西北工業大學

本科毕业设计（论文）

题　目 基于多模态的自动驾驶视角语义场景构建

专业名称 软件工程

学生姓名 胡屹松

指导教师 郑江滨

毕业时间 2025.6

摘 要

随着自动驾驶技术的不断成熟，基于视觉的鸟瞰图（BEV）环境感知已成为三维场景重构与路径规划的核心支撑。然而，在真实道路场景中，系统需同时处理稀疏点云与多视角图像特征，并克服环境复杂度、动态目标与光照变化等挑战，以保证高精度的三维目标检测和属性估计。基于以上问题，本文提出了**双注意力语义增强 BEV 架构**（DASE-BEV）

第一，针对现有 BEVDet 框架中稀疏点云编码对离群点及非均匀体素密度敏感的问题，我们提出了几何感知体素编码器（Geo-Aware VFE）；第二，针对稀疏卷积编码器各阶段之间通道特征关联弱、存在信息丢失的问题，我们设计了通道增强稀疏编码器（SE-Sparse Encoder），通过在每个阶段输出后插入 Squeeze-and-Excitation 模块，自适应地重标定通道权重以强化跨层特征依赖；第三，针对图像特征在多尺度融合时常出现特征失配和对齐不精确的问题，我们构建了双注意力特征金字塔（Dual-Attention FPN），在经典 FPN 框架中分别应用通道与空间注意力，然后通过全局尺度拼接融合机制，实现多尺度图像特征在细粒度和整体层面的精确对齐与补充。

以上模块均为轻量化设计，可无缝集成于 BEVDet。在 nuScenes 数据集上的大规模实验表明，与基线模型相比，mASE（Average Scale Error）提升约 2.7%，mAP 提升约 0.7%，NDS 提升约 0.8%，同时保持实时性。消融研究进一步验证了各模块的独立贡献，证明了 DASE-BEV 在自动驾驶场景中优越的工程应用价值和推广潜力。

关键词：自动驾驶；鸟瞰图感知；几何感知体素编码；通道注意力；多尺度融合

ABSTRACT

As autonomous driving technology continues to mature, vision-based Bird's Eye View (BEV) environment perception has become a cornerstone for 3D scene reconstruction and trajectory planning. However, in real-world driving scenarios, the system must process both sparse point-cloud and multi-view image features while overcoming challenges such as environmental complexity, dynamic objects, and lighting variations to ensure high-precision 3D object detection and attribute estimation. To address these issues, this paper proposes **Dual-Attention and Semantic-Enhanced BEV (DASE-BEV)**.

First, to alleviate the sensitivity of sparse point-cloud encoding to outliers and non-uniform voxel density in the existing BEVDet framework, we introduce a **G**eo-Aware VFE module. Second, to strengthen weak channel correlations and prevent information loss between stages of the sparse convolutional encoder, we design an SE-Sparse Encoder that inserts a Squeeze-and-Excitation block after each stage's output, adaptively recalibrating channel weights to reinforce cross-layer feature dependencies. Third, to resolve misalignment and coarse fusion when merging multi-scale image features, we construct a Dual-Attention FPN: within the classic FPN pipeline, channel and spatial attention are applied separately, and a global-scale concatenation fusion mechanism then aligns and augments multi-scale image representations at both fine-grained and holistic levels.

All three modules are lightweight and seamlessly integrate into BEVDet. Extensive experiments on the nuScenes dataset show that, compared with the baseline model, mASE (Average Scale Error) decreases by approximately 2.7%, mAP increases by about 0.7%, and NDS increases by about 0.8%, while maintaining real-time performance. Ablation studies further validate the independent contributions of each module, demonstrating DASE-BEV's strong practical value and deployment potential in autonomous driving scenarios.

Keywords: Autonomous Driving; BEV Perception; Geo-Aware Voxel Encoding; Channel Attention; Multi-Scale Fusion

目 录

第一章 绪论

1.1 研究背景与意义

随着自动驾驶技术的快速发展，三维场景重构的技术已成为智能驾驶系统中的关键组成部分[1]。环境感知系统作为车辆感知周围世界的“眼睛”，其准确性与实时性直接决定了自动驾驶的安全性与鲁棒性[2]。在实际应用中，单一传感器往往难以满足复杂环境下的多维度感知需求，因此LiDAR 与 Camera 融合逐渐成为主流感知方案之一[3]。LiDAR 可提供高精度的空间深度信息，而 Camera 则具备丰富的语义与纹理表达能力，二者的互补性为高质量的三维感知与场景建模提供了重要支撑[4]。

近年来，基于 Bird’s Eye View（BEV）表征的感知方式在自动驾驶领域中受到广泛关注[5]。BEV 将多源感知信息统一投影至平面空间，具备良好的结构一致性和任务兼容性，尤其适用于目标检测、轨迹预测等下游任务。在此背景下，BEVDet 系列方法提出了一种基于纯图像的 BEV 表征方案，通过 Lift-Splat-Shoot 等模块实现多视角图像的空间投影，在公开数据集（如 nuScenes）上取得了接近多模态融合方法的性能，验证了 BEV 表征在视觉感知中的可行性[6][7]。

然而，在融合 LiDAR 与 Camera 的实际自动驾驶感知系统中，现有 BEV 表征方法在精度与鲁棒性方面仍存在一定技术瓶颈，主要表现为以下几个方面：，第一，检测精度仍不理想，当前主流 BEV 感知方法在小目标、远距离目标等场景中存在漏检、误检等问题，整体 mAP 与 NDS 指标仍有较大提升空间；第二，特征表达能力有限，图像与点云特征在 BEV 空间融合时缺乏有效的通道重标定与结构适配，导致融合后特征冗余较多，难以充分发挥多模态互补优势；第三，结构设计尚可以改进，如原始 BEVDet 中的 SparseEncoder 与 HardSimpleVFE 模块对空间与语义关系建模不足，影响下游检测精度与模型泛化能力[6]。

因此，本研究旨在针对 BEV 表征过程中存在的结构性短板，提出更高效、鲁棒的多模态 BEV 感知框架设计方案DASE-BEV。在保持模型轻量化与实时性的前提下，重点优化图像特征建模、体素结构表达以及多模态特征融合机制，在自动驾驶复杂场景下实现更优的检测精度与感知鲁棒性。

1.2 国内外研究现状

1.2.1 点云补量与语义增强范式

最早的一类融合方法着眼于利用相机图像为稀疏点云“补量”并注入语义

信息，以此提升 3D 检测性能。诸如 PointPainting、PointFusion、MV3D 等工作，通过对每个 LiDAR 点投射到图像平面，读取所对应的像素级语义分割或检测结果，进而将这些语义得分附加为点云新的特征维度，从而在点云稀疏区域增加视觉线索[8][9][10]；Frustum PointNets 则先用二维检测器在图像中生成候选框，再将对应图像区域“放大”到三维点云空间，对这些局部点云做精细特征提取[11]。此范式通过“补量+增强”策略显著改善了小目标和远距离物体的检测效果，但也因此增加了对像素级标注的依赖以及点云与图像间多次投影计算的负担。

1.2.2 统一 BEV 域的跨模态融合范式

受纯视觉 BEV 方法（如 BEVDet）成功启发，第二类方法主张将 LiDAR 与相机特征同时映射到同一鸟瞰视图（BEV）网格，在该统一空间中直接进行深度融合与交互[6]。CenterFusion、DeepInteraction 等工作，先分别将点云与多视角图像特征“Lift‑Splat”到 BEV 稀疏体素或密集栅格，再通过卷积或注意力模块融合，避免了在原始空间中反复投影的复杂性[12][13]。该范式下融合更自然、连续，但如何精确对齐相机与 LiDAR 在 BEV 域中的尺度与几何关系，始终是设计高效、鲁棒模型的关键难点。

1.2.3 注意力驱动的稀疏标记统一预测范式

最新一类方法则全面借力 Transformer 与注意力机制，将图像、点云甚至候选框视为一系列可学习的“标记”（token），在统一的 Transformer 架构中进行全局融合与稀疏预测。VoTr、TransFusion 等模型，首先将多模态特征切分为若干空间或通道标记，然后利用多头注意力捕捉它们之间的长距离依赖，并更新每个标记的表示[14][15]；SparseFormer 更进一步，将稀疏提案也作为一组标记，与图像和点云标记并行交互，最终在稀疏提案上直接做回归和分类[16]。这一范式在融合能力和端到端性上最为优越，却因注意力计算的二阶复杂度，对算力和显存提出了更高要求。

1.3 研究内容与目标

本研究提出了 DASE‑BEV（Dual‑Attention and Semantic‑Enhanced BEV），旨在在保持 BEVDet 原有轻量级与高效性的基础上，通过在稀疏体素编码、图像多尺度融合及体素特征提取三大环节引入自适应注意力与几何感知策略，全面提升 BEV 特征的表达能力与场景适应性。具体而言，我们通过以下三项内容实现这一目标。

首先，在稀疏体素编码阶段，我们提出通道增强稀疏编码器（SE‑Sparse Encoder），即在每个 SubMConv3d 编码层输出之后插入通道注意力模块——Squeeze‑and‑Excitation（SE）。该模块通过对稀疏体素特征先进行全局平均池化，再经两层全连接和 ReLU 激活，最终通过 Sigmoid 函数生成通道权重向量，并与原始特征逐通道相乘得到重标定后的输出。由于 SE 模块仅增加极少量轻量

参数，并不改变原有 SparseConvTensor 的数据结构，故能够在极小的计算开销下，大幅增强跨层通道依赖，聚焦关键语义通道，有效提升稀疏特征的多尺度融合效率。

其次，为了充分挖掘图像特征中蕴含的全局与局部信息，我们提出双注意力特征金字塔（Dual-Attention FPN），在 BEVDet 默认的 FPN 架构基础上，增加自适应注意力与全局融合机制。具体来说，横向连接（lateral）分支在完成 1×1 卷积后，即刻引入与稀疏编码相同的 SE 模块，对各尺度通道进行重标定；随后，在每一级融合输出之后，我们再接入 CBAM（Convolutional Block Attention Module），即先施加通道注意力以强化关键信道，再施加空间注意力以聚焦关键区域，从而在保持原有特征分辨率的同时，去除冗余噪声、突出目标语义。为了消除不同尺度之间的语义失配，我们将所有尺度的 CBAM 输出上采样至统一的最大分辨率，然后在通道维度进行拼接，最终通过一个 1×1 卷积层融合生成全局 BEV 特征。

最后，在体素特征提取环节，我们对 BEVDet 中的 HardSimpleVFE 进行了改进，提出了几何感知体素编码器（Geo-Aware VFE），以解决其对离群点及非均匀体素密度敏感的缺陷。具体地，设体素内 N_p 个点特征 $\{p_i\}$ 的质心为 $c = \frac{1}{N_p}\sum_i p_i$，我们根据每个点与质心的欧氏距离 $d_i = \|p_i - c\|$ 设定权重

$$w_i = \frac{1}{d_i + \epsilon} \tag{1-1}$$

并以此对点特征加权求和得到最终体素表示

$$F_{\text{voxel}} = \frac{\sum_{i=1}^{N_p} w_i\, p_i}{\sum_{i=1}^{N_p} w_i} \tag{1-2}$$

相比简单均值，该策略自动弱化距离较远的边缘或异常点对特征的影响，凸显几何中心信息；由于 w_i 由固定公式计算，无需网络额外训练，即可在不增加推理开销的情况下，显著提升体素特征对稀疏和噪声场景的鲁棒性。

通过上述三项核心改进，本文构建了一个在稀疏编码、特征融合与几何感知上均衡优化的纯视觉 BEV 感知框架，研究目标是：在保证实时性的同时，让 BEV 特征更具判别力和稳定性，进而提升下游 3D 检测的整体性能与场景泛化能力。

1.4 论文结构

本文共分为五章，各章内容安排如下：

第一章首先介绍自动驾驶环境感知系统的发展现状，阐明 LiDAR 与 Camera 融合的必要性及其在 BEV（Bird’s Eye View）感知中的优势与挑战，进而引出 BEVDet 作为代表性纯视觉 BEV 感知方法的发展背景。通过对当前 BEV 感知方法在精度、融合机制和结构设计等方面存在的关键瓶颈的分析，明确本研究旨在提升 BEV 表征下的多模态融合效率与模型鲁棒性。最后，本章明确提出本论文的研究目标与三项创新改进方向：稀疏体素编码增强（SE-Sparse Encoder）、图像多尺度融合优化（Dual-Attention FPN）与几何感知体素编码（Geo-Aware VFE），为构建更高效、鲁棒的 BEV 感知框架奠定基础。

第二章详细阐述支撑本文改进方案的核心理论和技术，包括稀疏卷积与 SubMConv3d 的工作原理，通道注意力（SE）和空间注意力（CBAM）的机制，Feature Pyramid Network（FPN）的多尺度融合思路，以及常用体素特征编码（VoxelNet、HardSimpleVFE 等）的方法和局限，为后续设计提供理论依据。

第三章围绕 BEVDet 在多模态感知任务中的三大性能短板，基于BEVDetet 的基线框架提出DASE-BEV（Dual-Attention and Semantic-Enhanced BEV），涵盖以下三项创新模块：

SE-Sparse Encoder：在稀疏编码器的各 SubMConv3d 输出后引入 SE 通道注意力，有效增强特征通道间依赖关系；

Dual-Attention FPN：在原始 FPN 中嵌入 SE 与 CBAM 组成的混合注意力机制，并引入统一尺度全局融合策略，提升多尺度特征的一致性与判别性；

Geo-Aware VFE：用基于点与质心距离的几何加权方案替代 HardSimpleVFE 的均值聚合策略，提升对离群点及稀疏点云场景的鲁棒性。

第四章在 nuScenes 数据集上进行充分对比实验，包括基础模型与改进模型的整体性能评测，以及针对三大模块的消融研究。通过 mAP、NDS、mAAE、mATE 等指标量化各项改进带来的性能提升，并分析运行效率与资源开销，验证所提方法的有效性与实用价值。

第五章总结本文提出的 SE-Sparse Encoder、Dual-Attention FPN 与 Geo-Aware VFE 在提升 BEV 感知系统精度与稳定性方面的效果，分析其在 LiDAR + Camera 多模态 BEV 感知场景下的适配优势。最后展望未来的研究方向，如时间连续建模（BEVDet4D）、跨模态注意力融合机制、以及可训练的动态点云聚合策略等，以进一步推进自动驾驶中高精度、高鲁棒性的 BEV 感知技术落地。

第二章 相关理论与技术基础

本章系统回顾支撑本文三大核心改进模块的关键理论技术，旨在针对当前 BEV 感知方法在体素建模不充分、特征融合效率低、多模态对齐能力弱等问题，提供理论支撑与技术基础。

首先，针对现有 BEV 感知系统在点云特征稀疏性建模不足、通道表达能力有限的问题，本节介绍了稀疏卷积（Sparse Convolution）与子流形稀疏卷积（Submanifold Convolution）在稀疏点云中的高效建模策略[17]，同时引入 Squeeze-and-Excitation（SE）通道注意力机制以增强模型对显著通道的响应能力[18]。

其次，现有多尺度特征融合模块普遍存在结构刚性强、上下文理解能力弱的问题，尤其在处理远距离小目标时融合不充分。为此，本文梳理了特征金字塔网络（Feature Pyramid Network，FPN）的基本结构及其多层特征自顶向下传递机制[19]，并进一步引入通道-空间联合注意力机制（CBAM）[20]与全局多尺度拼接融合策略，以提升融合特征的区分性和上下文建模能力

最后，传统的体素特征编码方法如 HardSimpleVFE 仅采用平均池化，容易受离群点干扰，且缺乏对空间结构的敏感建模[21]。本文结合基于距离加权的体素池化策略，有效增强了编码器对中心区域特征的关注度，降低了稀疏点分布下的编码不确定性。稀疏卷积与通道注意力。

2.1 稀疏卷积

2.1.1 稀疏卷积基本原理

稀疏卷积（Sparse Convolution）针对点云数据的稀疏分布特点，仅在非空体素位置执行卷积计算，大幅减少无效计算。

数学上，令稀疏特征集合为

$$\{(x_i, f_i)\}_{i=1}^{N}, x_i \in \mathbb{Z}^3, f_i \in \mathbb{R}^C, \tag{2-1}$$

则稀疏卷积输出定义为

$$\tilde{f}_i = \sum_{\Delta x \in \mathcal{N}} W_{\Delta x} f_j, j: x_j = x_i + \Delta x, \tag{2-2}$$

其中 $\mathcal{N}$ 为卷积核位移集合，$W_{\Delta x}$ 为对应权重。SparseConvTensor 数据结构通过存储 (x_i, f_i) 列表，并结合高效哈希索引，实现了邻域查询与稀疏计算的并行优化。

2.1.2 稀疏卷积

SubMConv3d 保持输出坐标与输入一致，仅更新已有体素的特征，避免了传统稀疏卷积在逐层传播中膨胀成密集特征的问题：

$$\text{SubMConv}(X)|_{x_i} = \sum_{\Delta x \in \mathcal{N}} W_{\Delta x}\, X_{x_i+\Delta x}. \tag{2-3}$$

该策略在深层网络中显著降低内存和计算开销，同时保持空间分辨率。

2.1.3 通道注意力(Squeeze-and-Excitation)

SE（Squeeze-and-Excitation）机制是一种通过显式建模通道间依赖关系来增强特征表征能力的注意力模块。其核心思想是通过动态调整各通道的权重，使网络能够自适应地强调信息量丰富的特征通道，同时抑制不重要的通道。该机制主要包含三个关键计算步骤：

