weight the local and spatial features of points through the attention mechanism, considering the contribution of different point cloud features for a more accurate feature representation. The soft regression loss was introduced to address label ambiguity arising during the data annotation phase. Experiments conducted on the public KITTI 3D object detection dataset demonstrate that the proposed method achieves significant improvements in detection accuracy compared to other publicly available point cloud-based 3D object detection methods. The detection results of the test set are submitted to the official KITTI server for public evaluation, achieving detection accuracies of 91.45%, 82.76%, and 79.74% for easy, moderate, and hard difficulty levels in car detection, respectively.

Keywords: 3D object detection, point cloud, self-attention mechanism, spatial coordinate encoding, soft regression loss

0 引言

近年来，三维目标检测技术作为机器人和自动驾驶感知系统的关键技术之一，已经取得了显著的进步。该技术利用由激光雷达捕获的点云数据来描绘物体的三维结构，估计其姿态，并感知空间距离。因此，激光雷达成为三维目标检测的首选传感器。基于原始点云的三维目标检测旨在利用这些点云数据来识别

收稿日期：2023−06−15；**修回日期：**2023−09−05
通信作者：孙洋，sun.yang@ncepu.edu.cn
基金项目：国家自然科学基金资助项目（No.62371188）；河北省在读研究生创新能力培养基金资助项目（No.CXZZBS2023153）
Foundation Items: The National Natural Science Foundation of China (No.62371188), Hebei Province Postgraduate Innovation Capability Training Project (No.CXZZBS2023153)

环境中物体的类别、位置、大小和方向，为深入理解场景提供基础。然而，与图像不同，点云数据是无序且不均匀的，这使无法直接使用卷积神经网络（CNN, convolutional neural network）来学习特征，从而增加了基于点云的三维目标检测技术的挑战性。

目前，大多数检测方法采用两阶段范式，以获得更好的检测效果。例如，PV-RCNN[1]使用 SECOND（sparsely embedded convolutional detection）[2]和 PointNet++[3]作为其基础网络，以分别提取点和体素的特征，并在第二阶段通过采用最大池化方法对点特征进行聚合。Voxel R-CNN[4]则省略了 PV-RCNN 中的点采样步骤，并在第二阶段同样基于最大池化方法聚合多尺度的体素特征，来学习点云的局部特征。目前，现有算法大都基于 PointNet[5]及其变种[3]对点云进行特征提取和基于置换不变特性的最大池化法聚合局部点云特征，没有充分考虑点云之间的几何关系。当遇到点云稀疏情况，例如距离较远时，仅依靠局部特征聚合难以学习到更鲁棒的目标特征。为了进一步提高特征的表达能力以改善检测效果，需要对点和点之间的关系进行建模。

Transformer[6]架构在自然语言处理领域取得了显著成功，其将输入的文本序列切分成多个单独的词或字符，然后通过自注意力机制来学习每个词或字符之间的关系。其置换不变的特性适于对无序的点云数据进行编码。PCT（point cloud transformer）[7]和 Point transformer[8]将 Transformer 应用于点云的分类和分割任务，取得了较好的效果。本文将 Transformer 引入点云目标检测领域，以更好地处理点云数据的无序性和点之间的关联信息，学习更鲁棒的点云特征。

另一方面，在实际复杂环境中，检测效果往往受到多种因素的影响，例如遮挡和噪声等，这些因素可能导致点云数据的质量不稳定，而提升性能的关键在于从稀疏点云中提取更鲁棒的特征。此外，点云的稀疏性导致在人工标注数据时易受到环境因素影响，从而使数据标签含有模糊信息，并对学习目标点云的鲁棒特征造成影响。传统方法[9]通常把回归目标当作一个固定值，而忽略了标签不确定性可能造成的影响，限制了检测性能的进一步提升。同时，如果目标包含的点较少，那么围绕目标的候选框位置的不确定性就会增加，如图 1 所示。对于尺寸相同的目标，由于其包含的点云的稀疏性，可能会产生不同的回归目标，从而对检测性能产生不利影响。为解决标签不确定性问题，本文引入了一种基于概率分布的软回归损失。通过检测模块预测候选框位置的不确定性，并将其作为回归损失的一部分，在训练过程中重新量化预测框与其对应标签的相似度，从而提升模型的检测性能。

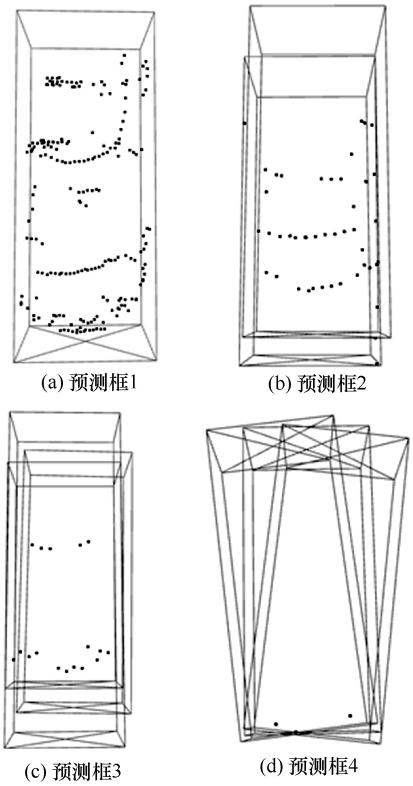


图 1　数据标注中的不确定性

综上所述，本文提出了一种基于原始点云网格自注意力机制的二阶段三维目标检测方法 GT3D。该方法在第二阶段采用基于 Transformer 的自注意力机制来对第一阶段得到的感兴趣区域（RoI）内部的点云进行上下文编码，能够更有效地学习点云之间的依赖关系，提取更鲁棒的目标特征。同时，考虑到数据标注过程中的不确定性对回归任务的影响，使用基于概率分布的回归损失重新度量预测框和真实标签的相似性，降低由数据标注过程带来的标签歧义问题。在公开的三维目标检测数据集 KITTI[10]上对本文所提方法进行评估，结果显示，本文所提方法比现有目标检测方法具有竞争力的性能优势。此外，本文将 KITTI 测试集检测结果提交至 KITTI 官网进行验证，并公开实验结果。

1　相关工作

按照从非结构化点云中提取特征的方式划分，现有的三维目标检测方法主要分为三类：基于体素

的方法、基于点的方法以及点和体素融合的方法。

基于体素的方法通过将点云划分成规则网格，并利用三维卷积技术来提取特征。例如，Zhou 等[9]提出 VoxelNet，首先将点云进行体素化，然后对这些体素进行特征编码，并应用三维卷积来提取特征，最后将这些特征压缩到鸟瞰视角（BEV, bird's eye view）以生成候选框。Yan 等[2]提出 SECOND，通过设计专门针对点云特征提取的三维稀疏卷积模块，有效地提升了三维卷积的处理效率。为了进一步提高三维目标检测的效率，Lang 等[11]提出 PointPillars，该方法直接将特征压缩至 BEV 中来生成候选框，从而避免了三维卷积的过程。然而，基于体素的方法在进行体素特征编码的过程中可能会丢失点云的精确位置信息，限制了方法性能的提升。