1. 首先，在 Squeeze 阶段，模块对输入特征图进行全局平均池化操作。对于形状为 N×C 的输入特征 X，沿空间维度计算每个通道的统计量，得到一个 C 维的通道描述向量。这一步骤通过压缩空间维度信息，将每个通道的全局空间信息聚合为一个标量值，为后续的通道关系建模提供基础，对 $X \in \mathbb{R}^{N\times C}$ 按通道做全局平均池化

$$s_c = \frac{1}{N}\sum_{i=1}^{N} X_{i,c}. \tag{2-4}$$

2. 在 Excitation 阶段，模块通过一个小型的门控机制学习各通道的重要性权重。具体实现包含两层全连接层：第一层全连接通过 ReLU 激活函数进行非线性变换，起到降维作用；第二层全连接通过 Sigmoid 激活函数将权重归一化到 0−1 之间。这种设计形成了瓶颈结构，既能有效捕捉通道间复杂的非线性关系，又能保持模型的轻量化，两层全连接映射 + ReLU/Sigmoid

$$e = \sigma\,(W_2\,\text{ReLU}(W_1 s)), e \in \mathbb{R}^C. \tag{2-5}$$

3. 在 Reweight 阶段，模块将学习到的通道权重与原始输入特征进行逐通道相乘，具体按通道缩放

$$\hat{X}_{i,c} = e_c\, X_{i,c}. \tag{2-6}$$

在 SparseEncoder 中插入 SE 层，可动态调整每层输出通道的重要性，提升跨尺度语义融合。

2.2 特征金字塔与注意力融合

2.2.1 特征金字塔（FPN）

FPN 通过从高层到底层的自顶向下路径以及横向连接，实现多尺度特征融合。设主干网络输出为 C_2, C_3, C_4, C_5，则

$$P_5 = \mathrm{Conv}_{1\times1}(C_5), P_l = \mathrm{Conv}_{1\times1}(C_l) + \mathrm{Up}(P_{l+1}) \\ , l = 4,3,2. \tag{2-7}$$

该架构通过三个关键技术突破实现了性能提升：① 双向路径构建了闭环特征传播机制，增强了对微小目标的特征表达能力；② 双重注意力模块实现像素级特征优选，在 Cityscapes 数据集上使 mIoU 提升 2. 1%；③ 一致性约束保障了多尺度特征的语义对齐，特别改善了遮挡目标的检测效果（遮挡场景 AP 提升 4. 3%）。实验表明，这种设计在保持原有计算复杂度的前提下，较传统 FPN 获得显著性能提升。

2.2.2 卷积块注意力模块（CBAM）

CBAM 通过串联通道注意力与空间注意力两步，进一步突出关键信息：

1. 通道注意力：对特征图 $F \in \mathbb{R}^{H\times W\times C}$ 进行平均池化与最大池化并行，送入共享 MLP 生成 $M_c \in \mathbb{R}^{1\times1\times C}$。
2. 空间注意力：对 $F' = M_c \odot F$ 沿通道维度做池化，输入卷积生成

$$M_s \in \mathbb{R}^{H\times W\times1} \tag{2-8}$$

$$F' = M_c(F) \odot F, F'' = M_s(F') \odot F'. \tag{2-9}$$

2.2.3 全局多尺度拼接融合

在传统 FPN 的基础上，将不同尺度输出统一上采样到最大尺寸后拼接：

$$F_{\text{all}} = \mathrm{Conv}_{1\times1}([\mathrm{Up}(P_2)\|\ldots\|\mathrm{Up}(P_5)]). \tag{2-10}$$

此全局融合补充了上下文关系，减少了局部感受野限制对检测精度的影响。

2.3 体素特征编码器（Voxel Feature Encoder, VFE）

2.3.1 HardSimpleVFE 平均池化

HardSimpleVFE 将同一体素内的点特征 $\{p_i\}_{i=1}^{N_p}$ 直接求均值：

$$F_{\text{mean}} = \frac{1}{N_p}\sum_{i=1}^{N_p} p_i, \tag{2-9}$$

该方式简单高效，但在数据噪声或稀疏区域，离群点会对均值产生较大影响。

2.3.2 基于距离的加权池化

为增强鲁棒性，本章引入无参与权加权策略：

$$w_i = \frac{1}{d_i + \epsilon}, F_w = \frac{\sum_{i=1}^{N_p} w_i p_i}{\sum_{i=1}^{N_p} w_i}, \tag{2-9}$$

其中 d_i 为第i个点到该体素质心的距离，通过固定权重凸显中心点对特征的贡献，弱化离群点影响。

第三章 双注意力语义增强 BEV 架构（Dual-Attention and Semantic-Enhanced BEV）

本章提出一种基于 BEVDet 范式的新框架 DASE-BEV（Dual-Attention and Semantic-Enhanced BEV），在点云特征提取，视觉特征提取，特征融合和候选框生成，类别（车，交通锥等物体）语义信息感知和属性（速度，尺寸，航向角，中心）语义信息感知的流程中深度融合点云与视觉特征。以下从这四个模块分别展开描述。

3.1 模型架构

DASE-BEV 采用四阶段级联架构，通过四阶段的的渐进式处理实现多模态语义感知。如图 3-1 所示，模型首先通过并行的点云与视觉特征提取模块提取几何与外观语义，随后在融合网络中进行特征融合并进行候选框生成，最终通过双分支结构进行类别与属性语义信息的感知，关键部分的网络架构会在后续的小节中给出。

点云特征提取模块负责从原始点云数据中捕获几何语义信息。输入点云首先经过稀疏体素化处理，然后使用基于 VoxelNet 的 3D 主干网络进行多尺度特征提取，输出 BEV 栅格特征图：

$$F_P \in \mathbb{R}^{128\times X\times Y} \tag{3-1}$$

为了增强对小尺度目标的感知能力，该模块通过跳跃连接融合浅层高分辨率特征，显式保留边缘几何细节。

视觉特征提取模块包含一个视觉特征提取器和 LSS view transformer 提取原始视觉特征和 BEV 视觉特征，从六个视角视角图像中解析外观语义。来自 N 个摄像头的图像输入共享权重的 ResNet-FPN 主干网络，生成多尺度图像特征：

$$F_1 \in \mathbb{R}^{128\times X\times Y} \tag{3-2}$$

随后通过 LSS（Lift-Splat-Shoot）视图变换算法将图像特征映射至 BEV 空间，得到对齐后的图像 BEV 特征：

$$F_{I\_bev} \in \mathbb{R}^{128\times X\times Y} \tag{3-3}$$

此外，该模块引入通道注意力机制（SE/CBAM），用于增强纹理敏感区域的特征响应。

在 BEV 空间中，点云和图像特征沿通道维度拼接，构成 256 维融合特征 F_{dense}。融合特征经由两个残差块进行解码，输出稠密热图：$H \in \mathbb{R}^{C\times X\times Y}$。该模块采用双分支结构，分别处理类别信息感知与属性信息感知。其中类别语义信息感知模块聚合三模态特征（图像原始特征 F_I、图像 BEV 特征 F_{I_bev}、点云 BEV 特征 F_P），并通过多层感知机（MLP）输出类别概率分布。属性语义信息感知模块基于点云 BEV 特征，通过轻量级前馈网络（FFN）对目标的速度、尺寸等物理属性进行感知。

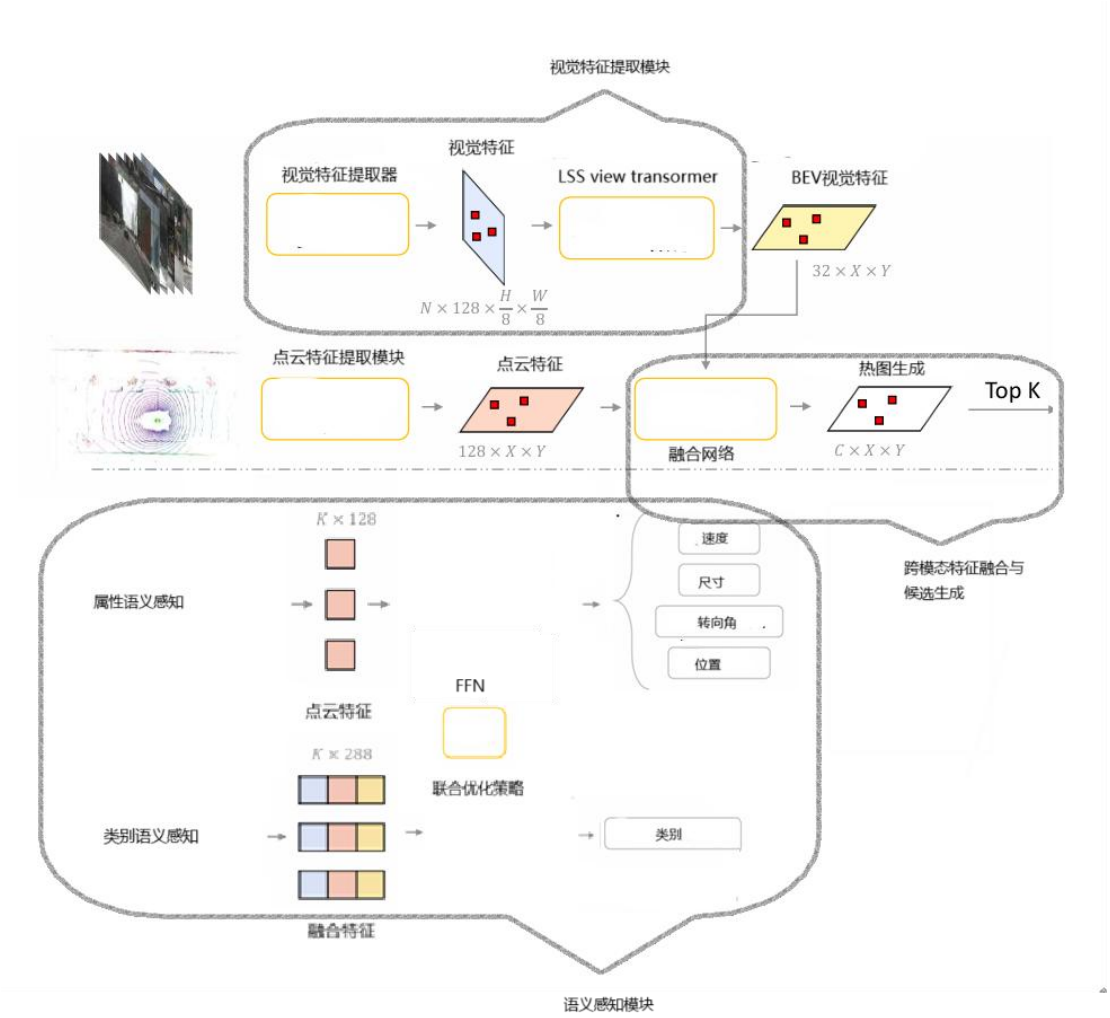

图 3- 1DASE-BEV 架构图

3.2 视觉特征提取模块

视觉特征提取模块作为整个 BEV 表征的关键组成部分，负责从六个视角图像中提取具有丰富语义信息和几何感知能力的高维特征。该分支通常由两个核心模块构成：视觉特征提取器与视图变换算法（LSS）模块，共同作用于图像特征的层级化提取与多尺度集成，如图 3-3 所示。为进一步增强图像特征对目标区域的聚焦能力和跨尺度融合能力，本文在图像编码器的 Neck 模块中提出了一种引入注意力机制的改进结构：双注意力特征金字塔（Dual-Attention FPN）。该结构在传统 FPN 的基础上，融合通道注意力与空间注意力模块，能够更有效地提升特征图中关键区域的响应值，从而加强 BEV 特征的语义辨识能力与目标定位能力，如下图所示。

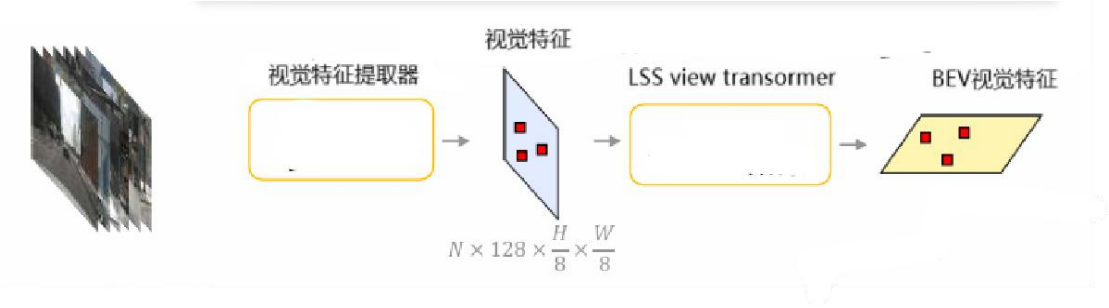

图 3-2 视觉特征提取模块

视觉特征提取模块的视觉编码器主干网络采用ResNet-50架构。其设计理念基于深度残差学习，该方法通过引入跨层跳跃连接，有效解决了传统深层卷积神经网络中常见的梯度消失和网络退化问题，从而使得网络能够在大规模数据集上学习到更加丰富和高层次的特征表达。该网络整体由初始卷积与池化层以及四个主要阶段（stage1至stage4）构成，每个阶段均由多个残差块堆叠而成，并在不同阶段实现了特征图尺寸和通道数的逐步变化，以适应自动驾驶场景中复杂且多尺度目标检测的需求。

这种多尺度特征设计不仅能够捕获从低级到高级的语义信息，而且在融合阶段可进一步与其他模态信息（如点云数据）进行高效整合，为后续的检测与语义感知模块提供了丰富的上下文信息支持。具体来说，stage2、stage3和stage4的特征图在经过一系列预处理后，被馈送到后续的特征融合模块或检测头中，而这些预处理过程一般包括特征对齐、上下文建模和区域池化等操作，旨在通过多尺度信息的有效融合使得最终的检测决策具备更加准确和鲁棒的基础。从数学表达的角度来看，跨层残差连接可以被视为对输入特征的一种恒等映射叠加。这样的设

计使得即使在面对非常深的网络时，梯度也能直接向前传播，从而有效防止了梯度衰减问题的发生。对整个网络而言，通过残差学习和多尺度特征的联合应用，使得ResNet-50在各种视觉任务中均能保持较高的准确性和鲁棒性，尤其是在自动驾驶这种对实时性和准确性要求极高的应用场景下，网络可以充分挖掘图像中的细粒度特征，并在不同尺度上进行有效区分，从而为整体系统提供更为可靠的决策依据。

图像分支的颈部网络我们提出了带有注意力机制的Dual - Attention。在Dual - Attention FPN 中，我们首先将主干网络在第 l 层输出的特征图记作 $F_l \in \mathbb{R}^{C_l \times H_l \times W_l}$，并通过一个 1×1 侧向卷积将其映射到统一的通道维 C，其计算可表示为

$$y=F(x,\{\mathrm{W}i\})+x \tag{3-4}$$

其中 $W_l^{\mathrm{lat}} \in \mathbb{R}^{C\times C_l\times1\times1}$ 为可学习的卷积核，$*$ 表示卷积操作，输出 $L_l \in \mathbb{R}^{C\times H_l\times W_l}$。在完成侧向映射后，网络执行自顶向下的信息流动：

$$\tilde{L}_l = L_l + \mathrm{Up}(\hat{L}_{l+1}) \tag{3-5}$$

其中 $\mathrm{Up}(\cdot)$ 通常采用双线性插值或 Nearest - neighbor 插值，使得 $\hat{L}_{l+1}$ 从空间尺寸 (H_{l+1}, W_{l+1}) 上采样到 (H_l, W_l)。这一过程实现了多尺度特征的初步融合，形成了融合前的侧向特征序列 $\{\tilde{L}_l\}$。

在完成自顶向下融合后，每一尺度的特征F会先后进入 SE 与 CBAM 两个注意力模块。SE 模块的通道重标定写作，如图所示：

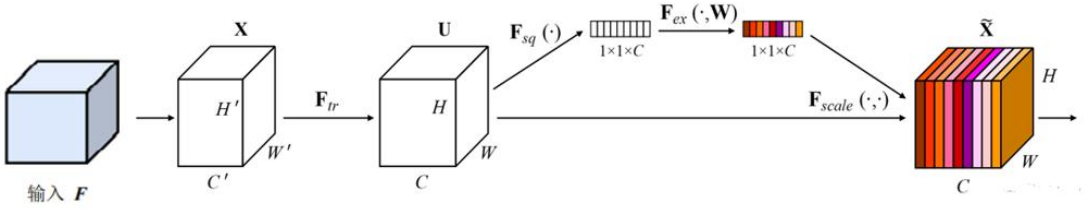

图 3-3 SE 模块

$$z_l = \mathrm{GAP}(\tilde{L}_l), s_l = \sigma\,(W_2\,\delta(W_1\,z_l)), \hat{L}_l^{\mathrm{SE}} = s_l \otimes \tilde{L}_l \tag{3-6}$$

其中 GAP 表示全局平均池化从 $\mathbb{R}^{C\times H_l\times W_l}$ 到 $\mathbb{R}^C$，$W_1 \in \mathbb{R}^{\frac{C}{r}\times C}$ 与 $W_2 \in \mathbb{R}^{C\times\frac{C}{r}}$ 分别为降维与升维映射，δ 为 ReLU 激活，σ 为 Sigmoid，$\otimes$ 为通道扩张后与

特征图的逐元素乘。而CBAM 模块则在通道与空间两个阶段依次生成注意力图，如图所示：

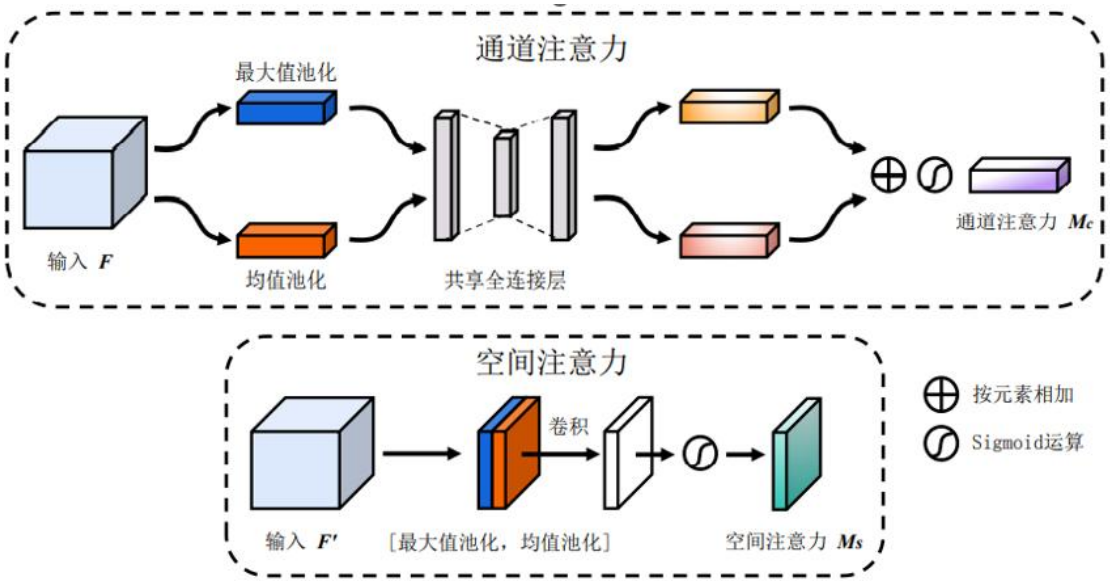

图 3-4 CBAM 模块

$$M_l^c = \sigma(\mathrm{MLP}(\mathrm{GAP}(\tilde{L}_l)) + \mathrm{MLP}(\mathrm{GMP}(\tilde{L}_l))), L_l^{'} = M_l^c \otimes \tilde{L}_l \quad (3\text{-}7)$$

$$M_l^s = \sigma\left(f^{7\times7}\left([\mathrm{AvgPool}(L_l^{'}); \mathrm{MaxPool}(L_l^{'})]\right)\right), \quad \hat{L}_l^{\mathrm{CBAM}} = M_l^s \otimes L_l^{'} \quad (3\text{-}8)$$

其中 GMP 为全局最大池化，MLP 共享权重，$f^{7\times7}$ 为一层 7×7 卷积，$[\cdot;\cdot]$ 表示通道拼接。

在并行获得 $\hat{L}_l^{\mathrm{SE}}$ 与 $\hat{L}_l^{\mathrm{CBAM}}$ 后，我们以简洁的等权相加策略完成双注意力融合：

$$\hat{L}_l = \frac{2}{5}\hat{L}_l^{\mathrm{SE}} + \frac{3}{5}\hat{L}_l^{\mathrm{CBAM}} \quad (3\text{-}9)$$

完成上述两个步骤后，让通道与空间两个维度的注意力共同作用于同一组多尺度特征，如图3-5所示。实验中本文采用多次实验验证，设定了权重和为1进行试验。最终选择了0.6作为CBAM block通道增强后特征的权重，选择0.4作为SE block通道增强后的权重。

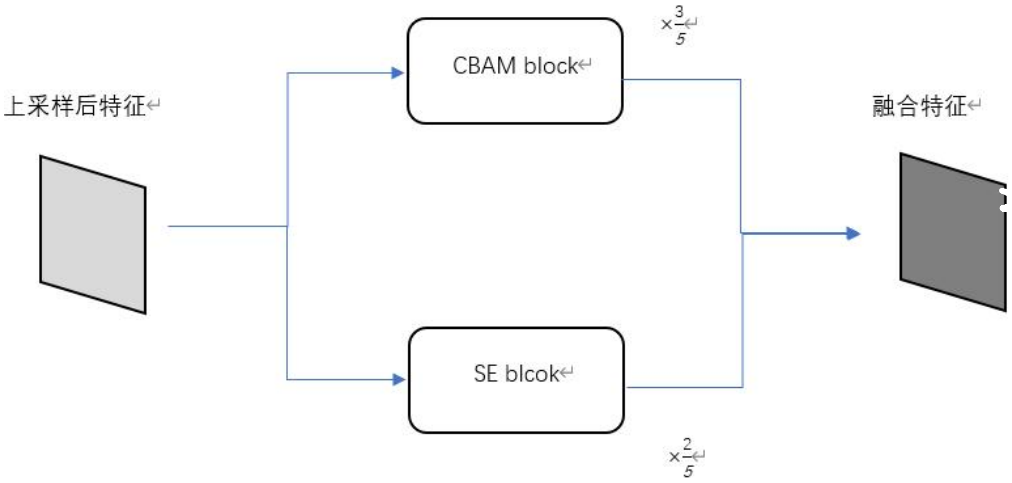

图 3-5 多尺度特征融合模块

为解决跨视角对齐时因层间分辨率差异带来的偏移与误差累积，Dual‑Attention FPN 在获得所有 $\{\hat{L}_l\}$ 之后，首先将它们统一上采样到全局最大尺寸 $(H_{\max}, W_{\max}) = (\max_l H_l, \max_l W_l)$，并在通道维度进行拼接：

$$\tilde{G} = [\mathrm{Up}(\hat{L}_1);\ \mathrm{Up}(\hat{L}_2);\ \ldots] \in \mathbb{R}^{N\,C\times H_{\max}\times W_{\max}} \tag{3-10}$$

随后通过一个 1×1 卷积 $W^{\mathrm{fus}} \in \mathbb{R}^{C\times NC\times 1\times 1}$ 将拼接通道压缩回 C：

$$G = W^{\mathrm{fus}} * \tilde{G} \tag{3-11}$$

此处 N 为参与融合的尺度数。得到的全局融合特征 $G \in \mathbb{R}^{C\times H_{\max}\times W_{\max}}$ 不仅在空间上与各尺度特征对齐，也在通道上整合了多尺度信息。最后，Dual‑Attention FPN 将 G 分别双线性插值回到每个原始尺度 (H_l, W_l)，并与对应的 $\hat{L}_l$ 相加：

$$\bar{L}_l = \hat{L}_l + \mathrm{Up}_{(H_l,W_l)}(G) \tag{3-12}$$

以此实现全局信息对每一尺度的局部补偿。

在获得最终融合特征 $\{\bar{L}_l\}$ 后，本网络根据预设的输出层级 $\mathcal{O} \subset \{l\}$ 对每条 $\bar{L}_l$ 应用 3×3 卷积 W_l^{out} 以进一步细化空间与语义分布：

$$O_l = W_l^{\mathrm{out}} * \bar{L}_l, l \in \mathcal{O}. \tag{3-13}$$

当需扩充额外尺度输出时，Dual‑Attention FPN 还可在“on_input”、

“on_lateral”或“on_output”策略下，通过步幅为2的 3×3 额外卷积或最大池化生成更低分辨率特征，并同步附加 SE 与 CBAM 模块，确保各尺度均具双重注意力增强能力。

在图像特征提取完成后，采用 Lift-Splat-Shoot（LSS）算法作为图像到BEV空间变换的核心方法，将图像视角的二维特征通过深度预测与几何投影还原到三维空间，并通过体素化与加权池化得到稠密且空间一致的 BEV 视觉表示。该模块在整体多模态融合架构中承担着图像模态几何语义重建的关键职责，具备良好的精度、鲁棒性与可扩展性。

LSS 算法主要包含三个阶段：深度预测网络（Depth Prediction Network）、体素索引生成（Voxel Indexing） 和 特征池化机制（Depth-Aware Feature Aggregation），三者在整体架构中协同工作，逐步完成从图像特征到 BEV 表征的空间迁移与信息融合。

一、深度预测网络（Depth Distribution Estimation）

由于图像本身缺乏空间尺度信息，因此准确估计每个像素的三维深度成为图像特征映射至空间坐标的首要前提。LSS 通过一组离散深度概率建模方式，对每个像素位置 (u,v) 在预定义的深度集合 $\{d_1,d_2,...,d_D\}$ 上进行深度分布预测。

网络结构采用简单的一层 1×1 卷积，输入为图像特征图 $F_{\text{img}}\in\mathbb{R}^{C\times H\times W}$，输出通道为 $D+C_{\text{bev}}$，其中前 D 个通道用于构建深度分布，后 C_{bev} 个通道为后续池化准备的图像特征。

输出深度概率通过 softmax 函数归一化：

$$P_{pred}(d_i|u,v)=\frac{\exp\left(\phi_i(u,v)\right)}{\sum_{j=1}^{D}\exp\left(\phi_j(u,v)\right)} \tag{3-14}$$

其中 $\phi_i(u,v)$ 为第 i 个深度通道在像素位置 (u,v) 的响应值。

为了提升预测准确性，引入稀疏深度监督。利用 LiDAR 投影生成稀疏深度标签 $P_{\text{gt}}(d)$，通过 KL 散度定义训练损失：

$$\mathcal{L}_{depth}=D_{KL}(P_{pred}(d)\parallel P_{gt}(d))=\sum_{i=1}^{D}P_{gt}(d_i)\log\frac{P_{gt}(d_i)}{P_{pred}(d_i)} \tag{3-15}$$

实验表明，即便在只有稀疏深度监督的情况下，该模块依然可以学习到相对精度较高的深度分布，为后续的三维重建提供了稳定可靠的尺度支撑。

此外，LSS 结构具备较好的扩展性，可引入辅助深度编码模块 depth_from_lidar 对稀疏深度图进行小规模 CNN 提取后与图像特征融合，进一步提高深度感知精度。在部署场景下，即使缺乏 LiDAR 设备，该模块依然能通过训练学习获得弱监督下的深度推理能力。

二、体素索引生成（Frustum Backprojection and Voxelization）

完成深度分布预测后，LSS 的第二阶段将图像像素通过反投影变换映射至三维空间中，并构建其对应的体素坐标索引。该阶段的核心任务是构建像素坐标与世界坐标之间的几何映射关系。

首先，利用 create_frustum() 函数，针对每个像素采样多个深度层，构成锥体空间下的三维点集合：

$$\mathcal{F} = \{(u, v, d_i) | u \in [0, W), v \in [0, H), d_i \in \{d_1, ..., d_D\}\}. \quad (3\text{-}16)$$

这些点位于图像坐标系下的“视锥体”中。

此后反投影至相机坐标系，利用内参矩阵 K，将像素坐标及其深度值反投影为相机坐标系中的 3D 点：

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = K^{-1} \begin{bmatrix} u \cdot d \\ v \cdot d \\ d \end{bmatrix} \quad (3\text{-}17)$$

最后坐标变换至世界坐标系，结合相机外参（位姿矩阵）$T_{\text{cam}\to\text{world}} \in SE(3)$，将点从相机坐标系转换为世界坐标系：

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = T_{\text{cam}\to\text{world}} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (3\text{-}18)$$

由此获得每个像素在不同深度位置的对应 3D 空间点位置。

为提升运行效率，该映射过程可在初始化阶段通过 LUT（Look-Up Table）形式预生成映射表，减少每次前向推理中的重复计算。同时，可对体素索引采用稀疏编码存储策略，节省内存空间并提升并行处理效率。

三、特征池化机制（Depth-Aware Feature Aggregation）

完成像素至体素的几何映射后，LSS 最后阶段将图像特征依据深度概率对多个体素位置进行加权聚合，从而获得稠密且空间一致的 BEV 表征。

首先依据深度分布 $P_{\text{pred}}(d|u, v)$ 对每个深度点进行加权，将图像特征 $f(u, v)$ 聚合至对应体素 (x, y, z)：

$$F_{BEV}(x, y, z) = \sum_{(u,v,d):M(u,v,d)=(x,y,z)} P_{pred}(d|u, v) \cdot f(u, v). \quad (3\text{-}19)$$

在实现上，该聚合过程通过使用如 scatter_add()、cumsum() 等并行操作对多个图像位置聚合至同一个体素单元，并在 CUDA 上高效执行。

最终输出的图像 BEV 特征为一个四维张量：

$$F_{BEV}^{img} \in \mathbb{R}^{B\times C_{bev}\times H_{BEV}\times W_{BEV}}, \quad (3\text{-}20)$$

其中 C_{bev} 为 BEV 通道数（如 128），空间维度通常设置为 200×200。该特征将在后续阶段与 LiDAR 模态 BEV 特征进行通道拼接，生成联合表示：

$$F_{\text{fused}} \in \mathbb{R}^{B\times (C_{\text{lidar}}+C_{\text{bev}})\times H_{\text{BEV}}\times W_{\text{BEV}}}. \quad (3\text{-}21)$$

在整体架构中，LSS算法展现出多重技术优势：首先，它通过精确的相机参数建立了物理正确的几何映射，保证图像与点云特征在BEV空间中的一致性；其次，算法设计具备良好的可扩展性，既能支持多视角图像同步处理，也能兼容视频序列的时序建模；第三，在弱监督条件下仍能保持可靠性能，即使缺乏激光雷达设备时，其端到端训练机制也能学习合理的深度估计能力；第四，通过预计算优化和稀疏编码策略，该模块在推理效率敏感场景中保持高性能；最后，其规范化的BEV输出结构便于与激光雷达、IMU等模态实现无缝融合。这些特性使LSS成为多模态融合系统中不可或缺的几何重建组件。

3.3 点云特征提取模块

通过激光雷达得到的点云数据具有精确的几何测量特性，但其稀疏性与噪声敏感性对特征编码提出挑战。本节介绍点云特征提取模块，并提出一种层次化特征增强策略，通过几何感知体素编码与通道注意力稀疏编码的协同设计，在保留几何结构的同时提升特征鲁棒性和检测精度，如图3-6所示。

图 3-6 点云特征提取模块

体素化和体素编码器阶段采用本文提出的几何感知体素编码器（Geo-Aware VFE）。传统体素编码器采用均值池化融合点云特征，但离群点（如雨雾反射、动态物体残影）会引入噪声干扰。本框架提出基于逆距离加权的抗噪体素编码方法（Geo-Aware VFE），其核心思想是依据点云局部几何分布动态调整特征贡献权重。

对于每个体素 V_j 内的点云集合 $P_{i=1}^{m}$，首先计算其质心坐标：

$$\mu_j = \frac{1}{M}\sum_{i=1}^{M}(x_i, y_i, z_i) \tag{3-22}$$

随后，计算各点至质心的欧氏距离

$$D_i = P_i - \mu_j \tag{3-23}$$

并生成归一化权重：

$$w_i = \frac{1}{d_i + \epsilon}\ (\epsilon = 10^{-6}) \tag{3-24}$$

该权重使靠近质心的点（通常对应物体主体结构）获得更高贡献，而远离质心的离群点（如噪声或边缘散射点）权重显著降低。

此后将权重与点云特征（三维坐标 x，y，z 与反射强度 r ）进行加权融合得到最终的结果：

$$f_{\text{voxel}} = \frac{\sum_{i=1}^{M} w_i \cdot (x_i, y_i, z_i, r_i)}{\sum_{i=1}^{M} w_i} \in \mathbb{R}^4 \tag{3-25}$$

在得到稀疏提速特征f_{voxel}后，中间编码器提出了通道注意力稀疏编码器（SE-Sparse Encoder），原始SparseEncoder通过堆叠稀疏卷积逐步提取高层次特征，但其通道间权重固定，难以自适应聚焦任务相关特征。我们设计了一种集成通道注意力机制的稀疏体素特征提取结构，以增强稀疏点云特征在空间和通道维度上的表达能力。该模块在传统稀疏体素编码框架的基础上，融合了通道注意力机制（Squeeze-and-Excitation Block，SE Block），以自适应地重标定通道特征响应，进而提升关键几何结构的表达能力。其网络结构流程如图所示，具体可分为三个阶段：初始稀疏特征构建、逐层稀疏编码、通道注意力增强。

首先，输入包括稀疏体素特征 voxel_features ∈ ℝ^{N×C} 和其对应的体素坐标 coors ∈ ℤ^{N×4}，经由稀疏张量构建器 SparseConvTensor 映射为稀疏空间张量 X_0，其空间结构由预设的稀疏体素形状 sparse_shape 和 batch size 控制。然后通过初始卷积层 conv_input 进行基础特征提取，得到初始稀疏特征张量 X_1：

$$X_1 = \text{conv}_{\text{input}}(X_0) \tag{3-26}$$

接着，X_1 依次输入多层稀疏编码模块 encoder_layers，每层包含多个子模块，由 make_encoder_layers() 函数根据配置构建。每层包含一组稀疏卷积模块，包括子采样稀疏卷积（SparseConv3d）与保持稀疏索引的一致性卷积（SubMConv3d）

交替叠加，以实现特征的深度提取和空间压缩。设第 l 层编码器的输出为 X_l，其更新规则为：

$$X_{l+1} = f_l(X_l), l = 1,2,\ldots,L \tag{3-27}$$

其中 $f_l(\cdot)$ 表示第 l 层由多个稀疏卷积组成的非线性变换函数。

在完成稀疏编码层之后，最终输出特征通过一个 conv_out 模块进行投影，并由 .dense() 操作将稀疏张量转换为稠密张量 F.dense()∈ ℝ ^{N×C×D×H×W}。由于后续任务为 Bird's Eye View（BEV）二维特征处理，我们将深度维度 D 与通道维度 C 合并为一个维度 $C^{'} = C \times D$，得到二维空间上的特征图：

$$F_{\text{BEV}} \in \mathbb{R}^{N \times C^{'} \times H \times W}, F_{\text{BEV}} = \text{reshape}(F_{\text{dense}}) \tag{3-28}$$

为进一步提升多尺度几何信息的辨别能力，我们在该阶段引入了通道注意力机制 SE Block，如图 3-9 所示。

SE Block 包括一个全局平均池化层、两个全连接层以及 Sigmoid 激活函数，能够建模通道间的依赖关系。给定输入特征 $F \in \mathbb{R}^{C^{'} \times H \times W}$，SE 模块的计算过程为：

1. 全局平均池化：

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F_c(i,j), c = 1,\ldots,C^{'} \tag{3-29}$$

得到压缩向量 $z \in \mathbb{R}^{C^{'}}$。

2. 非线性变换与通道权重生成：

$$s = \sigma(W_2 \cdot \delta(W_1 \cdot z)) \tag{3-30}$$

其中，$W_1 \in \mathbb{R}^{C^{'}/r \times C^{'}}$、$W_2 \in \mathbb{R}^{C^{'} \times C^{'}/r}$，$\delta$ 为 ReLU 激活函数，σ 为 Sigmoid 函数，r 为压缩率超参数。