基于点的方法使用原始点云进行检测，并且由于点的数量众多，它们通常采用多层次的采样和特征聚合。PointNet[5]和 PointNet++[3]通常被用作这类方法的基础网络。PointRCNN[12]将点云分为前景点和背景点，并在前景点上生成高质量的候选框。3DSSD[13]利用欧氏距离和特征距离进行分层点采样，以获取更多的前景点，并去除了效率较低的上采样和细化步骤，从而在准确性和效率之间取得了良好的平衡。BADet[14]通过将每个候选区域视为一个节点来构建局部图，从而显式地利用边界间的相关性来优化候选框。CIA-SSD[15]引入了一个置信度修正模块，以解决目标定位精度与类别置信度之间的不一致问题，从而获得更加精确的边界框和类别置信度预测。PDV（point density-aware voxel）[16]则为点云引入密度信息，并使用 Transformer 对点进行编码。基于点的方法需要在原始点云中进行分层采样，这通常会导致较低的处理效率。

有很多研究尝试融合点和体素各自的优势来进行检测。例如，CT3D[17]在使用三维体素特征生成区域建议的同时，利用逐通道的 Transformer 从原始点中提取特征。同样，PV-RCNN 引入了体素集抽象模块，使用三维体素特征生成建议后，利用点特征进行特征精细化。后续的工作尝试通过引入新的特征提取方法来改进第二阶段，例如 RefinerNet[18]和 VectorPool[19]。然而，将点和体素的特征相融合在加强检测性能的同时，不可避免地增加了内存的占用，并对检测效率产生影响。在这种将点和体素相结合的主干网络中，特征的整合通常取决于具体的特征转换机制，这可能会导致额外的计算负担。需要注意的是，这类方法虽然在检测精度上往往超过纯粹基于体素的方法，但通常以增加推理过程的时间开销为代价。

Transformer 架构在自然语言处理领域已取得显著成功，其核心模块自注意力机制能够对输入序列间的相关性进行建模。DETR[20]将 Transformer 应用到图像目标检测领域，并把目标检测当作集合预测问题来处理，为使用 Transformer 进行目标检测建立了新的范式。接着，DETR 的一个变种——Deformable DETR[21]，引入了可变形注意力模块，以提升 DETR 的训练效率。文献[7-8]则将 Transformer 应用于点云的特征提取。但是，由于点的数量较多，直接将 Transformer 应用到点云中可能会导致计算复杂度过高、检测效率难以提高的问题。

2 本文模型

GT3D 是一个两阶段的三维目标检测模型，第一阶段用于生成 RoI，第二阶段则利用原始点云来精细化特征，以充分保留点云的空间信息。图 2 展示了 GT3D 的框架，输入为原始点云。首先，通过三维主干网络生成包含目标的 RoI。然后，对每个 RoI 进行网格化，并对 RoI 内部的原始点云进行采样。接着，对采样点的空间信息进行建模，并输入多头 Transformer 中进行上下文编码。最后，将编码后的 RoI 特征输入检测头中，以进行候选框的分类和回归。

2.1 基于体素法的三维主干网络

虽然体素化会带来点云空间信息的损失，但是检测方法在第一阶段主要关注如何快速找到包含目标的 RoI。考虑到体素法具有较高的处理效率，本文使用基于体素法的 SECOND 作为第一阶段的主干网络，并基于多尺度体素特征生成 RoI。具体来说，输入原始点云 $p_i=\{x_i,y_i,z_i,r_i\}$，$i\in[1,n]$，其中 x_i、y_i、z_i 为点云的三维空间坐标，r_i 为反射率，n 为点的数量。然后将点云进行体素化处理，对点云空间进行等间距划分。对于每个体素所包含的点，使用 PointNet 对其进行升维处理，记作 $f_i=\{a_{(i,1)},a_{(i,2)},\cdots,a_{(i,m)}\}$，$i\in[1,n]$，$m\in[1,k]$，其中 k 为点映射到高维空间后的维度。接着，通过最大池化函数对每个体素内的点进行特征聚合。最后，使用多层子流形卷积和稀疏卷积[2]对体素进行特征提取，如图 3 所示，并将提取到的特征 f_v 沿 z 轴压缩到 BEV，输入区域建议网络（RPN, region proposal network）中生成 RoI，其中，k 为卷积核尺寸，pad 为填充操作，s 为步长。

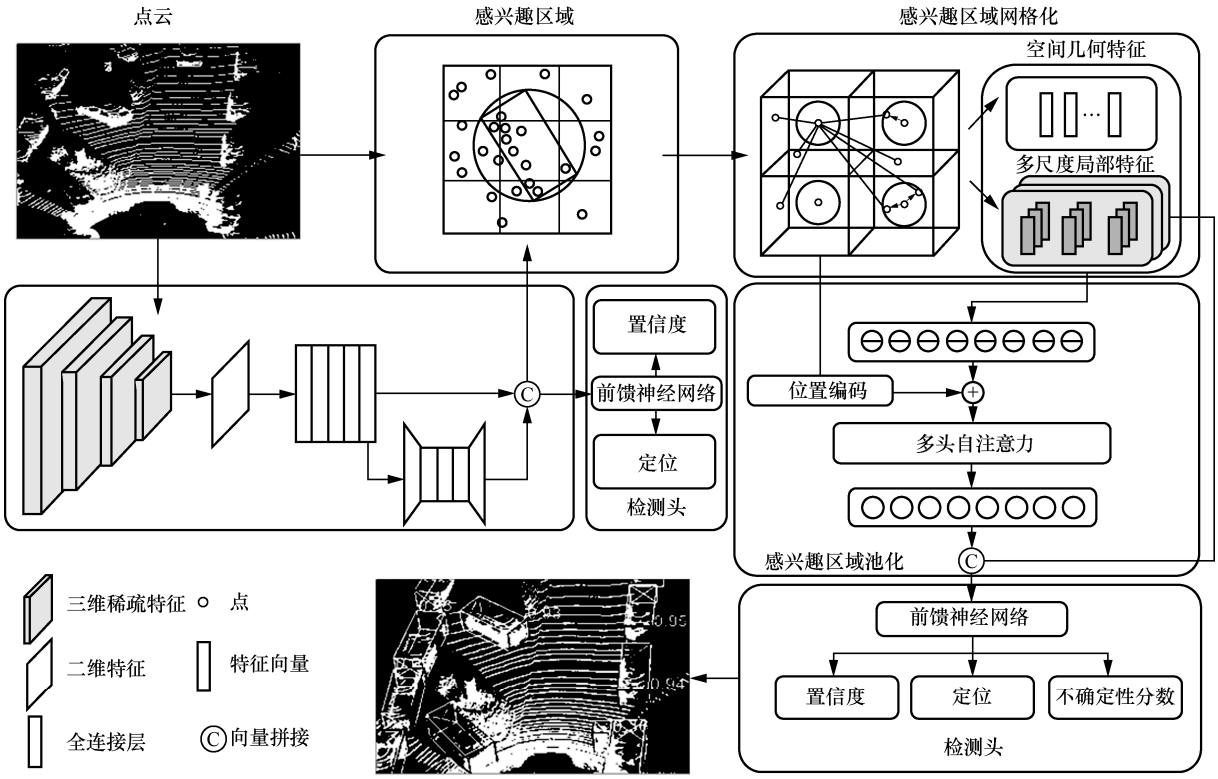


图 2　GT3D 的框架

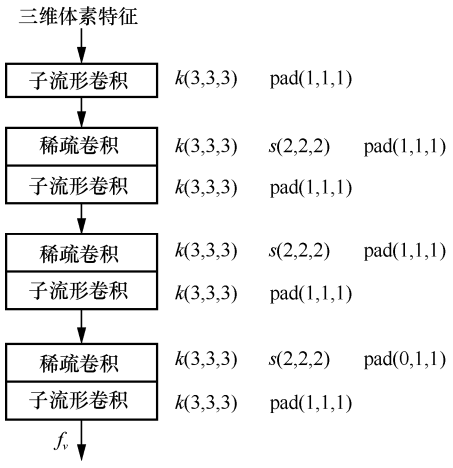


图 3　三维主干网络结构

2.2　网格特征编码

为了更准确地提取点云的局部特征，本文采用两步策略对点云进行有效编码。第一步，采用最远点采样对 RoI 内的点进行采样，并对 RoI 进行网格化处理。计算采样点到每个网格中心的距离，以增强采样点的空间信息。第二步，对网格中心点的局部特征进行聚合。通过使用 PointNet++来聚合网格中心点附近的多尺度局部特征，能够进一步增强中心点的特征表达能力。

2.2.1　网格中心点位置编码

本文对每个 RoI 应用最远点采样策略。值得注意的是，目标的真实框与 RoI 之间可能在角度和位置上有差异。在特定情况下，例如当目标位于树木下或紧邻突出的建筑物时，如果不限制采样空间的高度，可能会导致目标的采样点数量减少，从而对检测结果产生不利影响。为了在最大程度上采样到真实框内的点，同时减少对检测效果不利的背景点的采样，本文采用圆柱体空间结构来对 RoI 进行采样，如图 4 所示。

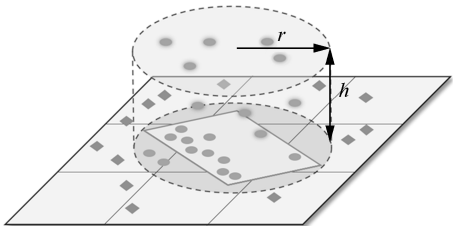


图 4　RoI 点采样

图 4 中，圆形点表示采样区点，方形点表示非采样区点。圆柱体的底部半径为 $r=\alpha\sqrt{\left(\frac{w_r}{2}\right)^2+\left(\frac{l_r}{2}\right)^2}$，高度为 $h=\beta h_r$，其中，w_r、

l_r、h_r 分别表示 RoI 的宽、长和高，α 和 β 表示柱体的扩张比例参数。在这个区域内对点云进行采样，本文设定采样点的数量为 256。如果 RoI 内的点数少于 256，则重复进行随机采样，直到达到 256 个点。本文将 α 设置为 1.1，β 设置为 1。

定义 $P=\{p_1,p_2,p_3,\cdots,p_n\}\subset R^n$，其中 p_i ($i\in[1,n]$) 表示点云中的点坐标，R^n 表示通过 RPN 生成的 RoI。那么，该区域内点 p_i 到任意点 p_j 的距离为

$$d_f(p_i,p_j)=\sqrt{(p_i-p_j)^2} \tag{1}$$

首先，从点云中随机选取一个点 p_0 作为起始点，然后利用式(1)计算其他 $n-1$ 个点与 p_0 的距离 $d_1,d_2,\cdots,d_{(n-1)}$，并将距离 p_0 最远的点 p_m 放入采样点集合 S 中。然后，计算剩余点与采样点集合 S 中所有点的距离，选择到所有采样点的距离最远的点加入采样点集合 S 中。重复这个过程，直到采样点的数量达到预定值。

通过实验发现，对空间点的几何特征进行建模可以增强点的特征表达能力。基于此，本文提出一种新的坐标位置编码方法，用于精细化点的空间位置信息，如图 5 所示。首先，将 RoI 划分为均匀网格，网格数量设置为 $6\times6\times6$（长、宽、高 3 个方向），则每个 RoI 包含 216 个网格。然后，定义每个网格的中心点为 g_m（m 表示 RoI 内的网格索引），并计算每个网格中心点到采样点的相对距离 $\Delta d_i=g_m-p_i$，$m\in[1,216]$，$i=[1,256]$。使用 Δd_i 对网格点的空间位置进行建模并统一位置编码的坐标尺度，最终得到 g_m 的位置特征 f_d。具体计算方式为

$$f_d=g\left(\Delta d_{(i,1)},\Delta d_{(i,2)},\cdots,\Delta d_{(i,m)}\right) \tag{2}$$

$$\Delta d_{(i,m)}=\left\{\Delta x_{i,m},\Delta y_{i,m},\Delta z_{i,m},\Delta f_{i,m}\right\} \tag{3}$$

其中，$g(\cdot)$ 表示特征变换函数（这里使用前馈神经网络（FFN, feed forward network）将距离特征映射到高维特征空间），$\Delta x_{i,m}$、$\Delta y_{i,m}$ 和 $\Delta z_{i,m}$ 分别表示点 p_i 到每个网格中心点的欧氏距离的 3 个分量，$\Delta f_{i,m}$ 表示点的额外特征，包括反射率等。

与 PointPillars 所采用的柱体特征编码（PFE, pillar feature encoding）方法不同，本文通过将采样区域网格化，并计算采样点到每个网格中心点的相对距离，以实现对点的空间位置信息更精细的表达，而 PFE 则是通过计算点与每个柱体的中心点的相对距离来强化点坐标的空间位置信息，精细程度有所欠缺。

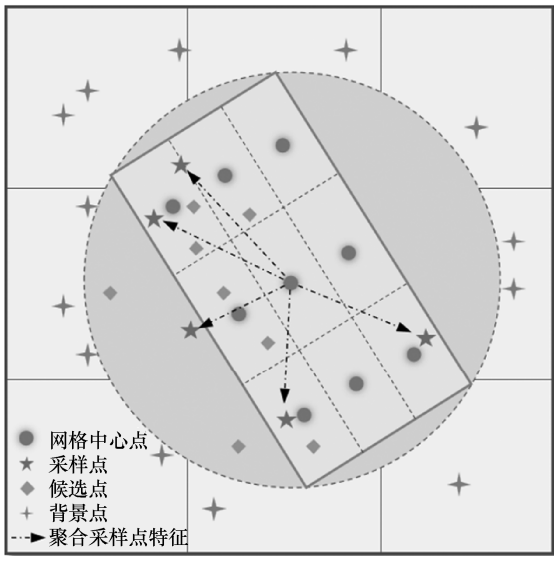


图 5 网格中心点坐标编码

2.2.2 网格中心点多尺度局部特征编码

考虑到原始点云包含更准确的空间信息，本文利用原始点云对网格点进行多尺度局部信息编码。具体而言，对于每个网格的中心点 g_m，查询其周围半径为 r 的球形区域内的点，并使用 PointNet 对这些点进行升维处理，以获得该网格中心点在指定半径内的所有点的特征集合 $f_g^r=\left\{f_1^r,f_2^r,\cdots,f_k^r\right\}$，其中 k 表示该半径范围内的点的数量，如图 6 所示。为了满足置换不变性要求，本文使用最大池化函数对特征进行聚合，从而得到该中心点在特定半径下的特征。