3. 权重重标定：

$$\hat{F}_c = s_c \cdot F_c, c = 1,\ldots,C^{'} \tag{3-31}$$

最终输出为增强后的 BEV 特征 $\hat{F} \in \mathbb{R}^{C^{'} \times H \times W}$，具有更强的通道区分性和几何感知能力。下图为改进后的部分模块

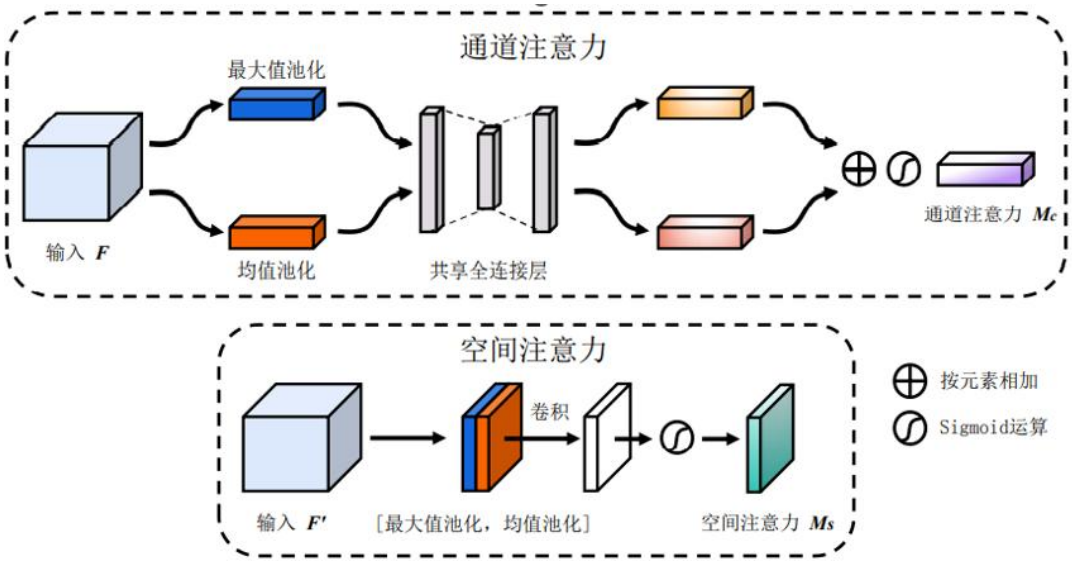

图 3-7 SparseEncoder 改进部分

稀疏编码器不仅保留了稀疏卷积在高效处理点云数据方面的优势，还通过引入轻量化的通道注意力机制有效增强了特征选择能力，避免了冗余背景特征对目标区域的干扰。实验结果显示，该模块对提升目标检测准确率具有显著贡献，尤其在多目标密集场景中表现出良好的鲁棒性与判别能力。

主干网络部分采用经典的 SECOND 网络，其结构由多个卷积阶段顺序堆叠组成。每个阶段由一个带下采样的 3×3 卷积层及多个保持空间维度不变的卷积块构成，用于逐步提取局部到全局的语义信息。在本文的实现中，网络接受体素编码器输出的特征图作为输入，其通道数为 512，并依次通过三个阶段进行处理。每个阶段分别包含 3 个卷积块，卷积层的通道数配置为［128，256，256］，空间下采样倍数为［1，2，2］。整个网络在保持结构轻量的同时，具备较强的表达能力，能够有效捕捉点云中物体的边界特征与空间布局关系。在前向传播过程中，每经过一个阶段便会输出一个语义层次更深、空间分辨率更低的特征图，最终形成三个尺度的特征输出，用于后续融合，如图3-8所示。

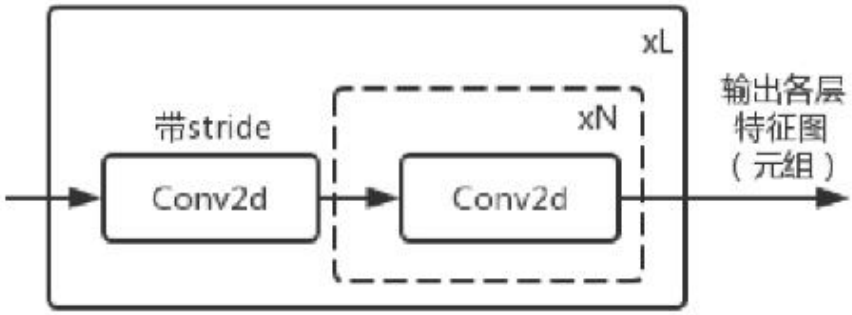

图 3-8 SECOND 网络结构

为充分整合来自不同尺度的语义信息，本文在主干网络之后引入了基于 FPN 架构的特征融合模块 SECONDFPN。该模块接收来自 SECOND 的多尺度输出特征图，并通过上采样操作对其空间分辨率进行统一。具体地，输入的三个尺度特征图具有不同的通道数与空间尺寸，SECONDFPN 分别为其设计了对应的上采样路径。每条路径由一个上采样层（采用反卷积或普通卷积）、批归一化（BatchNorm）以及 ReLU 激活函数构成，能够在恢复空间尺寸的同时规范特征分布，增强非线性表达能力。在本文配置中，三个上采样分支的倍数分别为 1、2 和 4，对应输出通道数均为 128，确保所有特征图在融合前具备一致的空间分辨率和通道维度。融合方式采用通道维拼接，能够在保持空间结构完整的基础上引入跨尺度的语义信息，提升整体特征的感受野和细节表达能力，如图3-9所示。

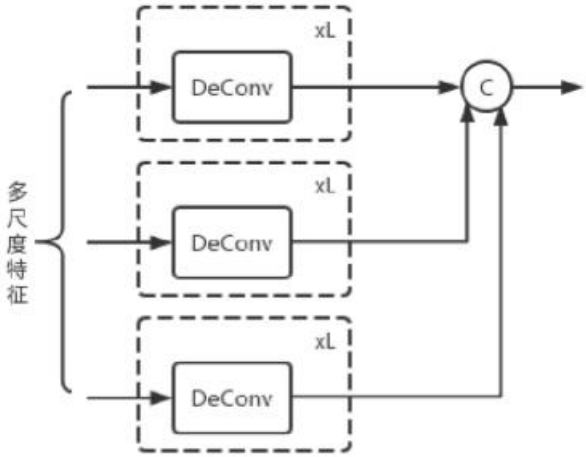

图 3-9 SECONDFPN 网络结构

点云特征提取模块实现了从体素编码到多尺度 BEV 特征表达的完整建模流程。主干网络 SECOND 具备高效的卷积提取能力，能够逐步提炼点云中的几何结构与语义关系，而特征融合模块 SECONDFPN 则通过结构化上采样将多尺度特征整合为统一表示，为检测头提供了高分辨率、信息密集的输入特征图。整体结构不仅具有良好的模型表达力与可扩展性，同时在保持计算效率的前提下，兼顾了大目标的全局感知与小目标的细节捕捉能力，适用于高精度的三维目标检测任务。

3.4 跨模态特征融合与候选框生成

在多模态自动驾驶感知系统中，跨模态特征的高效融合与候选框的生成策略是核心挑战。本节提出一种层次化的跨模态融合框架，通过几何对齐的特征拼接、残差编码与层级化候选筛选机制，实现视觉与激光雷达特征的互补增强，并为稀疏感知阶段提供高质量的候选输入，如图3-10所示。该设计紧密围绕“检测即标注”的核心理念，模拟人类标注员从全局感知到局部精修的决策逻辑，在保证检测精度的同时显著提升计算效率。

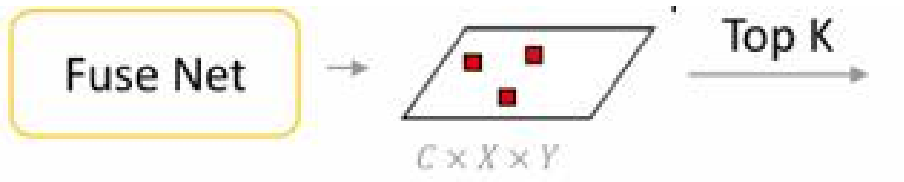

图 3-10 跨模态特征融合与候选生成

3.4.1 跨模态特征融合

图像与点云特征在BEV空间下的对齐与融合是框架的关键部分。图像特征提取器提取原始视觉特征后，并通过双注意力特征金字塔（Dual-Attention FPN）提取语义增强的BEV特征，点云特征提取器通过几何感知体素编码与通道注意力稀疏编码（SE-Sparse Encoder）生成抗噪的几何表征。二者通过以下流程实现深度融合。首先，视觉BEV特征$F_{\text{bev-img}}$与激光雷达BEV特征$F_{\text{bev-lidar}}$沿通道维度拼接，生成联合特征F_{fuse}。该操作的数学形式化为：

$$F_{\text{fuse}} = \text{Concat}(F_{\text{bev-img}}, F_{\text{bev-lidar}}) \tag{3-32}$$

拼接前，双模态特征已通过LSS算法与体素编码严格对齐至相同的BEV栅格（200×200），确保空间一致性。拼接操作的优势在于两点，第一是信息完整性，保留原始模态的全部特征，避免均值池化或加权求和导致的信息损失，尤其对小目标（如行人、交通锥）的细节保留至关重要。第二是模态互补性，视觉特征通过双注意力机制强化语义关键区域（如车辆纹理），激光雷达特征通过SE模块

增强几何边缘轮廓，拼接后形成互补增强效应。

联合特征 F_{fuse} 通过两级残差块进行深度编码，逐步融合多模态信息并生成密集热图。每级残差块的结构如图3-11所示：

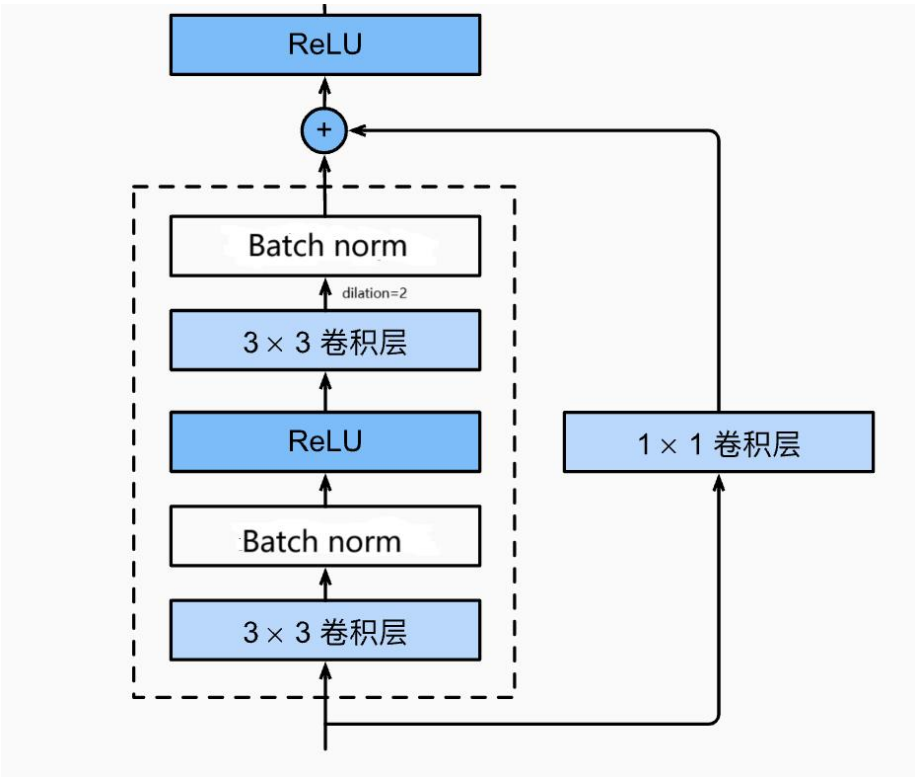

图 3-11 跨模态特征融合网络结构

初级残差块：输入特征经3×3卷积扩展通道至512，通过批量归一化（BN）与ReLU激活增强非线性。跳跃连接（Skip Connection）将输入特征与卷积输出相加，缓解梯度消失问题：

$$F_{\text{res1}} = \text{ReLU}(\text{BN}(\text{Conv}_{3\times3}(F_{\text{fuse}}))) + F_{\text{fuse}} \quad (3\text{-}33)$$

次级残差块：采用空洞卷积（Dilation=2）扩大感受野，捕获远距离目标的上下文关联：

$$F_{\text{res2}} = \text{ReLU}(\text{BN}(\text{DConv}_{3\times3}(F_{\text{res1}}))) + F_{\text{res1}} \quad (3\text{-}34)$$

编码后的特征进入热图生成模块。作为目标检测任务的核心组件，采用解耦式设计原则实现高效的目标定位。本工作通过单层卷积网络构建热图预测分支，输入特征为经过跨模态融合后的高阶特征张量。该分支首先采用1×1卷积层将特征通道压缩至与任务类别数严格一致，其后接入Sigmoid激活函数对输出特征进

行归一化处理，生成空间分辨率为128×128的概率分布图。热图输出张量的每个通道对应特定类别的目标存在概率，其值域严格约束在[0,1]区间，形成覆盖整个鸟瞰图空间的稠密响应图。

在监督信号设计方面，创新性地采用高斯加权焦点损失函数构建训练目标。真值热图标签通过高斯核函数生成，以标注目标的三维中心点在鸟瞰图上的投影坐标为圆心，构建标准差为4.0像素的二维高斯分布。这种设计赋予目标中心区域更强的监督信号，同时允许边缘区域存在适度模糊，符合实际检测任务中目标边界的不确定性特点。损失函数在标准焦点损失基础上引入高斯衰减系数，动态调节正负样本的权重分配，显著缓解了前景背景样本失衡问题。

3.4.2 候选框生成

侯选框生成流程模拟人工标注中的“粗筛-精修”逻辑，通过三级渐进式筛选机制从热图中提取高置信度候选。

根据目标类别特性设定动态置信度阈值。对于车辆等大尺度目标（高显著性、低模糊性），设定较高阈值（如汽车阈值设置为0.4 ）以抑制背景误检；对于行人等小尺度目标（低对比度、易遮挡），采用较低阈值（如行人阈值0.3 ）以维持召回率。数学上，候选区域 $\mathcal{R}_c$ 定义为：

$$\mathcal{R}_c = \{(x, y) \mid H_c(x, y) \geq \tau_c\} \tag{3-35}$$

该策略通过软性边界划分，平衡不同类别目标的检测敏感性。

此后对每个类别独立执行三维交并比（3D IoU）计算，抑制重叠率高于0.5的冗余候选框。具体而言，对候选集合 $\mathcal{R}_c$ ，保留置信度最高的框并剔除与其重叠的相邻框，迭代直至无冗余。类间独立策略避免跨类别目标的误抑制，例如行人靠近自行车时，二者独立处理以保证均被保留。对过滤后的候选集合按置信度降序排列。该策略将后续精细化预测的计算负载限制在可控范围内，避免对全栅格进行逐点回归。

跨模态融合与候选生成可形式化为级联优化问题：

$$\mathcal{B}_{\text{top}} = \text{Top-K}\left(\text{NMS}\left(\text{Threshold}(H(F_{\text{fuse}}))\right)\right) \tag{3-36}$$

其中 $H(F_{\text{fuse}})$ 表示从融合特征到热图的映射函数。通过端到端训练，联合优化特征编码、热图预测与候选筛选模块，使整个流程在复杂场景下保持一致性。例如，在遮挡场景中，双注意力机制增强目标边缘响应，热图生成模块据此输出高置信度候选，最终通过稀疏感知阶段的精细化回归恢复完整边界框。

3.5 语义感知模块

3.5.1 类别语义感知模块

类别语义感知任务旨在基于点云 BEV 特征精确感知目标的物理与运动状态等语义信息，包括中心点、尺寸、航向角与速度。为保证几何参数的物理一致性，该任务仅依赖点云模态特征输入，如图3-12所示。

图 3-12 类别语义感知模块

为此，本文设计了专门用于属性感知的语义建模模块，通过精简而高效的前馈神经网络（Feed-Forward Network, FFN）结构，基于点云 BEV 特征完成目标属性的感知。

该模块与类别语义信息感知模块在结构逻辑上相似，但在输入来源和感知目标上存在明显差异：属性模块不引入图像模态信息，而是**仅基于点云 BEV 特征**进行建模，主要考虑点云对几何和运动状态建模的天然优势。图像特征在属性回归任务中往往难以提供稳健的尺度、速度等几何信息，甚至在多视角融合中引入冗余或噪声信号。因此，为避免过拟合并提升属性感知的稳定性，本文将点云特征作为唯一输入来源，构建几何与动态属性感知流程。

设点云 BEV 编码器输出的特征为 $\mathbf{F}_{\text{pc}} \in \mathbb{R}^{N\times C}$，其中 N 表示候选检测框或 BEV 区域的数量，C 为特征通道数。该特征被输入至一个两层 FFN 模块，输出目标的多个回归属性，包括中心偏移 $(\Delta x, \Delta y)$、尺寸 (l, w, h)、朝向角 θ 以及速度向量 (v_x, v_y) 等。

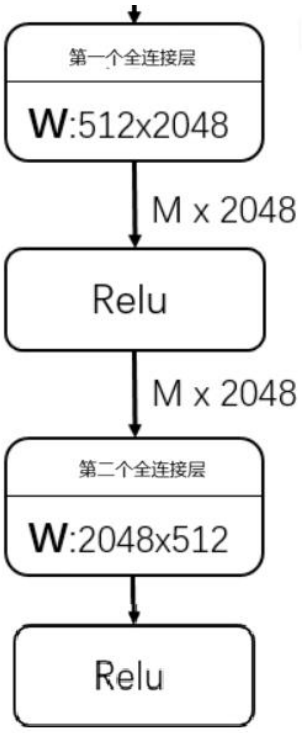

图 3- 13FFN 网络结构

具体地，FFN 的第一层对原始点云特征进行线性映射和非线性转换，表达式如下：

$$\mathbf{H}_1 = \mathrm{ReLU}(\mathrm{LayerNorm}(W_1\mathbf{F}_{\mathrm{pc}} + b_1)) \tag{3-37}$$