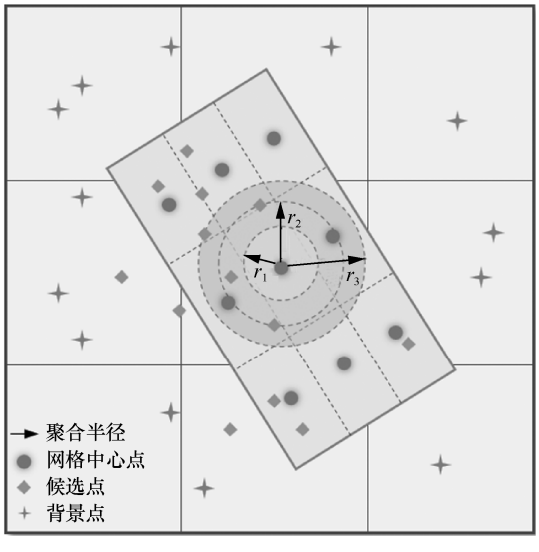


图 6 网格中心点多尺度局部特征编码

$$f_g^{r_i*} = \text{maxpool}\left(G(f_i^r)\right),\ i \in [1,k] \tag{4}$$

其中，$G(\cdot)$ 表示聚合函数，这里采用向量拼接来进行处理。然后，通过调整球查询半径大小，获得中心点在不同尺度下的特征表达。最后，将多尺度特征进行拼接处理，得到最终的中心点局部特征

$$f_g = G(f_g^{r_i*}),\ i \in [1,n] \tag{5}$$

多尺度局部特征编码模块如图 7 所示。本文设定了多个不同尺寸的半径来对点进行聚合。由于不同半径内的点数量可能不同，本文对每个半径内的点的数量进行统一限制：如果点的数量超过规定值，则进行随机选取；如果点的数量低于规定值，则使用点坐标的平均值进行填充；如果该半径内没有点，则使用 0 进行填充。然后，通过三层 FFN 对聚合后的坐标进行升维，并利用最大池化函数对各个尺度的特征进行聚合。最终，通过 FFN 调整 f_g 的维度，并将位置编码特征与多尺度局部特征进行相加，得到网格中心点特征

$$f_{\text{grid}} = \text{ReLU}\left(f_d + \text{FFN}(f_g)\right) \tag{6}$$

其中，f_{grid} 表示 RoI 的空间几何特征和点云多尺度局部特征。

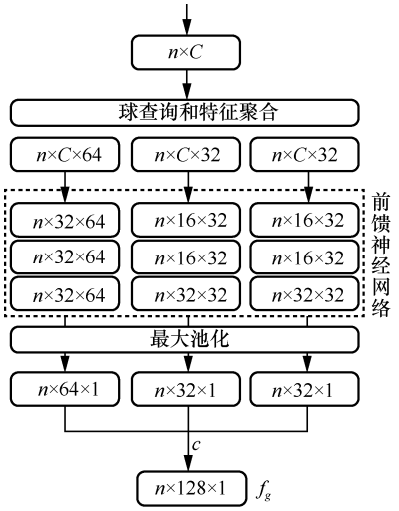
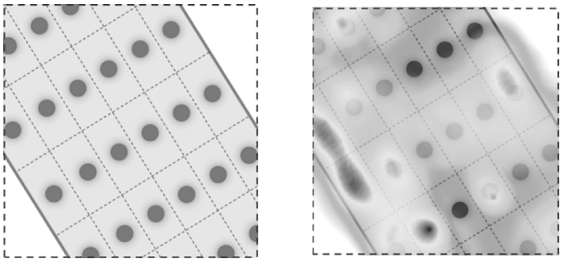


图 7　多尺度局部特征编码模块

2.2.3　空间上下文编码

虽然每个网格编码了目标的空间特征和多尺度局部特征，但仍然缺乏对网格点之间相互依赖关系的建模。为解决此问题，本文引入自注意力机制来捕捉网格点间的远程依赖关系，为网格点的特征赋予不同的权重。这使算法能够捕捉到网格点特征与 RoI 之间更加复杂的关系。图 8 展示了通过自注意力机制加权后的网格点特征对 RoI 特征的贡献度，其中亮度较高的区域表示对 RoI 特征的贡献权重较大。



(a) 原始网格点特征　　(b) 注意力加权后网格点特征

图 8　自注意力机制对网格点特征加权

Transformer 在处理点云数据方面展现出显著的效果，但由于包含大量线性运算，常常伴随较高的计算成本和内存消耗。针对这个问题，本文选择不对 RoI 内的原始点云直接进行注意力编码，而是采纳局部注意力策略，即通过在网格中心点聚合点云的空间和局部特征以降低输入特征的维度。此外，这种策略也使本文提出的两阶段细化方法能够适于不同密度的点云数据。

在 Transformer 的编码阶段，本文对网格点的特征进行注意力编码计算。假设输入特征为 $f_G = \left[f_{\text{grid}}^1, f_{\text{grid}}^2, \cdots, f_{\text{grid}}^i\right]$，$i \in [1,n]$，且 $f_{\text{grid}}^i \neq 0$。没有特征的空网格则不参与注意力编码，仅保留其位置编码。本文采用网格中心点的原始坐标作为位置编码

$$f_{\text{pos}} = g\left(p_{\text{grid}}^i\right),\ i \in [1,m] \tag{7}$$

接着，使用标准 Transformer 编码器计算特征注意力矩阵

$$F_i = f^i{}_{\text{grid}} + f^i{}_{\text{pos}} \tag{8}$$

$$\boldsymbol{K}_i = W_k \odot F_i \tag{9}$$

$$\boldsymbol{Q}_i = W_q \odot F_i \tag{10}$$

$$\boldsymbol{V}_i = W_v \odot F_i \tag{11}$$

$$A_i = \text{softmax}\left(\frac{\boldsymbol{Q}_i \boldsymbol{K}_i^{\mathrm{T}}}{\sqrt{d_q}}\right) \tag{12}$$

其中，W_k、W_q 和 W_v 分别为线性映射函数，d_q 为矩阵 $\boldsymbol{Q}_i$ 的特征维度，$\odot$ 为点乘运算。本文采用多头自注意力机制来处理 $\boldsymbol{K}_i$、$\boldsymbol{Q}_i$ 和 $\boldsymbol{V}_i$，以捕获 RoI 更丰富的特征。多头注意力的计算式为

$$A_{\text{grid}}^i = \text{FFN}\left(\text{concat}\left(A_i \boldsymbol{V}_i\right)\right) \tag{13}$$

其中，concat(·) 用来将多头注意力特征进行拼接，

FFN 用来对特征进行维度变换。

接着，在网格空间位置编码与注意力编码之间构建类似于残差连接的结构，将点的空间位置编码和注意力特征进行拼接，以增强特征的表达能力。经过 FFN 处理后，得到最终的 RoI 特征

$$f_i = \mathrm{FFN}\left(\mathrm{ReLU}\left(A^i_{\mathrm{gird}} + f^i_d\right)\right) \tag{14}$$

最后，将 f_i 输入检测头进行候选框的分类和回归。

2.3 软回归损失

本文提出的软回归损失函数可用于量化预测候选框与其对应的标签之间的相似度，以减轻点云数据标注过程中的不确定性。首先，用高斯分布来表示预测框的位置，并将其所对应的标签视为该分布中的概率，计算式为

$$p(G \mid D_R) = p(G \mid N(\mu,\sigma)) \tag{15}$$

其中，$G = \{g_x, g_y, g_z, g_l, g_w, g_h, g_\theta\}$ 表示候选框所对应的真实标签值；p 为概率密度；$N(\cdot)$ 为二维高斯分布，可表示为

$$N(\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{16}$$

其中，μ 和 σ 表示高斯分布中的均值和方差。本文将检测头对候选框位置的预测 $\{\mu_x, \mu_y, \mu_z, \mu_l, \mu_w, \mu_h, \mu_\theta\}$ 作为 μ，并在检测头部增加一个额外的分支来预测不确定性得分，可表示为 $\{\sigma_x, \sigma_y, \sigma_z, \sigma_l, \sigma_w, \sigma_h, \sigma_\theta\}$，分别对应 μ 中每个位置的不确定分数。在计算出真实标签在预测框分布中的概率后，使用 softmax 函数对这些概率进行归一化处理

$$p_s = \mathrm{softmax}\left(p\left(G \mid D_R\right)\right) \tag{17}$$

最后，使用 p_s 对回归目标进行加权。值得一提的是，本文所提出的软回归损失仅在训练阶段使用，以辅助训练检测头的回归分支，而不会在推理阶段增加额外的计算成本。

2.4 检测头与损失

算法的损失分为RPN损失 L_{rpn} 和细化阶段损失 L_{rcnn} 两部分，其中 L_{rpn} 包括框的置信度损失 L_{cls} 和位置回归损失 L_{reg}。框的编码格式为 $(x, y, z, w, l, h, \theta)$，其中，$x$、$y$、$z$ 表示框的中心点坐标，w、l、h、θ 分别表示框的宽、长、高、朝向角度。真实框与候选框之间位置的误差 $\left(x^*, y^*, z^*, d_{x^*}, d_{y^*}, d_{z^*}, \theta^*\right)$ 为

$$\begin{aligned}
&x^* = \frac{x_g - x_r}{d}, y^* = \frac{y_g - y_r}{d}, z^* = \frac{z_g - z_r}{d_{z_r}},\\
&d_{x^*} = \lg\left(\frac{w_g}{w_r}\right), d_{y^*} = \lg\left(\frac{l_g}{l_r}\right), d_{z^*} = \lg\left(\frac{h_g}{h_r}\right),\\
&\theta^* = \theta_g - \theta_r
\end{aligned} \tag{18}$$

其中，下标 g 表示训练集中真实框的参数，下标 r 表示候选框参数，$d = \sqrt{(w_a)^2 + (l_a)^2}$。

对于 L_{rpn}，使用交叉熵函数来计算置信度损失，以平衡正、负样本对损失的贡献程度

$$L_{\mathrm{cls}} = -c_b \lg(\hat{c}_b) - (1 - c_b)\lg(1 - \hat{c}_b) \tag{19}$$

其中，$\hat{c}_b$ 为预测置信度，c_b 为真实标签值。

框位置回归使用 smooth-L1 损失函数

$$L_{\mathrm{reg}} = \sum_i p_s L_{\text{smooth-L1}}\left(\hat{\gamma}_b, \gamma_b\right) \tag{20}$$

其中，$\hat{\gamma}_b$ 表示边界框的预测残差值，γ_b 表示预测框距离真实框位置的残差值，i 表示正样本的数量。

最后得到总的 L_{rpn} 损失为

$$L_{\mathrm{rpn}} = \beta_1 L_{\mathrm{cls}} + \beta_2 L_{\mathrm{reg}} \tag{21}$$

其中，β_1 和 β_2 为损失的权重系数，用于平衡分类和回归对 L_{rpn} 的贡献程度。

L_{rcnn} 的计算方式和 L_{rpn} 类似，最后得到算法总损失为

$$L_{\mathrm{loss}} = L_{\mathrm{rpn}} + L_{\mathrm{rcnn}} \tag{22}$$

3 实验结果与分析

为验证本文所提方法的有效性，使用公开的自动驾驶数据集 KITTI 对其进行验证，并进行充分的消融实验，以分析 GT3D 各模块的有效性。KITTI 数据集包含 7 481 个训练样本和 7518 个测试样本。与 Chen 等[22]的工作保持一致，将训练样本划分为 3 712 个训练样本集和 3 769 个验证样本集。本文分别在验证集和测试集中对简单、中等和困难 3 个难度等级的目标进行实验，使用平均准确率（AP, average accuracy）衡量所提方法性能。

3.1 实验硬件环境

表 1 为实验所需的软硬件环境及其参数配置。

表 1　实验所需的软硬件环境及其参数配置

环境	参数
CPU	AMD EPYC 7543 32-Core Processor 15 核
GPU	NVIDIA A40
显存	48 GB
操作系统	Ubuntu 20.04
Python 版本	3.8
深度学习框架	PyTorch 1.8.1
CUDA 版本	11.1
cuDNN 版本	8.0
代码编辑环境	Visual Studio Code 1.66.1

3.2　实验细节

对于 KITTI 数据集，其 x 轴检测范围为[0,70.4] m，y 轴为 $[-40.0,40.0]$ m，z 轴为 $[-3.0,1.0]$ m，每个体素块在 x、y、z 这 3 个方向上的大小设置为 $(0.05,0.05,0.1)$ m。每个体素在 3 个方向上的大小均为 0.05 m，体素数量在训练阶段设置为 16 000，推理阶段设置为 40 000。为避免目标包含点云数量太少以至于难以提取到特征，对点数少于 20 的目标进行过滤[2]。本文采用与 SECOND 相同的数据增强方法，具体包括：1) 增加场景中待检测目标的数量；2) 对点云场景中的点按照范围在 $[0.95,1.05]$ 内的随机倍数进行缩放，范围在 $\left[-\frac{\pi}{4},\frac{\pi}{4}\right]$ 内的随机角度进行旋转；3) 对所有真实框在 $\left[-\frac{\pi}{2},\frac{\pi}{2}\right]$ 范围内进行随机角度旋转来模拟目标转向；4) 将点云沿 x 轴进行随机翻转。

RoI 网格数量设置为 $6\times6\times6$，网格点的球形查询半径 $r=[0.2,0.4,0.6]$ m，以此来聚合多尺度的局部特征，每个半径内采样点数量为 $[32,32,64]$，分别被编码为 $[32,32,64]$ 维向量，最后每个网格中心点的局部特征编码共 128 维。Transformer 的头部数量为 4，dropout 设置为 0.1，隐含层数量为 3。

在训练阶段，使用 8 个 NVIDIA A40 GPU 对整个网络进行端对端训练，对于 KITTI 数据集，batch size 设置为 6，使用 Adam_onecycle 优化器训练 80 个 epoch，学习率最大值为 0.001，使用 one-cycle 策略和余弦退火策略[23]对学习率进行更新。在训练阶段，RoI 数量设置为 128，测试阶段设置为 100。

算法的损失由基于 Focal Loss[24]的分类损失和基于 smooth-L1 的回归损失组成。其中，分类损失和回归损失的权重比例设置为 1:1。在后处理阶段，使用非极大值抑制算法来去除冗余框，交并比（IoU, intersection over union）阈值设置为 0.1，置信度阈值为 0.3。其他网络参数选择 OpenPCDet 工具箱中提供的默认值。

训练损失曲线如图 9 所示，其中，rpn_loss 表示第一阶段损失，rcnn_loss 表示第二阶段损失。第一阶段、第二阶段以及总训练损失的曲线在训练初期下降较快，但随着迭代次数的增加，损失的下降速度逐渐变慢，最后趋于平稳，这表示模型已经收敛。

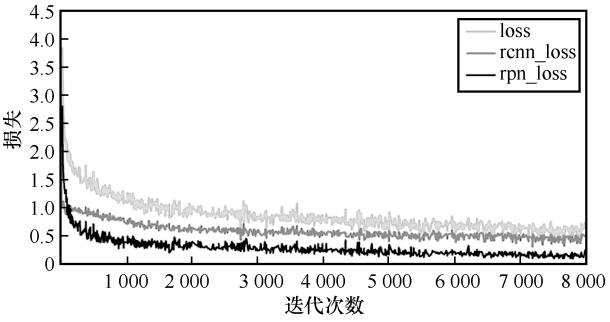


图 9　训练损失曲线

3.3　与其他算法对比

本文在 KITTI 验证集和测试集上将 GT3D 方法与其他先进的三维目标检测方法进行了比较和分析。对于汽车类别，设定 IoU 阈值为 0.7，本文分别给出了所提方法在 11 个和 40 个召回位置上的平均准确率。此外，将 GT3D 的测试结果提交至 KITTI 在线测试服务器，并将结果公开，如表 2 所示，所有实验结果均来自 KITTI 官方基线。