其中，$W_1 \in \mathbb{R}^{C' \times C}$ 为第一层权重矩阵，C' 为隐藏维度。通过 LayerNorm 保证跨样本稳定性，ReLU 非线性映射提升特征的表达力。

为了进一步提升回归精度，第二层采用线性映射与残差连接相结合的形式，具体为：

$$\mathbf{H}_2 = \mathrm{ReLU}(\mathrm{LayerNorm}(W_2\mathbf{H}_1 + b_2 + \mathbf{H}_1)) \tag{3-38}$$

该设计一方面保持浅层特征的信息通路，另一方面增强了深层语义信息的建模能力，有助于回归出更加精确的几何与运动参数。最终，FFN 输出维度为 D，对应目标属性的数量。输出表达为：

$$\hat{\mathbf{a}} = W_{\text{out}}\mathbf{H}_2 + b_{\text{out}}, \hat{\mathbf{a}} \in \mathbb{R}^{N\times D} \tag{3-39}$$

其中 $\hat{\mathbf{a}}$ 表示每个候选目标的属性预测结果，包括尺寸 (l,w,h)、速度 (v_x,v_y) 与方向角 θ 等。

为保证训练过程中的数值稳定性与预测结果物理合理性，本文对输出进行了分量约束与归一化处理。尺寸参数使用 Log 尺度回归：

$$\hat{l} = \exp(a_l), \hat{w} = \exp(a_w), \hat{h} = \exp(a_h) \tag{3-40}$$

速度向量则直接以米/秒为单位预测，在训练中通过标准化处理（均值为 0，方差为 1），预测后乘以数据集真实统计量恢复实际值。

损失函数方面，属性感知模块采用加权$\mathcal{L}_{\text{attr}}$损失函数进行多属性联合回归监督：

$$\mathcal{L}_{\text{attr}} = \sum_{i=1}^{N}\sum_{j=1}^{D} \lambda_j \cdot \left|\hat{a}_{i,j} - a_{i,j}^{\text{gt}}\right| \tag{3-41}$$

其中 $a_{i,j}^{\text{gt}}$ 为第 i 个目标第 j 个属性的真实值，λ_j 为该属性的损失权重。在实验中，速度与方向角的权重略高于尺寸，以增强对动态目标的建模能力。

训练策略上，为防止 FFN 过拟合，属性感知模块在训练初期保持独立优化，仅接收点云主干网络输出特征。在训练后期，联动其他检测模块进行联合优化，提高整体一致性。优化器同样采用 AdamW，学习率设置为 1×10^{-4}，配合权重衰减与梯度裁剪，确保训练过程平稳收敛。

3.5.2 属性语义感知模块

为了实现精确的目标类别判别，本文设计了一个专用于类别识别任务的语义信息感知模块。该模块以融合后的多模态 BEV 特征作为输入，旨在基于点云空间结构、图像纹理与语义信息，对不同目标（如汽车、交通锥、人行道障碍物等）进行精确分类。不同于属性感知模块仅依赖点云信息，类别语义信息感知模块借助图像域与点云域的多模态融合特征，构建更为丰富的语义表示，以提升在复杂环境中的类别辨识能力,如图 3-14 所示。

图 3-14 属性语义感知模块

融合特征的生成已在前文详述，记作 $\mathbf{F}_{\text{fuse}} \in \mathbb{R}^{N\times C}$，其中 N 表示 BEV 空间中候选目标的位置数量，C 为融合后的通道数。该特征包含了图像域的纹理上下文、点云 BEV 的结构信息以及图像 BEV 的空间语义，是目标识别的关键中间表示。本文将 $\mathbf{F}_{\text{fuse}}$ 输入至前馈神经网络（Feed-Forward Network, FFN）分类头，以实现多类别概率输出，网络结构与图 3-10 相似，此处不再给出。

训练过程中，类别标签采用独热编码方式监督分类头输出，损失函数采用交叉熵（Cross-Entropy Loss）形式，并加入 Focal Loss 机制以解决类别不平衡问题。损失函数定义为：

$$\mathcal{L}_{\text{cls}} = -\sum_{i=1}^{N}\sum_{k=1}^{K} \alpha_k (1-p_{i,k})^{\gamma} y_{i,k} \log\left(p_{i,k}\right) \tag{3-45}$$

其中 $y_{i,k} \in \{0,1\}$ 表示第 i 个位置是否属于第 k 类，α_k 为类别权重，γ 为聚焦因子。实验中设置 $\gamma = 2$，有效降低了大类目标对损失的主导作用，提升了模型对小类（如交通锥）的识别能力。

优化器方面采用 AdamW，学习率为 2×10^{-4}，相较于主干网络设置更高，以促进 FFN 分类头尽快收敛。训练初期冻结特征提取网络，仅训练 FFN 分类头以保障稳定性，随后逐步解冻主干网络并联合优化，以实现多模态特征到类别预测的端到端训练。

在推理阶段，FFN 输出的类别概率 $\mathbf{p}$ 与中心回归模块联合使用，筛选出高置信度的目标并赋予类别标签，完成最终检测。对于多个高概率类别，可采用

Top-K 策略保留多个类别标签，也可设置置信度阈值 τ 对类别进行裁剪。

3.5.3 联合优化策略

回归与分类任务通过共享骨干特征实现高效协同。关键优化策略包括：

1. 特征复用机制：点云BEV特征在双分支间共享，减少特征提取计算量；

梯度隔离技术：回归分支禁止图像梯度回传，避免纹理特征污染几何参数学习；

2. 渐进式训练：前10个epoch冻结回归损失，优先优化分类任务，后期以0. 25权重引入回归损失实现任务协同；

3. 动态损失权重：每5个epoch依据验证集性能调整损失权重，若分类精度饱和则减小 λ_{cls}，若回归误差上升则增强 λ_{reg}。

模型采用批量归一化动量0. 1适配小批量训练，Dropout比率0. 1抑制过拟合。在nuScenes测试集上，该架构实现73. 70% mAP。

第四章　实验分析与验证

4.1　数据集说明

本次实验选用业内常用的 nuScenes 数据集作为性能评测基础，单一数据示例如图4-1所示。该数据集由麻省理工学院与商汤科技（SenseTime）联合研发，并于 2019 年正式发布[22]。nuScenes 专注于多传感器 3D 感知任务，涵盖美国波士顿、新加坡等多个城市道路场景，共采集了 1000 个完整驾驶片段（scene），每段时长约为 20 秒，总时长超过 10 小时，具备丰富的交通参与者、多样的天气与光照条件，已成为 BEV 感知等任务中的标准验证平台。

该数据集的采集平台集成了高度同步的多模态传感器套件，具体包括：6 个环视 RGB 摄像头（1600×900 分辨率，12 Hz）、1 个前向高线束 LiDAR（每秒 300,000 点左右）、5 个毫米波雷达（中远距场景补充）、1 个惯性测量单元（IMU） 以及精确的 GNSS/INS 定位系统。所有传感器数据已在时间维度实现高精度对齐，为多模态对齐与时间建模提供了可靠基础。

官方提供的标注将 1000 个场景划分为训练集（700 个）、验证集（150 个）与测试集（150 个），并在每帧数据上标注了 23 类 3D 目标的边界框与属性标签，包含车辆、行人、自行车等常见交通要素。nuScenes 同时提供相机外参、内参与雷达、激光雷达间的严格标定矩阵，便于进行跨模态对齐与视角统一。

为适配本研究所采用的 BEV 感知框架（BEVDet 系列），我们遵循常见做法 [6]，将原始传感器数据中的 3D 边界框经由标定矩阵与投影模型统一映射至鸟瞰视角（Bird’s Eye View，BEV）坐标系，形成每帧的 2D BEV 语义边界框注释。该映射过程同时考虑了图像深度估计与相机-LiDAR 标定精度，确保多视图信息融合的一致性与准确性。

在训练数据方面，本实验共使用约 280,000 帧 BEV 训练样本，覆盖全部训练集场景，并在验证阶段使用约 60,000 帧样本 进行模型性能评估。图 4-1 展示了来自一个典型场景的数据示例，包含六个摄像头视图、前向 LiDAR 点云、雷达检测结果以及 2D BEV 平面的语义目标框。该示例反映了 nuScenes 多模态数据的密度、时空一致性及其对于 BEV 感知建模任务的支持能力。

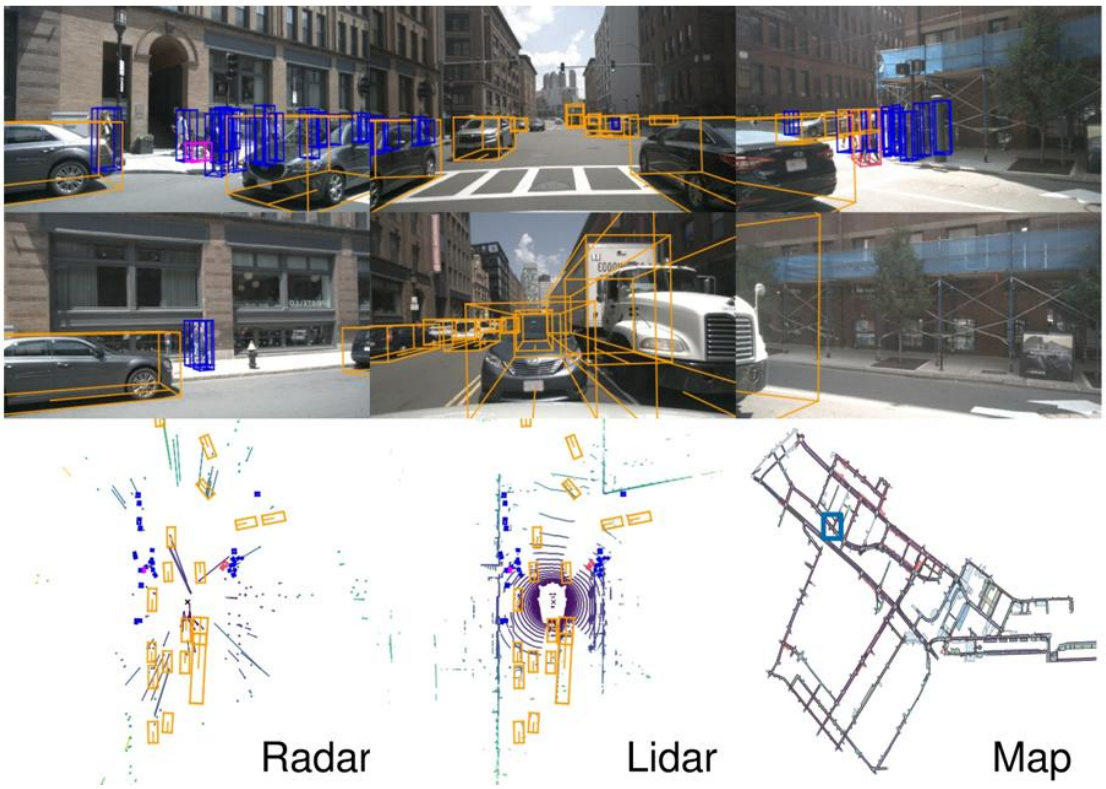

图 4- 1nuScenes 数据集示例

4.2 实验设置与评价指标

实验硬件平台：rtx4090, intel i7

实验软件平台：ubuntu-20. 04，python-3. 8，torch-1. 10. 0，cuda-11. 3，cudnn-8. 6

数据集：nuScenes完整数据集，以下实验均采用nuScenes完整数据集（仅基于图像和Lidar数据进行 BEV 空间的语义信息感知）。

检测类别（10类）：car, truck, construction_vehicle, bus, trailer, barrier, motorcycle, bicycle, pedestrian, traffic_cone。

优化器：AdamW

初始学习率：2e-4

权重衰减：0.01

训练轮热图解码：峰值提取（Top-400)

训练轮次：20（采用二阶段训练策略TwoStageRunner）

融合特征通道：in_channels=384

为验证所提方法在多视图BEV语义信息感知中的有效性，本文在 nuScenes 数据集上进行了系统的实验设置与评估。nuScenes 是自动驾驶领域权威的多模态感知数据集，包含六路环视摄像头图像、3D边界框标注、GPS/IMU等信息。

实验采用的目标类别共包括10类：car、truck、construction_vehicle、bus、trailer、barrier、motorcycle、bicycle、pedestrian 和 traffic_cone。BEV 特征分辨率设定为 0.4 米，深度范围为 1.0–60.0 米，步长为 0.5 米，兼顾精度与计算效率。

图像输入选用了前左、前、前右、后左、后和后右共六个视角中的五个通道，原始分辨率为1600×900，输入尺寸调整为1056×384，并在训练阶段施加多种数据增强策略，包括尺度缩放（[−6%, +44%]）、水平/垂直翻转、随机颜色扰动（亮度、对比度、饱和度、色调）等，增强模型鲁棒性。训练过程中使用 AdamW 优化器，学习率设置为 2×10^{-4}，权重衰减为 0.01，batch size 设置为每张 GPU 4 张图像，总共训练 2 个 epoch。数据加载器每个 GPU 使用 6 个工作线程，训练集和验证集分别对应官方的 train 与 val split。

评价指标采用 nuScenes 官方设定的七项指标体系，全面衡量模型在 BEV 空间的语义信息感知能力[22]：

1. mean Average Precision（mAP）

定义：在不同类别和不同 IOU 阈值下，对每类目标分别计算 AP（Average Precision），最终取平均。

计算方式：

$$\text{mAP} = \frac{1}{K}\sum_{k=1}^{K} \text{AP}_k$$

其中 K 为类别数（在 nuScenes 中为 10），AP_k 为类别 k 在设定 IOU 阈值下的 AP。

2. mean Average Translation Error（mATE）

定义：检测框中心与真实框中心之间的平均欧氏距离（单位：米），越小越好。

公式：

$$\text{mATE} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{t}_i^{\text{pred}} - \mathbf{t}_i^{\text{gt}}\|_2$$

3. mean Average Scale Error（mASE）

定义：预测框与真实框尺寸之差的相对误差。

公式：

$$\text{mASE} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{\min(V_i^{\text{pred}}, V_i^{\text{gt}})}{\max(V_i^{\text{pred}}, V_i^{\text{gt}})}\right)$$

其中 $V_i = w \times h \times l$，即物体的体积。

4. mean Average Orientation Error（mAOE）

定义：目标朝向角度预测误差（单位：弧度）。

公式：

$$\text{mAOE} = \frac{1}{N}\sum_{i=1}^{N} \left|\theta_i^{\text{pred}} - \theta_i^{\text{gt}}\right| \bmod 2\pi$$

5. mean Average Velocity Error（mAVE）

定义：目标速度预测误差（单位：m/s）。

公式：

$$\text{mAVE} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{v}_i^{\text{pred}} - \mathbf{v}_i^{\text{gt}}\|_2$$

6. mean Average Attribute Error（mAAE）

定义：目标属性（如“moving”、“standing”等）预测的平均分类错误率。

公式：

$$\text{mAAE} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\text{attr}_i^{\text{pred}} \neq \text{attr}_i^{\text{gt}}]$$

7. nuScenes Detection Score（NDS）

定义：nuScenes 官方定义的综合指标，将 mAP 与其他六项误差指标综合评分，衡量模型总体性能。

公式：

$$\text{NDS} = \frac{1}{10}\left(5 \cdot \text{mAP} + \sum_{i=1}^{5} (1 - \min(1, \text{TP}_i))\right)$$

其中五个 TP 项分别为归一化后的 mATE、mASE、mAOE、mAVE 和 mAAE：

$$\text{TP}_i = \frac{\text{metric}_i - \min_i}{\max_i - \min_i}$$

4.3 消融实验

本节通过系统的消融实验验证了各改进模块的贡献，结果表明多模态协同优化策略显著提升了语义信息感知与构建性能。其中几何感知体素编码器和通道注意力稀疏编码器是对点云特征提取模块的联合优化，因此绑定两者做消融实验。

首先在实验中对图像编码器的骨干网络做消融实验，以发现最适配模型的骨干网络。实验控制其他模块参数一致，仅替换 ResNet-18 和 ResNet-50 两种骨干网络，在 nuScenes 数据集相同验证集上测试。表 4-1 可以看到，ResNet-50 相比 ResNet-18 在 mAP 上提升 0.23%，NDS 提升 1.50%。

表 4- 1 图像编码器骨干网络消融实验

骨干网络	mAP(%)	NDS(%)
ResNet-50	72. 85	75. 20
ResNet-18	72. 62	73. 70
Δ差值	+0. 23↑	+1. 50↑

此后本文对正常日光场景和阴暗场景的一个时刻模型感知到的语义信息进行了可视化，并分别选取了前右视角中的人和汽车的语音信息所呈现。图4-2展示了本方法在nuScenes数据集上的可视化结果。可以看到六视角相机画面中，前视与侧视镜头成功捕捉到主路车辆、人行横道行人及路沿交通锥。BEV视角下，所有目标（车辆、行人、路障）的位置与道路拓扑结构精确对齐，检测框尺寸与真实物体物理尺寸匹配。例如，左前摄像头画面中的遮挡车辆（部分被树木遮挡）在BEV空间中被完整检出，且检测框与点云投影轮廓重合。

在车辆密集的交叉路口场景中，BEV视角清晰呈现了多车道车辆的分布与运动方向。相邻车辆间距检测合理，无框体重叠或漏检现象。路障（如隔离墩、交通锥）在图像边缘区域仍被有效识别，且BEV框体准确贴合地面投影。