为保证公平，本文基于 40 个召回位置来计算测试集的平均准确率。在 KITTI 测试集上，GT3D 在 3 种不同难度等级上分别达到了 91.45%、82.76%和 79.74%的检测准确率，特别是在简单和困难等级汽车检测上显示出优势。这说明本文所提方法在检测准确性和泛化能力上表现良好。在评估方法的推理速度时，本文采用每秒帧数（FPS, frame per second）作为评价标准，本文所提方法达到了每秒 15 帧的检测速度，这显示 GT3D 在检测准确率和推理效率之间实现了良好的平衡（表 2 中，‘–’表示该方法未公开代码和推理速度）。

表 2 不同方法在 KITTI 测试集上对汽车的检测性能对比

模式	方法	简单	中等	困难	FPS
Image+Point Cloud	F-PointNet[25]	82.19%	69.79%	60.59%	5.9
	ContFuse[26]	83.68%	68.78%	61.67%	16.7
	PointSIFT+SENet[27]	85.99%	72.72%	64.58%	—
	UberATG-MMF[28]	88.40%	77.43%	70.22%	12.5
	3D-CVF at SPA[29]	89.20%	80.05%	73.11%	13.3
	CLOCs[30]	88.94%	80.67%	77.15%	—
Point-based	PointRCNN[12]	86.96%	75.64%	70.70%	10
	STD[31]	87.95%	79.71%	75.09%	12.5
	3DSSD[13]	88.36%	79.57%	74.55%	26.3
	CT3D[17]	87.83%	81.77%	77.16%	14.3
	PV-RCNN[1]	90.25%	81.43%	76.82%	8.9
Voxel-based	VoxelNet[1]	77.47%	65.11%	57.73%	4.4
	SECOND[2]	83.34%	72.55%	65.82%	30.4
	PointPillars[11]	82.58%	74.31%	68.99%	42.4
	Voxel R-CNN[4]	90.90%	81.62%	77.06%	25.2
	FV2P[32]	88.17%	81.81%	77.43%	8
	Focals Conv[33]	90.20%	82.12%	77.50%	8.9
	GraR-Vo[34]	91.29%	82.77%	77.20%	25.6
	PDV[16]	90.43%	81.86%	77.36%	10
	SA-SSD[35]	88.75%	79.79%	74.16%	25
	3D Cascade RCNN[36]	90.46%	82.16%	77.31%	14.2
	GT3D	91.45%	82.76%	79.74%	15

表 3 展示了不同方法在 KITTI 验证集上对汽车的检测性能对比，其中检测结果基于 11 个召回位置计算，IoU 阈值为 0.7。实验结果表明，GT3D 在 3 种不同难度汽车类别检测中分别达到了 89.78%、86.31%和 79.22%的准确率，相比其他先进方法表现出显著的提升，进一步验证了 GT3D 的有效性。这是因为 Transformer 在特征提取方面具有强大的能力，使模型能够有效地学习不同点云稀疏度下目标的特征。

为了进一步评估 GT3D 的性能，表 4 展示了不同方法在 KITTI 验证集上对自行车的检测性能对比，其中准确率基于 40 个召回位置计算。实验结果表明，本文所提方法在检测效果上具有较强的竞争力，展示出良好的性能。

表 3 不同方法在 KITTI 验证集上对汽车的检测性能对比

模式	方法	简单	中等	困难
Point-based	PointRCNN[12]	88.88%	78.63%	77.38%
	STD[31]	89.70%	79.80%	79.30%
	3DSSD[13]	89.71%	79.45%	78.67%
	CT3D[17]	89.54%	86.06%	78.99%
	PV-RCNN[1]	89.35%	83.69%	78.70%
Voxel-based	VoxelNet[1]	81.97%	65.46%	62.85%
	SECOND[2]	88.61%	78.62%	77.22%
	PointPillars[11]	86.62%	76.06%	68.91%
	Voxel R-CNN[4]	89.41%	84.52%	78.93%
	Focals Conv[33]	89.52%	84.93%	79.18%
	PDV[16]	89.52%	84.93%	79.18%
	SA-SSD[35]	90.15%	79.91%	78.78%
	GT3D	89.78%	86.31%	79.22%

表 4 不同方法在 KITTI 验证集上对自行车的检测性能对比

方法	简单	中等	困难
CT3D[17]	91.99%	71.60%	67.34%
Voxel R-CNN[4]	91.28%	72.54%	68.46%
PV-RCNN[1]	88.88%	71.95%	66.78%
PDV[16]	92.72%	74.23%	69.60%
GT3D	92.93%	74.65%	69.71%

表 5 展示了不同方法在参数量方面的对比，并提供了不同模型在 KITTI 测试集上对汽车的检测平均准确率，其中，mAP 为 KITTI 测试集上对汽车的检测平均准确率。从表 5 可以看出，尽管 GT3D 的参数量在两阶段方法中处于中等水平，但其平均准确率明显优于其他方法。这表明 GT3D 在有效提升检测效果的同时，没有显著增加参数规模。

表 5 不同方法在参数量方面的对比

阶段数	方法	mAP	参数量
单阶段	SECOND[2]	73.90%	5.33×10^6
	PointPillars[11]	75.29%	4.83×10^6
两阶段	PointRCNN[12]	77.77%	4.04×10^6
	3DSSD[13]	80.83%	7.56×10^6
	CT3D[17]	82.25%	7.83×10^6
	PV-RCNN[1]	82.83%	13.12×10^6
	Voxel R-CNN[4]	83.19%	7.59×10^6
	GT3D	84.65%	7.95×10^6

3.4　可视化分析

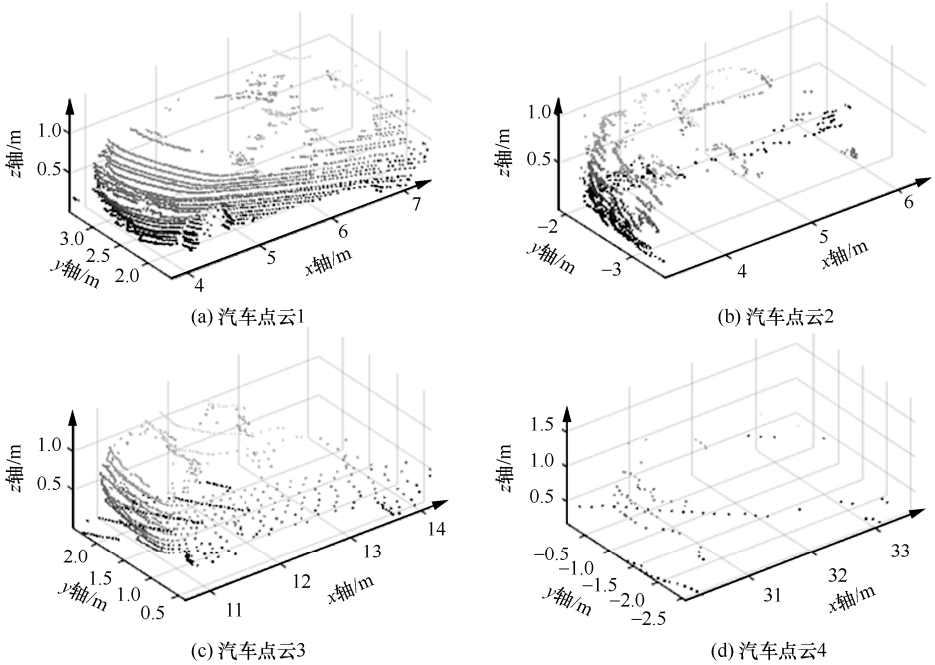
本文对 GT3D 方法的检测效果进行了可视化分析，如图 10 所示。通过比较方法输出的预测框（虚线）与真实框（实线）的位置来验证模型的检测效果。为了清晰展示，在 3 个场景中分别展示了相机和点云的视角。第一行展示场景的相机视角，第二行展示场景的点云视角和检测结果，第三行展示将检测到的目标框映射回相机视角的效果。