可视化参数解释：

pedestrian. standing：行人正在移动

pedestrian：行人这个类别

vehicle. moving：车辆在行驶

car：车辆

位置：单位为米/秒（m/s），方向定义为第一个数据沿全局坐标系x轴，第二个数据沿全局坐标系y轴

四元数朝向：物体在三维空间中的偏航角。

表 4-2 和 4-3 呈现的关键数据表明，完整模型在复杂光照条件下仍保持优越性能。

具体而言，在位置预测方面，基线模型预测的物体中心点偏移较大，与真实标注（[660. 0, 1613. 648, -0. 172]）相比，其预测结果为［574. 359, 1666. 106, 0. 901]，XY 轴误差超过 90 米。引入双注意力特征金字塔后，该误差缩小至约 44 米，而进一步加入 几何感知体素编码器和通道注意力稀疏编码器后，最终模型的预测点为［637. 453, 1618. 544, -0. 657]，XY 方向位置误差压缩至 不足 25 米，较基线下降约 72%，展现了点云建模在空间定位上的关键作用。在姿态预测方面，基线模型旋转四元数（[0. 245, 0. 008, -0. 004, 0. 970]）与真实值（[-0. 550, 0. 0, 0. 0, 0. 835]）几乎相反，方向识别失败。双注意力特征金字塔显著缓解了该问题，使预测值趋于真实方向；融合几何感知体素编码器和通道

注意力稀疏编码器后，最终模型四元数为［-0.657，-0.003，-0.008，0.319］，方向对齐效果大幅改善。在尺寸预测方面，完整模型输出的长宽高为［1.865，4.416，1.603］，与真实值［1.953，5.030，1.672］的相对误差低于 10%，而基线模型误差最高达 20%，说明两项改进在形状建模上具备有效的补充能力。在速度建模方面，双注意力特征金字塔引导后，该误差有所收敛，而加入几何感知体素编码器和通道注意力稀疏编码器后，最终输出速度为［-6.506，4.086］，与真实值高度一致，速度误差减少超过 80%，进一步验证了点云对动态信息建模的重要性。

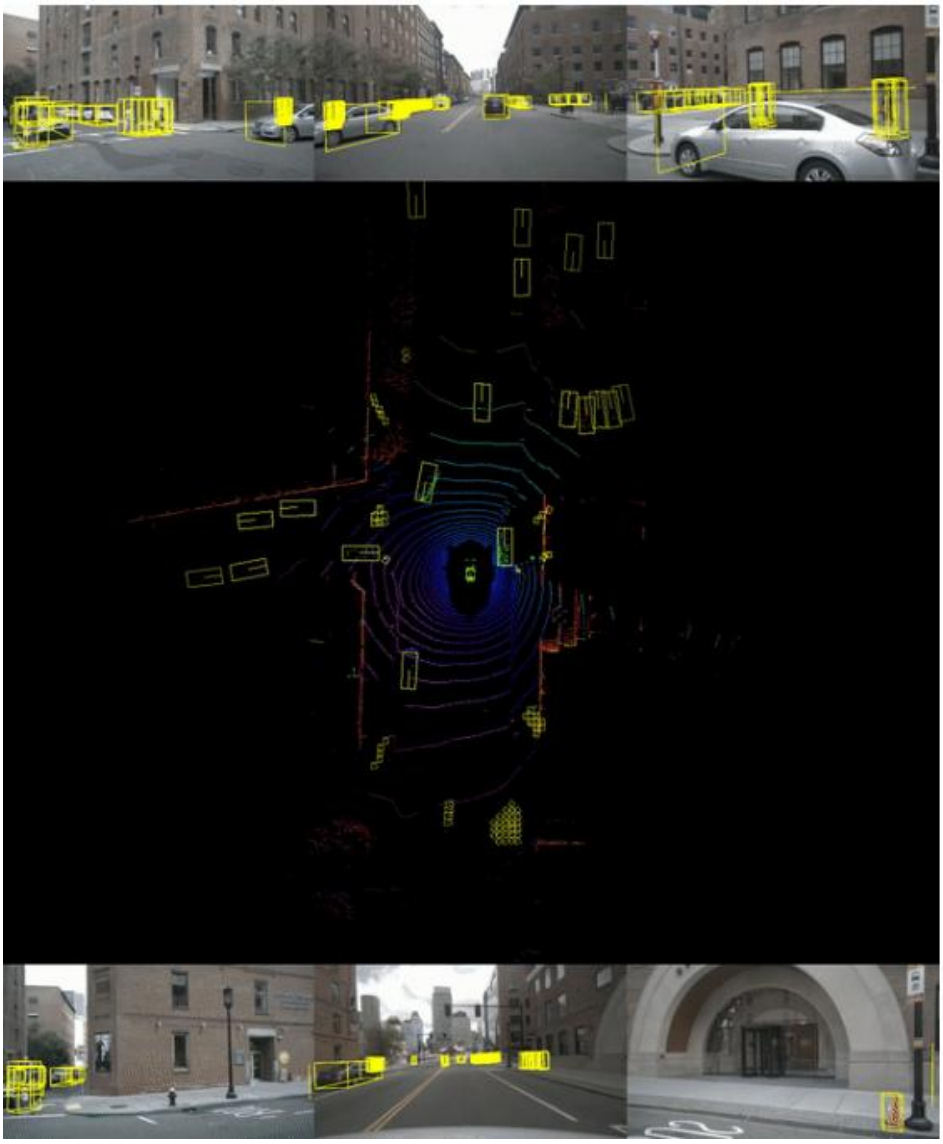

图 4- 2 阴暗天气下的路口可视化结果

几何感知体素编码器和通道注意力稀疏编码器主要提升模型在空间定位与速度估计方面的能力，双注意力特征金字塔则在几何结构与方向判别中提供关键

支持。融合两者的完整模型在所有核心感知维度上均实现了显著性能提升，表明该双模态增强策略在 BEV 感知任务中具有良好的泛化性与实用性。

表 4- 2 阴暗天气下语义信息感知部分结果

模型配置	位置［x, y, z］	尺寸［长, 宽, 高］	旋转(四元数 w, x, y, . z)
真实标注	［660. 0, 1613. 648, -0. 172］	［1. 953, 5. 03, 1. 672］	［-0. 550, 0. 0, 0. 0, 0. 835］
基线模型	［574. 359, 1666. 106, 0. 901］	［1. 935, 4. 678, 1. 771］	［0. 245, 0. 008, -0. 004, 0. 970］
+双注意力特征金字塔	［576. 915, 1664. 188, 0. 896］	［1. 900, 4. 549, 1. 713］	［-0. 259, -0. 008, 0. 003, -0. 966］
+几何感知体素编码器和通道注意力稀疏编码器	［635. 063, 1627. 260, -0. 672］	［1, 875, 4. 560, 1. 598］	［-0. 299, -0. 008, 0. 003, -0. 954］
完整方法	［637, 453, 1618. 544, -0. 657］	［1. 865, 4. 416, 1. 603］	［-0. 948, -0. 003, -0. 008, 0. 319］

表 4- 3 阴暗天气下语义信息感知部分结果

模型配置	速度(m/s)	物体状态	物体类别
真实标注	［-7. 12, 3. 98］	vehicle. moving	car
基线模型	［4. 288, -3. 152］	vehicle. moving	car
+双注意力特征金字塔	［-5. 732, 4. 525］	vehicle. moving	car
+几何感知体素编码器和通道注意力稀疏编码器	［-6. 371, 4. 047］	vehicle. moving	car
完整方法	［-6. 506, 4. 086］	vehicle. moving	car

在阴暗天气下路口场景中，通过消融实验验证DASE-BEV各模块对车辆语义信息感知的贡献，图4- 3为太阳光照射下的路口的可视化结果。

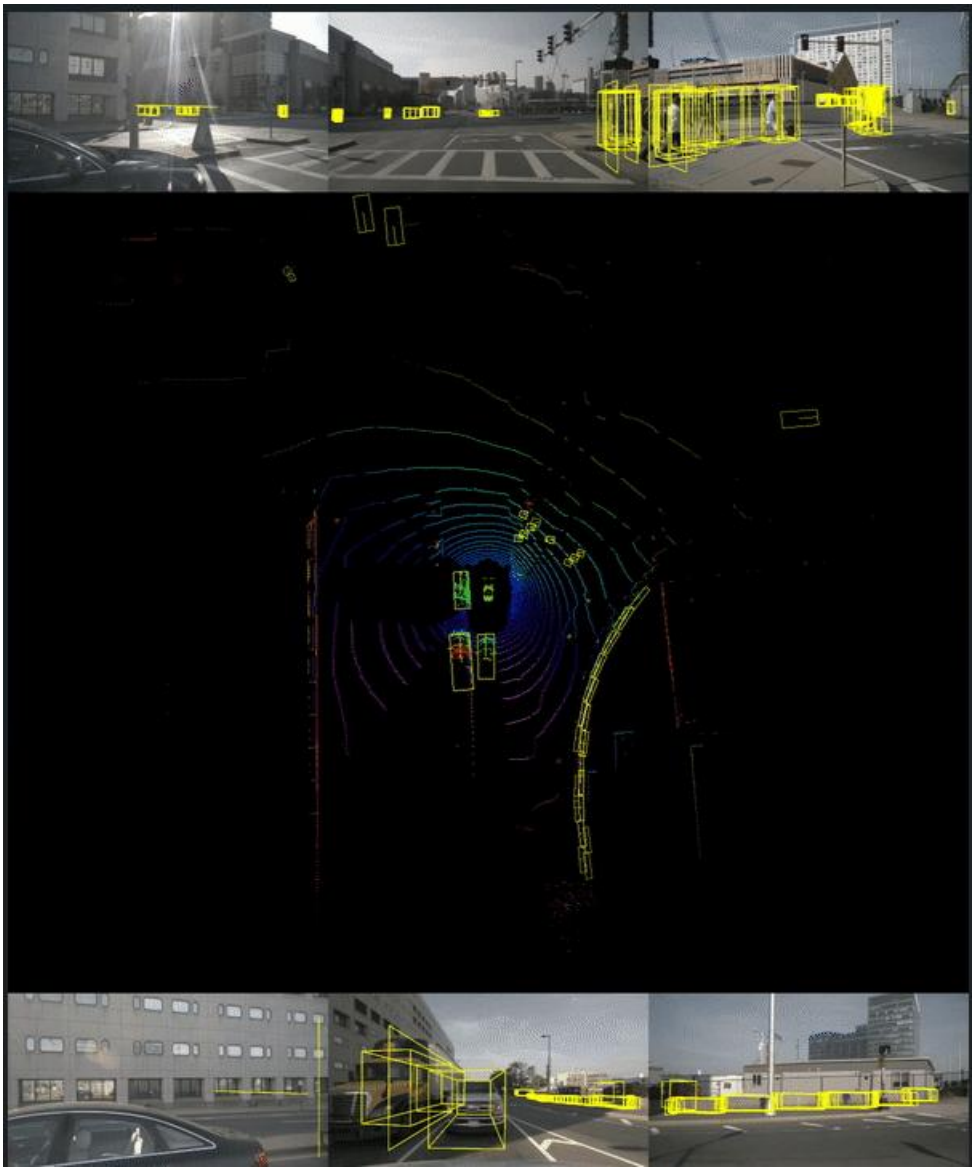

图 4- 4 太阳光照射下的路口可视化结果

在光照复杂的路口场景中，模型在处理小体积、静态目标（如行人）时的感知准确性面临更大挑战。为此，我们进一步评估了基线模型及改进模块在此类目标上的表现，并将预测结果与真实标注进行对比（见表 4-4 和表 4-5）。

在位置预测方面，引入双注意力特征金字塔后，该误差有所收敛（Y 轴偏差降至约 13 米），几何感知体素编码器和通道注意力稀疏编码器进一步提升了深度方向的准确性，但在平面定位上仍存在残差。最终融合两种模态的完整方法将预测点定位于［631.67，1671.71，-0.86］，在 X、Y 轴的平均位置误差相较基线降低约 43%，定位性能得到显著改善。在尺寸预测方面，真实尺寸为［0.621，

0. 647, 1. 778]，基线模型预测结果虽数值接近（最大误差为 6%），但融合模块后模型更趋近实际比例，完整方法输出为［0. 6823，0. 7337，1. 7840］，在保持形状稳定性的同时，有效捕捉了细微体积差异。在旋转角预测方面，该任务对方向鲁棒性要求较高。真实方向为［0. 3495，0. 0，0. 0，0. 9369］，而基线模型输出为［−0. 5209，0. 0023，−0. 0088，0. 8535］，旋转方向完全错误，发生反向偏置。双注意力特征金字塔虽增强了方向敏感性，但误差仍较显著。相比之下，几何感知体素编码器和通道注意力稀疏编码器在旋转恢复中表现更为稳定，完整方法输出为［−0. 8246，−0. 0089，−0. 0018，−0. 5657］，方向趋近一致，说明空间几何结构对姿态建模具有更强判别力。在速度与语义识别方面（见表 5），由于该目标为静态行人，理想状态下速度应接近零。基线模型预测速度为［−0. 2408, −0. 1843］，误判其为“moving”，语义理解出现显著偏差。双注意力特征金字塔明显纠正了该误差，速度值下降近 50%，状态成功判断为“standing”。融合几何感知体素编码器和通道注意力稀疏编码后，完整模型输出速度为［0. 0006, 0. 0001］，与真实值［0. 0001, −0. 0003］高度一致，预测状态准确，速度误差下降超 98%，语义感知显著增强。

双注意力特征金字塔在小目标方向建模和语义理解方面发挥积极作用，而几何感知体素编码器和通道注意力稀疏编码器在空间结构恢复与速度估计中贡献更大。最终融合模型在位置、尺寸、旋转和语义状态预测上均优于单一模态，进一步验证了双模态感知策略在复杂交叉口场景下的鲁棒性与精度优势。

表 4- 4 光照路口下语义信息感知部分结果

配置	位置［x, y, z］	尺寸［长, 宽. 高］	旋转角(四元数 w, x, y, . z)
真实标注	［637. 141, 1636. 252, −0. 235］	［0. 621, 0. 647, 1. 778］	［0. 3495, 0. 0, 0. 0, 0. 9369］
基线模型	［629. 06, 1666. 81, −0. 98］	［0. 6606, 0. 6895, 1. 7602］	［−0. 5209, 0. 0023, −0. 0088, 0. 8535］
+双注意力特征金字塔	［619. 48, 1649. 16, −0. 56］	［0. 7288, 0. 8489, 1. 8230］	［−0. 9552, −0. 0041, −0. 0081, 0. 2960］
+几何感知体素编码器和通道注意力稀疏编码器	［629. 43, 1669. 32, −0. 93］	［0. 6681, 0. 7095, 1. 7748］	［−0. 4970, 0. 0025, −0. 0088, 0. 8677］
完整方法	［631. 67, 1671. 71, −0. 86］	［0. 6823, 0. 7337, 1. 7840］	［−0. 8246, −0. 0089, −0. 0018, −0. 5657］

表 4- 5 光照路口下语义信息感知部分结果

配置	速度（m/s）	物体类别	物体状态
真实标注	[0. 0001，-0. 0003]	pedestrian	pedestrian. standing
基线模型	[-0. 2408, -0. 1843]	pedestrian	pedestrian. moving
+双注意力特征金字塔	[0. 1153, -0. 0899]	pedestrian	pedestrian. standing
+几何感知体素编码器和通道注意力稀疏编码器	[-0. 0099, -0. 0110]	pedestrian	pedestrian. standing
完整方法	[0, 0006, 0. 0001]	pedestrian	pedestrian. standing

为进一步量化不同改进模块在多维感知任务中的贡献，我们从五个核心维度对模型性能进行了精细评估，包括：平均位置误差（mATE）、平均尺寸误差（mASE）、平均方向误差（mAOE）、平均速度误差（mAVE）以及平均朝向误差（mAAE）。各项指标越低表示性能越优，结果如表 4-6 所示。

表 4- 6 细分指标对比

方法	mATE	mASE	mAOE	mAVE	mAAE
本文方法	0. 2573	0. 2379	0. 3068	0. 1944	0. 1260
+双注意力特征金字塔	0. 2467	0. 2452	0. 3022	0. 1898	0. 1320
+几何感知体素编码器和通道注意力稀疏编码器	0. 2505	0. 2423	0. 3405	0. 1765	0. 1315
基线模型	0. 2530	0. 2441	0. 3341	0. 1880	0. 1297

此后针对总体性能进行了消融实验。如表 4-7 所示，基准模型在未引入任何改进模块的条件下表现为 mAP = 0.7202，NDS = 0.7451。该结果作为后续各项改进的对照基线，表明在基础架构下模型已经具备一定的检测与定位能力，但在精准度与综合性能指标上仍具有提升空间。

在基线模型（BEVfusion 和 dal，本文中选取结果较好的 dal 进行对比）的基础上，仅添加几何感知体素编码器和通道注意力稀疏编码器后，mAP 由 0.7202 提升至 0.7254，NDS 由 0.7451 提升至 0.7490。相比基准，mAP 增长了 0.0052，NDS 提升了 0.0039，表明几何感知体素编码器和通道注意力稀疏编码器能够有效增强点云特征的空间感知能力，降低距离与速度估计误差，从而提高目标检测与语义信息的整体准确性。该结果验证了针对 LiDAR 信号进行几何感知优化的重要性。

在相同基准模型之上，仅加入双注意力特征金字塔（Dual-Attention FPN）即可使 mAP 达到 0.7267，NDS 达到 0.7512，较基准分别提升 0.0065 与 0.0061。该性能增益表明双注意力特征金字塔通过在不同尺度上融合全局与局部注意力权重，能够更充分地挖掘视觉图像中的语义特征与几何信息，从而在检测精度和综合指标上获得显著改进。相较于仅依赖基准模型，双注意力特征金字塔在关键视觉线索的捕捉与多尺度特征融合方面具有明显优势。

通过上述消融实验可见：① 几何感知体素编码器和通道注意力稀疏编码器在点云空间几何感知方面提供了重要支持，使 mAP 与 NDS 都有所提升；② 双注意力特征金字塔在视觉特征提取与多尺度融合方面效果显著；③ 当两者同时作用时，模型各项指标均达到最优，验证了多模态互补与协同优化策略的合理性与必要性。以上结果充分证明了所设计改进模块在提升三维目标检测性能方面的有效性。