由可视化结果可知，GT3D 在汽车类别上的检测准确率较高，如场景①所示，所有汽车都被成功检测到。在场景②中，尽管距离较远的汽车包含的点云数量较少，但仍然能被准确地检测到，甚至检测到了数据集中没有标注的汽车。对于场景③，该场景较复杂，包含的背景点较多，然而，GT3D 依然能够正确识别被遮挡的远处汽车。这表明本文通过使用均匀网格点来描述点云的空间特征，以及利用多尺度局部特征，对遮挡区域进行了有效的特征增强。

图 11 展示了模型检测到的汽车点云。其中，x 轴、y 轴和 z 轴表示以激光雷达传感器为原点的坐标系，坐标轴上的数值表示点云场景中的全局坐标。从图 11 可以看出，左上角、右上角和左下角的汽车的点云较密集，而右下角的汽车包含的点云较

(a) 场景①　　(b) 场景②　　(c) 场景③

图 10　GT3D 可视化结果



(a) 汽车点云1　　(b) 汽车点云2

(c) 汽车点云3　　(d) 汽车点云4

图 11　模型检测到的汽车点云

少。尽管如此，GT3D 仍然能够准确地进行检测。这说明 GT3D 在检测点云数量较少的目标时具有较强的鲁棒性。

为了进一步验证 GT3D 的性能，本文将 GT3D 与当前经典方法 Voxel R-CNN 和 PDV 进行比较，实验结果如图 12 所示。从图 12 中可以看出，Voxel R-CNN 和 PDV 在一些情况下出现误检，例如，在 BEV 视角下，由于左侧墙壁的点云较复杂，使 Voxel R-CNN 和 PDV 将其误检为汽车，而 GT3D 展示了较强的鲁棒性，对于复杂场景中的目标误识别率较低，取得了较好的检测效果。

3.5　消融实验

为了进一步验证 GT3D 中模块的有效性，本文针对网格中心点位置编码模块、多尺度特征聚合模块和软回归损失进行了消融实验，对每个模块训练 80 个 epoch。使用 40 个召回位置进行评估，IoU 阈值设置为 0.7，实验结果如表 6 所示。其中，A.L.F. 表示多尺度特征聚合模块，P.E.表示网格点位置编码模块，T.R.表示 Transformer 模块，S.L.R.表示软回归损失。

方法 1 不包含两阶段检测头，而是仅依靠 BEV 下的 RoI 特征来进行检测，其平均准确率较低，这突显了单阶段方法的局限性。

方法 2 在方法 1 的基础上加入了网格点的多尺度特征聚合模块，并采用 Transformer 对网格点进行注意力编码。与方法 1 相比，方法 2 在简单、中等和困难 3 个难度级别车辆检测中，准确率分别高约 1.16%、0.84%和 1.20%，这表明通过聚合多尺度的局部特征可以提高检测准确率，并且进一步证明了两阶段细化在提升模型性能上的重要性。

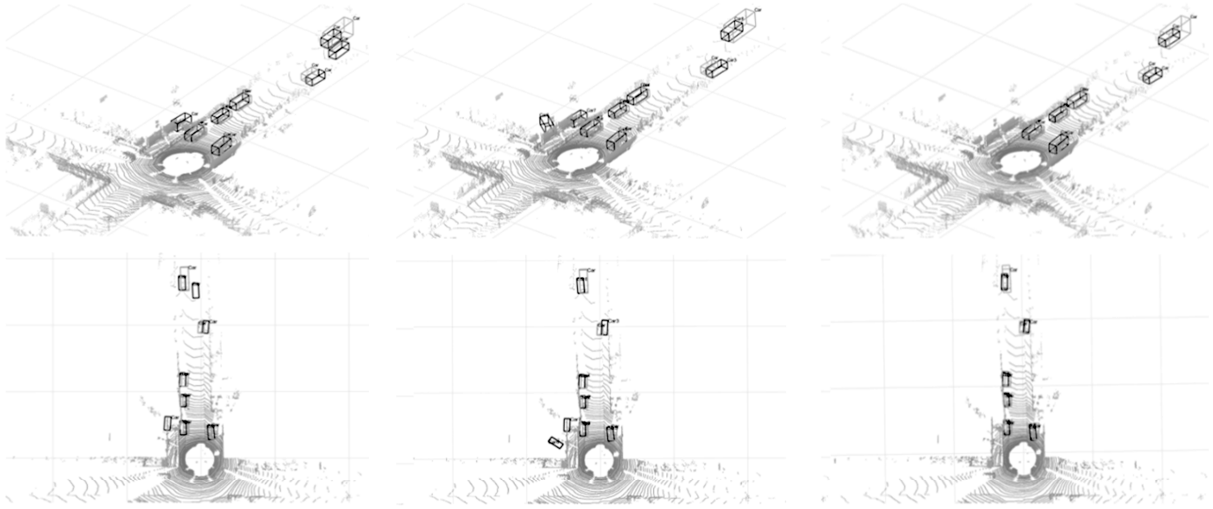
表 6　　GT3D 消融实验

方法	A.L.F.	P.E.	T.R.	S.L.R	简单	中等	困难
方法 1	—	—	—	—	90.20%	80.81%	77.79%
方法 2	√	—	√	—	91.36%	81.65%	78.99%
方法 3	—	√	√	—	92.35%	84.75%	82.35%
方法 4	√	√	√	—	92.59%	85.31%	83.26%
方法 5	√	√	√	√	92.71%	85.45%	83.28%

方法 3 在方法 1 的基础上加入了网格点位置编码模块，并同样采用 Transformer 对网格点进行注意力编码。与方法 1 相比，方法 3 在简单、中等和困难 3 个难度级别车辆检测中，准确率分别高约 2.15%、3.94%和 4.56%。这表明，通过显式地对点的空间位置进行编码可以有效提升模型的性能。

方法 4 在方法 1 的基础上，同时加入了多尺度特征聚合模块和网格点位置编码模块。与方法 1 相比，方法 4 在简单、中等和困难 3 个难度级别车辆检测中，准确率分别高约 2.39%、4.50%和 5.47%，这表明两者的结合使用可以进一步提升模型的性能。

方法 5 在方法 1 的基础上加入了多尺度特征聚合模块、网格点位置编码模块和软回归损失。与方法 1 相比，方法 5 在简单、中等和困难 3 个难度级别车辆检测中，准确率分别高约 2.51%、4.64%和 5.49%。实验结果表明，加入软回归损失后，模型