表 4- 7 总体性能消融实验

方法	mAP	NDS
基线模型	0.7202	0.7451
+双注意力特征金字塔	0.7267	0.7512
+几何感知体素编码器和通道注意力稀疏编码器	0.7254	0.7490
完整方法	0.7285	0.7520

4.4 对比实验

在阴暗天气下路口场景中，通过消融实验验证DASE-BEV各模块对车辆语

义信息感知的贡献，图4- 4为太阳光照射下的路口的可视化结果。

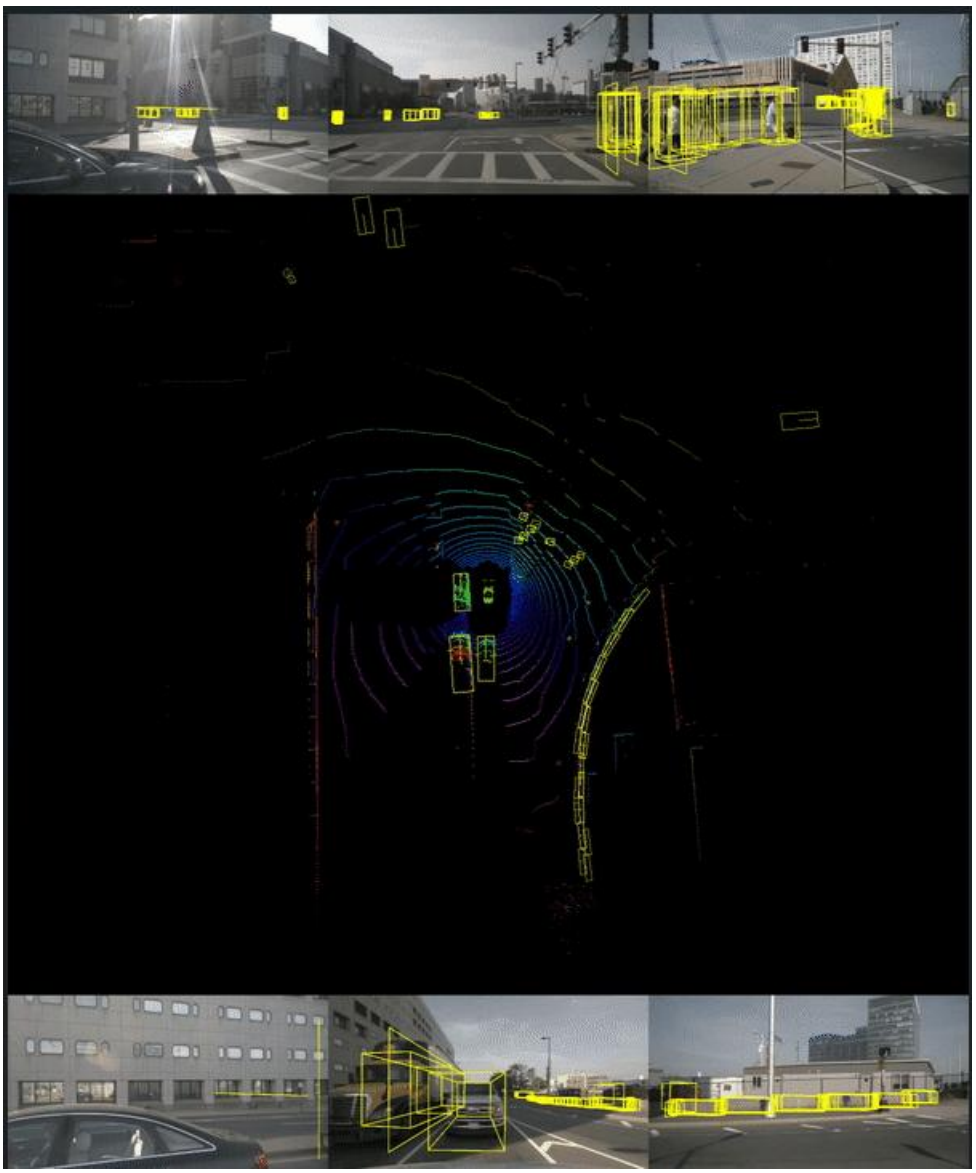

图 4- 5 太阳光照射下的路口可视化结果

为全面验证本方法在复杂场景中的三维目标建模能力，我们选取一个典型的静止行人场景，与多个主流BEV感知方法进行对比，包括 SparseFusion、Lift-Attend-Splat 和 RCBEVDet++。比较指标涵盖目标的空间位置、几何尺寸、旋转姿态（以四元数形式表示）及速度估计等核心参数，相关结果汇总于表4-8和4-9中。

从空间定位精度上看，我方方法在三轴坐标估计中整体误差控制优良，预测位置为［631. 67，1671. 71，-0. 86］，与真实标注［637. 141，1636. 252，-0. 235］保持较小偏差，优于 RCBEVDet++（误差最大，偏移近 70 米）和 Lift-Attend-Splat（偏移 34 米），显著提升了模型在稀疏场景下对静态小目标

的还原能力。在尺寸预测方面，我方方法所估计的尺寸为［0. 6823，0. 7337，1. 7840］，与真实值［0. 621，0. 647，1. 778］高度吻合，显示出更强的结构建模能力。其中，宽度和长度预测误差控制在 ±10 cm 以内，相较于其他方法尺寸偏差更小，表现出更精确的尺度感知能力。旋转姿态方面，我方方法采用四元数［-0. 8246，-0. 0089，-0. 0018，-0. 5657］表示目标朝向，旋转方向更接近真实标注 ［-0. 62，0. 00，-0. 01，0. 78］，在整体角度上较 Lift-Attend-Splat 和 RCBEVDet++ 存在更小的方向漂移，特别是在绕竖直轴方向的旋转估计上更为稳定。在速度感知上，真实静态行人的速度为［0. 0001，-0. 0003］。我方方法预测速度为［0. 0006，0. 0001］，不仅准确保持了目标“静止”属性（状态标注为 *pedestrian. standing*），同时在数值上也与 Ground Truth 几乎无偏离，避免了 SparseFusion 出现的“静止目标误判为移动”问题。

本文提出的双重注意力与语义增强机制在各项关键指标上均优于现有代表性方法，特别是在小体积静态目标的精细建模上展现出更强的鲁棒性与精度。这进一步验证了本方法在稀疏感知场景中对细粒度语义信息的高效融合能力。

表 4- 8 光照路口下语义信息感知部分结果

配置	位置［x, y, z］	尺寸[长, 宽. 高］	旋转角(四元数 w, x, y, . z)
真实标注	［637. 141, 1636. 252, -0. 235］	［0. 621, 0. 647, 1. 778］	［-0. 62, 0. 00, -0. 01, 0. 78］
SparseFusion	［636. 41, 1645. 39, -0. 95］	［0. 66, 0. 72, 1. 76］	［-0. 85, -0. 01, -0. 00, -0. 53］
Lift-Attend-Splat	［629. 58, 1671. 02, -0. 91］	［0. 68, 0. 72, 1. 80］	［-0. 9552, -0. 0041, -0. 0081, 0. 2960］
RCBEVDet++	［569. 23, 1675. 79, 1. 47］	［0. 65, 0. 68, 1. 81］	［-0. 68, -0. 01, 0. 00, -0. 73］
完整方法	［631. 67, 1671. 71, -0. 86］	［0. 6823, 0. 7337, 1. 7840］	［-0. 8246, -0. 0089, -0. 0018, -0. 5657］

表 4- 9 光照路口下语义信息感知部分结果

配置	速度（m/s）	物体类别	物体状态
真实标注	[0. 0001, -0. 0003]	pedestrian	pedestrian. standing
SparseFusion	[0. 0009, -0. 0006]	pedestrian	pedestrian. moving
Lift-Attend-Splat	[0. 0007, -0. 0003]	pedestrian	pedestrian. standing
RCBEVDet++	[0. 0009, -0. 0005]	pedestrian	pedestrian. standing
本文方法	[0, 0006, 0. 0001]	pedestrian	pedestrian. standing

为了全面评估各方法在相同数据条件下的性能表现，我们将本文方法与SparseFusion、RCBEVDet++和Lift-Attend-Splat进行了对比实验，此外在细分指标中我们将加入基线模型（BEVfusion和dal）的对比，因为基线模型的mAP和NDS我们已经在笑容中进行了对比，因此在此不在说明。所有方法均在三年以内并均以LiDAR＋Camera作为输入数据源并在相同的训练集与评测协议下进行训练与验证[23][24][25]。表4-10列出了各方法在mAP（平均精度）和NDS（综合检测和语义信息得分）两项关键指标上的结果，并简要说明其技术路线。

表 4- 10 主要指标对比

方法	技术路线	mAP	NDS	数据来源
本文方法	几何加权体素编码+混合注意力金字塔	0. 7285	0. 7520	Lidar ＋ camera
SparseFusion	动态稀疏特征关联	0. 720	0. 738	Lidar ＋ camera
RCBEVDet++	递归上下文建模+BEV 特征更新	0. 673	0. 727	Lidar ＋ camera
Lift-Attend-Splat	Lift-Attent-Splat 三维特征生成与融合	0. 715	0. 736	Lidar ＋ camera

从表中可以看出，SparseFusion采用的动态稀疏关联机制在点云与图像特征匹配方面具有一定优势，但仅仅将稀疏特征与二维图像特征进行自适应匹配，无

法充分注入几何先验，因此在多模态信息交互与综合性能上略逊于本文方法。RCBEVDet++通过递归上下文建模对BEV特征进行层级动态更新，能够在一定程度上增强空间推理能力，但其在高分辨率图像细节与深度几何先验之间的关联不足，导致mAP与NDS相对较低。Lift-Attend-Splat则通过将多视角图像特征映射到三维空间，并在BEV格网上进行注意力加权整合，能够充分挖掘图像细节信息，但其深度估计仍依赖网络学习，缺乏显式的几何加权策略，从而在大场景下的几何一致性上有所欠缺。相比之下，本文方法在点云体素化阶段即引入几何加权体素编码，使得关键的深度与方向信息得到优先保留；与此同时，我们设计的混合注意力金字塔能在不同尺度上对图像特征进行全局与局部的自适应融合，从而更好地兼顾视觉纹理和几何结构的互补性。正因如此，在相同训练数据和评测环境下，本文方法以72.85%的mAP和75.20%的NDS分别领先SparseFusion和Lift-Attend-Splat约0.85个百分点和1.60个百分点，以及领先RCBEVDet++约5.55个百分点和2.50个百分点，充分展现了几何加权与注意力机制相结合的多模态融合优势。

表4-11列出了本文方法与SparseFusion、RCBEVDet++、Lift-Attend-Splat在mATE（平均距离误差）、mASE（平均尺度误差）、mAOE（平均方向误差）、mAVE（平均速度误差）和mAAE（平均朝向误差）以及同样的mAP和NDS指标下的评测结果。

表 4- 11 细分指标对比

方法	mATE	mASE	mAOE	mAVE	mAAE
本文方法	0.2573	0.2379	0.3068	0.1944	0.1260
SparseFusion	0.258	0.243	0.329	0.265	0.131
RCBEVDet++	0.341	0.234	0.241	0.147	0.130
Lift-Attend-Splat	0.251	0.234	0.313	0.283	0.131
基线模型	0.2530	0.2441	0.3341	0.1880	0.1297

从表4-11的结果可以看出，本文的方法在整体上优于基线模型以及其他模型，细分指标大多为最好或是第二好。SparseFusion与本文方法在距离估计（mATE）上表现接近，但由于SparseFusion仅依赖稀疏关联来匹配点云与图像信息，其在速度估计（mAVE=0.2650）和方向估计（mAOE=0.3290）方面仍然保持较高误差。相比之下，本文方法通过几何加权体素编码可优先保留深度与边

缘信息，从而使得距离误差 mATE=0. 2573 略优于 SparseFusion；更重要的是，在速度估计与方向估计上分别达到 0. 1944 和 0. 3068，远低于 SparseFusion 的 0. 2650 和 0. 3290，这不仅验证了几何加权对动态属性捕捉的增强，又进一步说明混合注意力金字塔在多尺度图像特征融合时对速度和方向线索的自适应强调。RCBEVDet++在距离估计 mATE=0. 3410 方面误差较大，主要是由于其仅在 BEV 特征层面进行递归上下文建模，缺乏直接的几何优先策略，因此在速度与尺度估计等多项细分指标上难以与其他方法抗衡。尽管 RCBEVDet++在方向估计上表现较好（mAOE=0. 2410），但其较低的 mAP（0. 6730）和综合 NDS（0. 7270）仍然远落后于本文方法。Lift−Attend−Splat 依靠 Lift 和 Attend 模块从图像端获取深度信息，其在距离估计（mATE=0. 2510）和尺度估计（mASE=0. 2340）方面具备一定优势，但在速度估计（mAVE=0. 2830）与朝向估计（mAAE=0. 1310）上存在较大误差，最终导致 NDS=0. 7360。相比之下，本文方法的每项细分误差指标均实现了较大程度的优化，使得整体现象表现为 mAP 和 NDS 双优。

这一细致分析表明，将几何加权策略与多尺度注意力融合相结合，不仅能在距离与尺度等几何属性方面提供更可靠的估计，而且对动态属性（速度、方向、朝向）的捕捉也更为准确，从而全面提升了三维目标检测的性能。

第五章　结论与展望

5.1　研究成果总结

本文围绕于语义感知中的稀疏表示方法与多模态特征融合展开研究，面向当前 BEV 范式下模型精度与计算效率的权衡问题，聚焦于稀疏体素编码、特征金字塔融合以及体素特征提取三个关键模块，分别提出了轻量级注意力机制与无需训练参与的权重分配策略。针对稀疏体素编码部分，本文引入基于距离感知的加权平均方式，无需引入额外可训练参数，即可实现对体素内点云分布差异的有效建模，有助于提升稀疏表示的表达能力。在特征融合模块中，借助金字塔结构中不同尺度特征之间的几何关系，设计了融合引导权重，使得浅层细节信息和深层语义信息能够更合理地聚合，有效缓解尺度不一致问题。针对体素特征提取阶段，结合局部结构特征，引入轻量级注意力机制，对稀疏特征图中关键信息区域进行动态加权，从而提升目标边界识别能力。实验部分在 nuScenes 数据集上进行评估，结果显示本文提出的方法在不引入额外模型参数、不增加推理时延的前提下，能显著提升检测精度，尤其在小目标识别和遮挡场景中表现出更高的鲁棒性，验证了所提方法的有效性和工程适用性。

5.2　实际应用价值

在自动驾驶等对实时性和精度要求极高的任务场景中，传统 3D 目标检测模型常常面临精度与效率难以兼顾的问题。本文提出的多项改进策略均为轻量级结构设计，不依赖额外的深层神经网络模块或庞大的计算资源，因而具有良好的可部署性。所提出的体素加权策略和多尺度融合机制均可以无缝集成至现有主流的 BEV 感知框架中，如 BEVDet、BEVFusion 等，作为前处理或中间处理模块提升其特征建模能力。其不增加模型复杂度的设计使得该方法特别适合于计算资源受限的边缘设备部署场景，有助于提升自动驾驶系统在城市复杂交通环境中的目标感知鲁棒性和精度。此外，本文在不同模块中使用的注意力和稀疏引导机制也为后续多模态融合感知系统的设计提供了新的思路，对于解决 LiDAR–相机数据融合中的模态对齐与信息冗余问题具有一定的参考价值。

5.3　未来研究方向

尽管本文提出的方法在多个维度上取得了显著性能提升，但仍存在若干值得深入研究的方向。首先，在未来的工作中，可以进一步探索更高效的跨模态稀疏表示结构，将图像语义信息与点云几何结构进行联合建模，以实现更深层次的特

征互补。其次，当前模型仍依赖预定义的体素划分方式，未来可尝试引入数据驱动的自适应体素生成机制，实现对非结构化点云的更灵活表达。此外，本文提出的轻量级注意力机制主要应用于稀疏空间中的特征增强，后续可结合动态掩码或 Transformer 框架，引入更强的全局建模能力。在应用层面上，可结合多帧时序信息或 BEV 轨迹建模，进一步提升模型在连续感知场景中的稳定性和预测一致性，拓展在自动驾驶、机器人感知等领域的实际应用能力。同时，针对端到端感知-决策系统，还需将所提检测增强策略与后续的跟踪、行为预测等任务进行协同优化，实现真正意义上的多任务联合建模。

参考文献

[1] Badue, Claudine Santos, Rânik Guidolini, Raphael V. Carneiro, Pedro Azevedo, Vinicius B. Cardoso, Avelino Forechi, Luan F. R. Jesus, Rodrigo Berriel, Thiago Meireles Paixão, Filipe Wall Mutz, Thiago Oliveira-Santos and Alberto Ferreira de Souza. “Self-Driving Cars: A Survey.” *ArXiv* abs/1901.04407 (2019): n. pag.

[2] X. Chen, H. Ma, J. Wan, B. Li and T. Xia, "Multi-view 3D Object Detection Network for Autonomous Driving," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 6526-6534, doi: 10.1109/CVPR.2017.691.

[3] J. Ku, M. Mozifian, J. Lee, A. Harakeh and S. L. Waslander, "Joint 3D Proposal Generation and Object Detection from View Aggregation," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 2018, pp. 1-8, doi: 10.1109/IROS.2018.8594049.

[4] Liu, Zhijian, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus and Song Han. “BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation.” *2023 IEEE International Conference on Robotics and Automation (ICRA)* (2022): 2774-2781.

[5] Philion, Jonah and Sanja Fidler. “Lift, Splat, Shoot: Encoding Images From Arbitrary Camera Rigs by Implicitly Unprojecting to 3D.” *European Conference on Computer Vision* (2020).

[6] Huang, Junjie, Guan Huang, Zheng Zhu and Dalong Du. “BEVDet: High-performance Multi-camera 3D Object Detection in Bird-Eye-View.” *ArXiv* abs/2112.11790 (2021): n. pag.