(a) Voxel R-CNN　　(b) PDV　　(c) GT3D

图 12　Voxel R-CNN、PDV 和 GT3D 的检测结果可视化对比

性能得到有效提升，证明软回归损失在消除数据标注过程中引入的模糊性方面具有显著效果。

4　结束语

目前，基于原始点云的三维目标检测技术正在迅速发展，本文研究了两阶段检测方法在性能提升方面的关键技术，其中包括空间坐标的显式建模、多尺度局部特征聚合以及基于自注意力机制的特征编码。为此，本文提出了一种两阶段基于自注意力机制的三维目标检测方法 GT3D。该方法将 RPN 生成的 RoI 在空间上划分为均匀网格点，并对这些点进行多尺度局部特征聚合和空间位置编码。然后，采用自注意力机制对网格点特征进行编码，以获取更加有效的 RoI 特征。最后，通过软回归损失来消除数据标注过程引入的模糊性，从而进一步提升检测性能。接下来，本文将继续研究聚合多尺度信息的两阶段检测方法，进一步提高基于体素的三维目标检测方法准确率，拟考虑引入额外辅助网络对模型进行监督，以借助自注意力机制进一步提高目标检测的准确性。

参考文献：

[1] SHI S S, GUO C X, JIANG L, et al. PV-RCNN: point-voxel feature set abstraction for 3D object detection[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 10526-10535.

[2] YAN Y, MAO Y X, LI B. SECOND: sparsely embedded convolutional detection[J]. Sensors, 2018, 18(10): 3337.

[3] QI C R, YI L, SU H, et al. PointNet++: deep hierarchical feature learning on point sets in a metric space[J]. arXiv Preprint, arXiv: 1706.02413, 2017.

[4] DENG J J, SHI S S, LI P W, et al. Voxel R-CNN: towards high performance voxel-based 3D object detection[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI Press, 2021: 1201-1209.

[5] CHARLES R Q, HAO S, MO K C, et al. PointNet: deep learning on point sets for 3D classification and segmentation[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2017: 77-85.

[6] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proceedings of the 31st Conference on Neural Information Processing Systems. Massachusetts: MIT Press, 2017: 5998-6008.

[7] GUO M H, CAI J X, LIU Z N, et al. PCT: point cloud transformer[J]. Computational Visual Media, 2021, 7(2): 187-199.

[8] ZHAO H S, JIANG L, JIA J Y, et al. Point transformer[C]//Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2022: 16239-16248.

[9] ZHOU Y, TUZEL O. VoxelNet: end-to-end learning for point cloud based 3D object detection[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2018: 4490-4499.

[10] GEIGER A, LENZ P, URTASUN R. Are we ready for autonomous driving? the KITTI vision benchmark suite[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2012: 3354-3361.

[11] LANG A H, VORA S, CAESAR H, et al. PointPillars: fast encoders for object detection from point clouds[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 12689-12697.

[12] SHI S S, WANG X G, LI H S. PointRCNN: 3D object proposal generation and detection from point cloud[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 770-779.

[13] YANG Z T, SUN Y N, LIU S, et al. 3DSSD: point-based 3D single stage object detector[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 11037-11045.

[14] QIAN R, LAI X, LI X R. BADet: boundary-aware 3D object detection from point clouds[J]. Pattern Recognition, 2022, 125: 108524.

[15] ZHENG W, TANG W L, CHEN S J, et al. CIA-SSD: confident IoU-aware single-stage object detector from point cloud[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI Press, 2021: 3555-3562.

[16] HU J S K, KUAI T S, WASLANDER S L. Point density-aware voxels for LiDAR 3D object detection[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2022: 8459-8468.

[17] SHENGA H L, CAI S J, LIU Y, et al. Improving 3D object detection with channel-wise transformer[C]//Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2022: 2723-2732.

[18] CHEN Y L, LIU S, SHEN X Y, et al. Fast point R-CNN[C]//Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2020: 9774-9783.

[19] SHI S S, JIANG L, DENG J J, et al. PV-RCNN++: point-voxel feature set abstraction with local vector representation for 3D object detection[J]. International Journal of Computer Vision, 2023, 131(2): 531-551.

[20] CARION N, MASSA F, SYNNAEVE G, et al. End-to-end object detection with transformers[C]//European Conference on Computer Vision. Cham: Springer, 2020: 213-229.

[21] ZHU X, SU W, LU L, et al. Deformable DETR: deformable transformers for end-to-end object detection[C]//Proceedings of International Conference on Learning Representations. Singapore: OpenReview.net Press, 2021: 1-16.

[22] CHEN X, KUNDU K, ZHU Y, et al. 3D object proposals for accurate object class detection[C]//Proceedings of the 28th Conference on Neural Information Processing Systems. Massachusetts: MIT Press, 2015: 424-432.

[23] LOSHCHILOV I, HUTTER F. SGDR: stochastic gradient descent with warm restarts[J]. arXiv Preprint, arXiv: 1608. 03983, 2016.

[24] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection[C]//Proceedings of IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2017: 2999-3007.

[25] QI C R, LIU W, WU C X, et al. Frustum PointNets for 3D object

detection from RGB-D data[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2018: 918-927.

[26] LIANG M, YANG B, WANG S L, et al. Deep continuous fusion for multi-sensor 3D object detection[C]//European Conference on Computer Vision. Cham: Springer, 2018: 663-678.

[27] ZHAO X, LIU Z, HU R, et al. 3D object detection using scale invariant and feature reweighting networks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI Press, 2019: 9267-9274.

[28] LIANG M, YANG B, CHEN Y, et al. Multi-task multi-sensor fusion for 3D object detection[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 7337-7345.

[29] YOO J H, KIM Y, KIM J, et al. 3D-CVF: generating joint camera and LiDAR features using cross-view spatial feature fusion for 3D object Detection[C]//European Conference on Computer Vision. Cham: Springer, 2020: 720-736.

[30] PANG S, MORRIS D, RADHA H. CLOCs: camera-LiDAR object candidates fusion for 3D object detection[C]//Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Piscataway: IEEE Press, 2020: 10386-10393.

[31] YANG Z T, SUN Y N, LIU S, et al. STD: sparse-to-dense 3D object detector for point cloud[C]//Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2019: 1951-1960.

[32] LI J L, DAI H, SHAO L, et al. From voxel to point: IoU-guided 3D object detection for point cloud with voxel-to-point decoder[C]//Proceedings of the 29th ACM International Conference on Multimedia. New York: ACM Press, 2021: 4622-4631.

[33] CHEN Y K, LI Y W, ZHANG X Y, et al. Focal sparse convolutional networks for 3D object detection[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2022: 5418-5427.

[34] YANG H H, LIU Z L, WU X P, et al. Graph R-CNN: towards accurate 3D object detection with semantic-decorated local graph[C]//European Conference on Computer Vision. Cham: Springer, 2022: 662-679.

[35] HE C H, ZENG H, HUANG J Q, et al. Structure aware single-stage 3D object detection from point cloud[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 11870-11879.

[36] CAI Q, PAN Y W, YAO T, et al. 3D cascade RCNN: high quality object detection in point clouds[J]. IEEE Transactions on Image Processing: a Publication of the IEEE Signal Processing Society, 2022, 31: 5706-5719.

[作者简介]

鲁斌（1975− ），男，宁夏银川人，博士，华北电力大学教授，主要研究方向为智能计算与计算机视觉、综合能源系统与大数据分析。

孙洋（1991− ），男，河北保定人，华北电力大学博士生，主要研究方向为机器学习、计算机视觉。

杨振宇（1998− ），男，内蒙古呼和浩特人，华北电力大学博士生，主要研究方向为机器学习、计算机视觉。