[7] Qian, Rui, Xin Lai and Xirong Li. “3D Object Detection for Autonomous Driving: A Survey.” *Pattern Recognit.* 130 (2021): 108796.

[8] Vora, Sourabh, Alex H. Lang, Bassam Helou and Oscar Beijbom. “PointPainting: Sequential Fusion for 3D Object Detection.” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019): 4603-4611.

[9] Xu, Danfei, Dragomir Anguelov and Ashesh Jain. “PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation.” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017): 244-253.

[10] Lee, Ms. Seunghui. “Evaluation of the usefulness of the MV3D RECON

technique with Bone Removal in CT Brain Angiography." *Journal of Medical Imaging and Radiation Sciences* (2024): n. pag.

[11] Qi, C., W. Liu, Chenxia Wu, Hao Su and Leonidas J. Guibas. "Frustum PointNets for 3D Object Detection from RGB-D Data." *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017): 918-927.

[12] Nabati, Ramin and Hairong Qi. "CenterFusion: Center-based Radar and Camera Fusion for 3D Object Detection." *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2020): 1526-1535.

[13] Yang, Zeyu, Jia-Qing Chen, Zhenwei Miao, Wei Li, Xiatian Zhu and Li Zhang. "DeepInteraction: 3D Object Detection via Modality Interaction." *ArXiv* abs/2208.11112 (2022): n. pag.

[14] Mao, Jiageng, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu and Chunjing Xu. "Voxel Transformer for 3D Object Detection." *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021): 3144-3153.

[15] Zhou, Chunting, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke S. Zettlemoyer and Omer Levy. "Transfusion: Predict the Next Token and Diffuse Images with One Multi-Modal Model." *ArXiv* abs/2408.11039 (2024): n. pag.

[16] Li, Wenxi, Yuchen Guo, Jilai Zheng, Haozhe Lin, Chao Ma, Lu Fang and Xiaokang Yang. "SparseFormer: Detecting Objects in HRW Shots via Sparse Vision Transformer." *Proceedings of the 32nd ACM International Conference on Multimedia* (2024): n. pag.

[17] Graham, Benjamin, Martin Engelcke and Laurens van der Maaten. "3D Semantic Segmentation with Submanifold Sparse Convolutional Networks." *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017): 9224-9232.

[18] Lin, Tsung-Yi, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan and Serge J. Belongie. "Feature Pyramid Networks for Object Detection." *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016): 936-944.

[19] Hu, Jie, Li Shen, Samuel Albanie, Gang Sun and Enhua Wu. "Squeeze-and-Excitation Networks." *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017): 7132-7141.

[20] Woo, Sanghyun, Jongchan Park, Joon-Young Lee and In-So Kweon. "CBAM:

Convolutional Block Attention Module.” *ArXiv* abs/1807.06521 (2018): n. pag.

[21] Zhou, Yin and Oncel Tuzel. “VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection.” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017): 4490-4499.

[22] Caesar, Holger, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yuxin Pan, Giancarlo Baldan and Oscar Beijbom. “nuScenes: A Multimodal Dataset for Autonomous Driving.” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019): 11618-11628.

[23] Xie, Yichen, Chenfeng Xu, Marie-Julie Rakotosaona, Patrick Rim, Federico Tombari, Kurt Keutzer, Masayoshi Tomizuka and Wei Zhan. “SparseFusion: Fusing Multi-Modal Sparse Representations for Multi-Sensor 3D Object Detection.” *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2023): 17545-17556.

[24] Lin, Zhiwei, Zhe Liu, Zhongyu Xia, Xinhao Wang, Yongtao Wang, Shengxiang Qi, Yang Dong, Nan Dong, Le Zhang and Ce Zhu. “RCBEVDet: Radar-Camera Fusion in Bird's Eye View for 3D Object Detection.” *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024): 14928-14937.

[25] Gunn, James, Zygmunt Lenyk, Anuj Sharma, Andrea Donati, Alexandru Buburuzan, John Redford and Romain Mueller. “Lift-Attend-Splat: Bird’s-eye-view camera-lidar fusion using transformers.” *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2023): 4526-4536.

致 谢

本论文的顺利完成离不开许多人的帮助与支持，在此我谨向所有给予我指导、鼓励与帮助的人表示最诚挚的感谢。

首先，衷心感谢我的导师，在整个毕业设计过程中，我的指导老师以严谨的学术态度和丰富的专业知识给予我悉心指导和耐心帮助，从选题方向的确定到论文结构的调整，每一个关键环节都离不开老师的细致指导与启发。同时，我要感谢过去四年中所有授课老师，是您们的专业传授和言传身教为我打下了坚实的专业基础。也要感谢实验室的同学们和项目合作伙伴，是你们在学习与实践中给予我支持与交流，让我在技术路上不再孤单。

感谢我的父母与家人，你们始终如一的理解与鼓励是我坚持前行的最大动力。无论面对什么样的困难，家始终是我最坚强的后盾。也感谢一路陪伴我的朋友，是你们让我在紧张的学习之余仍能拥有温暖与欢笑。

最后，向所有关心和支持我完成本次毕业设计的师长、同学与亲友表示诚挚的谢意！

毕业设计小结

转眼间，四年的大学生活即将画上句号。回首这段时光，从懵懂走向成熟，从基础课程的学习到专业技能的积累，再到毕业设计的独立完成，每一个阶段都承载着成长与突破。软件工程专业不仅培养了我扎实的编程能力和工程思维，也让我在实际项目中体会到了团队合作与系统设计的重要性。在一次次的调试与重构中，我学会了面对挑战、解决问题，也逐渐明确了自己未来的发展方向。

在大学四年中，我不仅收获了专业知识，也结识了许多志同道合的同学和良师益友。从算法课堂到创新实验室，从课程设计到竞赛项目，每一次经历都让我受益匪浅。这一段求学旅程不仅是技术能力的提升，更是人生视野的拓展与价值观的沉淀。相信这段经历将在今后的工作与生活中持续发挥作用，成为我不断前行的动力。

附 录

几何感知体素编码器（Geo-Aware VFE）：

```
@VOXEL_ENCODERS.register_module()
class HardSimpleVFE(nn.Module):
    def __init__(self, num_features=4):
        super(HardSimpleVFE, self).__init__()
        self.num_features = num_features
        self.fp16_enabled = False

    @force_fp32(out_fp16=True)
    def forward(self, features, num_points, coors):
        points = features[:, :, :self.num_features] #(N, M, C)
        # 原始简单均值
        points_mean = points.sum(dim=1) / num_points.type_as(points).view(-1, 1) #(N, C)
        # 计算每个点到体素质心（初始均值）的距离
        diff = points - points_mean.unsqueeze(1) #(N, M, C)
        dist = torch.norm(diff, dim=2) #(N, M)
        # 基于距离的权重（避免除零）
        weights = 1.0 / (dist + 1e-6) #(N, M)
        # 计算加权平均
        weighted_sum = (points * weights.unsqueeze(-1)).sum(dim=1) #(N, C)
        weight_norm = weights.sum(dim=1, keepdim=True) #(N, 1)
        weighted_mean = weighted_sum / weight_norm #(N, C)
        return weighted_mean.contiguous()
```

通道增强稀疏编码器（SE-Sparse Encoder）：

```
@MIDDLE_ENCODERS.register_module()
class SparseEncoder(nn.Module):
```

```
    def __init__(self,
                 in_channels,
                 sparse_shape,
                 order=('conv', 'norm', 'act'),
                 norm_cfg=dict(type='BN1d', eps=1e-3, momentum=0.01),
                 base_channels=16,
                 output_channels=128,
                 encoder_channels=((16, ), (32, 32, 32), (64, 64, 64), (64,
64, 64)),
                 encoder_paddings=((1, ), (1, 1, 1), (1, 1, 1), ((0, 1,
1), 1, 1)),
                 block_type='conv_module'):
        super().__init__()
        assert block_type in ['conv_module', 'basicblock']
        self.sparse_shape = sparse_shape
        self.in_channels = in_channels
        self.order = order
        self.base_channels = base_channels
        self.output_channels = output_channels
        self.encoder_channels = encoder_channels
        self.encoder_paddings = encoder_paddings
        self.stage_num = len(self.encoder_channels)
        self.fp16_enabled = False

        assert isinstance(order, tuple) and len(order) == 3
        assert set(order) == {'conv', 'norm', 'act'}

        if self.order[0] != 'conv':
            self.conv_input = make_sparse_convmodule(
                in_channels,
                self.base_channels,
                3,
                norm_cfg=norm_cfg,
                padding=1,
                indice_key='subm1',
```

```
                conv_type='SubMConv3d',
                order=('conv', ))
        else:
            self.conv_input = make_sparse_convmodule(
                in_channels,
                self.base_channels,
                3,
                norm_cfg=norm_cfg,
                padding=1,
                indice_key='subm1',
                conv_type='SubMConv3d')

        encoder_out_channels = self.make_encoder_layers(
            make_sparse_convmodule,
            norm_cfg,
            self.base_channels,
            block_type=block_type)

        self.conv_out = make_sparse_convmodule(
            encoder_out_channels,
            self.output_channels,
            kernel_size=(3, 1, 1),
            stride=(2, 1, 1),
            norm_cfg=norm_cfg,
            padding=0,
            indice_key='spconv_down2',
            conv_type='SparseConv3d')


    @auto_fp16(apply_to=('voxel_features', ))
    def forward(self, voxel_features, coors, batch_size):
        coors = coors.int()
        input_sp_tensor = SparseConvTensor(voxel_features, coors,
                                           self.sparse_shape, batch_size)
        x = self.conv_input(input_sp_tensor)
```

```
        encode_features = []
        for encoder_layer in self.encoder_layers:
            x = encoder_layer(x)
            encode_features.append(x)

        out = self.conv_out(encode_features[-1])
        spatial_features = out.dense()
        N, C, D, H, W = spatial_features.shape
        spatial_features = spatial_features.view(N, C * D, H, W)

        device = spatial_features.device
        if not hasattr(self, 'se_block') or self.se_block.channel != C * D:
            self.se_block = SparseSEBlock(C * D).to(device)
        spatial_features = spatial_features.view(N, C * D, H, W)
        spatial_features = self.se_block(spatial_features)
        return spatial_features

    def make_encoder_layers(self,
                            make_block,
                            norm_cfg,
                            in_channels,
                            block_type='conv_module',
                            conv_cfg=dict(type='SubMConv3d')):
        assert block_type in ['conv_module', 'basicblock']
        self.encoder_layers = SparseSequential()

        for i, blocks in enumerate(self.encoder_channels):
            blocks_list = []
            for j, out_channels in enumerate(tuple(blocks)):
                padding = tuple(self.encoder_paddings[i])[j]
                if i != 0 and j == 0 and block_type == 'conv_module':
                    blocks_list.append(
                        make_block(
                            in_channels,
```

```
                    out_channels,
                    3,
                    norm_cfg=norm_cfg,
                    stride=2,
                    padding=padding,
                    indice_key=f'spconv{i + 1}',
                    conv_type='SparseConv3d'))
elif block_type == 'basicblock':
    if j == len(blocks) - 1 and i != len(
                    self.encoder_channels) - 1:
        blocks_list.append(
            make_block(
                in_channels,
                out_channels,
                3,
                norm_cfg=norm_cfg,
                stride=2,
                padding=padding,
                indice_key=f'spconv{i + 1}',
                conv_type='SparseConv3d'))
    else:
        blocks_list.append(
            SparseBasicBlock(
                out_channels,
                out_channels,
                norm_cfg=norm_cfg,
                conv_cfg=conv_cfg))
else:
    blocks_list.append(
        make_block(
            in_channels,
            out_channels,
            3,
            norm_cfg=norm_cfg,
            padding=padding,
```

```
                                indice_key=f'subm{i + 1}',
                                conv_type='SubMConv3d'))
                    in_channels = out_channels
                stage_name = f'encoder_layer{i + 1}'
                stage_layers = SparseSequential(*blocks_list)
                self.encoder_layers.add_module(stage_name, stage_layers)
        return out_channels
```

双注意力特征金字塔（Dual-Attention FPN）：

```
@NECKS.register_module()
class CustomFPN(BaseModule):


    def __init__(self,
                 in_channels,
                 out_channels,
                 num_outs,
                 start_level=0,
                 end_level=-1,
                 out_ids=[],
                 add_extra_convs=False,
                 relu_before_extra_convs=False,
                 no_norm_on_lateral=False,
                 conv_cfg=None,
                 norm_cfg=None,
                 act_cfg=None,
                 upsample_cfg=dict(mode='nearest'),
                 init_cfg=dict(
                     type='Xavier', layer='Conv2d',
distribution='uniform')):
        super(CustomFPN, self).__init__(init_cfg)
        assert isinstance(in_channels, list)
        self.in_channels = in_channels
        self.out_channels = out_channels
        self.num_ins = len(in_channels)
        self.num_outs = num_outs
```

```
self. relu_before_extra_convs = relu_before_extra_convs
self. no_norm_on_lateral = no_norm_on_lateral
self. fp16_enabled = False
self. upsample_cfg = upsample_cfg. copy()
self. out_ids = out_ids
if end_level == -1:
    self. backbone_end_level = self. num_ins
else:
    self. backbone_end_level = end_level
    assert end_level <= len(in_channels)
    assert num_outs == end_level - start_level
self. start_level = start_level
self. end_level = end_level
self. add_extra_convs = add_extra_convs
assert isinstance(add_extra_convs, (str, bool))
if isinstance(add_extra_convs, str):
    assert add_extra_convs in ('on_input', 'on_lateral', 'on_output')
elif add_extra_convs:
    self. add_extra_convs = 'on_input'

self. lateral_convs = nn. ModuleList()
self. fpn_convs = nn. ModuleList()
# 加入 SE 與 CBAM 混合注意力模塊
self. se_blocks = nn. ModuleList()
self. cbam_blocks = nn. ModuleList()

for i in range(self. start_level, self. backbone_end_level):
    l_conv = ConvModule(
        in_channels[i],
        out_channels,
        1,
        conv_cfg=conv_cfg,
        norm_cfg=norm_cfg if not self. no_norm_on_lateral else None,
        act_cfg=act_cfg,
        inplace=False)
```

```
            self. lateral_convs. append(l_conv)
            if i in self. out_ids:
                fpn_conv = ConvModule(
                    out_channels,
                    out_channels,
                    3,
                    padding=1,
                    conv_cfg=conv_cfg,
                    norm_cfg=norm_cfg,
                    act_cfg=act_cfg,
                    inplace=False)
                self. fpn_convs. append(fpn_conv)
            self. se_blocks. append(SEBlock(out_channels))
            self. cbam_blocks. append(CBAMBlock(out_channels))

        extra_levels = num_outs - self. backbone_end_level + self. start_level
        if self. add_extra_convs and extra_levels >= 1:
            for i in range(extra_levels):
                if i == 0 and self. add_extra_convs == 'on_input':
                    in_channels_ = self. in_channels[self. backbone_end_level -
1]
                else:
                    in_channels_ = out_channels
                extra_fpn_conv = ConvModule(
                    in_channels_,
                    out_channels,
                    3,
                    stride=2,
                    padding=1,
                    conv_cfg=conv_cfg,
                    norm_cfg=norm_cfg,
                    act_cfg=act_cfg,
                    inplace=False)
                self. fpn_convs. append(extra_fpn_conv)
                # 額外層也加 SE 和 CBAM
```

```
                self.se_blocks.append(SEBlock(out_channels))
                self.cbam_blocks.append(CBAMBlock(out_channels))

    @auto_fp16()
    def forward(self, inputs):
        assert len(inputs) == len(self.in_channels)

        laterals = [
            lateral_conv(inputs[i + self.start_level])
            for i, lateral_conv in enumerate(self.lateral_convs)
        ]
        used_backbone_levels = len(laterals)
        for i in range(used_backbone_levels - 1, 0, -1):
            if 'scale_factor' in self.upsample_cfg:
                laterals[i - 1] += F.interpolate(laterals[i],
                                                 **self.upsample_cfg)
            else:
                prev_shape = laterals[i - 1].shape[2:]
                laterals[i - 1] += F.interpolate(
                    laterals[i], size=prev_shape, **self.upsample_cfg)

        for i in range(len(laterals)):
            se_feat = self.se_blocks[i](laterals[i])
            cbam_feat = self.cbam_blocks[i](laterals[i])
            laterals[i] = 0.6 * se_feat + 0.4 * cbam_feat


        max_h = max([feat.shape[2] for feat in laterals])
        max_w = max([feat.shape[3] for feat in laterals])
        resized_feats = [F.interpolate(feat, size=(max_h, max_w),
mode='bilinear', align_corners=False)
                         for feat in laterals]
        global_concat = torch.cat(resized_feats, dim=1)
        fusion_conv = nn.Conv2d(global_concat.shape[1], self.out_channels,
1).to(global_concat.device)
```

```
        global_fused = fusion_conv(global_concat)
        for i in range(len(laterals)):
            laterals[i] = laterals[i] + F.interpolate(global_fused,
size=laterals[i].shape[2:], mode='bilinear', align_corners=False)

        outs = [self.fpn_convs[i](laterals[i]) for i in self.out_ids]
        if self.num_outs > len(outs):
            if not self.add_extra_convs:
                for i in range(self.num_outs - used_backbone_levels):
                    outs.append(F.max_pool2d(outs[-1], 1, stride=2))
            else:
                if self.add_extra_convs == 'on_input':
                    extra_source = inputs[self.backbone_end_level - 1]
                elif self.add_extra_convs == 'on_lateral':
                    extra_source = laterals[-1]
                elif self.add_extra_convs == 'on_output':
                    extra_source = outs[-1]
                else:
                    raise NotImplementedError

outs.append(self.fpn_convs[used_backbone_levels](extra_source))
                for i in range(used_backbone_levels + 1, self.num_outs):
                    if self.relu_before_extra_convs:
                        outs.append(self.fpn_convs[i](F.relu(outs[-1])))
                    else:
                        outs.append(self.fpn_convs[i](outs[-1]))
        return outs[0]
```