arXiv:2301.02364v3 [cs.CV] 6 Nov 2023

以物体为查询：将任意二维目标检测器提升为三维检测器

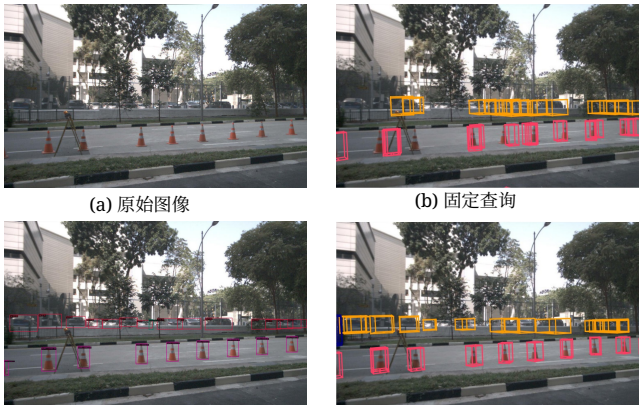
Zitian Wang[1] Zehao Huang[2] Jiahui Fu[1] Naiyan Wang[2] Si Liu[1]1北京航空航天大学人工智能研究院2TuSimple{wangzt.kghl,zehaohuang18,winsty}@gmail.com {jiahuifu,liusi}@buaa.edu.cn

摘要

近年来，基于多视图图像的三维目标检测受到了广泛关注。现有方法主要从多视图图像中构建三维表示，并采用密集检测头进行目标检测，或利用分布在三维空间中的目标查询来定位目标。本文设计了多视图二维目标引导的三维目标检测器（*MV2D*），可将任意二维目标检测器提升为多视图三维目标检测器。由于二维检测能够为物体存在性提供有价值的先验信息，*MV2D* 利用二维检测器生成由丰富图像语义调节的目标查询。这些动态生成的查询有助于 *MV2D* 召回视场中的目标，并展现出强大的三维目标定位能力。针对生成的查询，我们设计了一个稀疏交叉注意力模块，促使其关注特定目标的特征，从而抑制噪声干扰。在 *nuScenes* 数据集上的评估结果表明，动态目标查询和稀疏特征聚合能够提升三维检测能力。*MV2D* 的性能也优于现有方法，达到了最先进水平。我们希望 *MV2D* 能成为未来研究的新基线。代码见 https://github.com/tusen-ai/MV2D。

1. 引言

近年来，基于摄像头的三维目标检测在不受约束的真实场景中引起了广泛关注。早期的单目三维目标检测方法 [30,45, 1, 36, 18, 39, 40] 通常遵循二维目标检测流程构建框架，直接根据单视角图像预测物体的三维位置和属性。尽管这些方法取得了显著进展，但它们无法利用周围摄像头的几何布局和多视角图像对应关系，而这些信息对于确定物体在现实世界中的三维位置至关重要。此外，将这些方法适配到多视角场景还依赖复杂的跨摄像头后处理，这进一步降低了效率和效果。为了解决这些问题，近期研究者[41, 16, 23, 22, 26] 提出直接基于多视图图像在三维世界空间中定位物体，为基于视觉的三维目标检测提供了一种新范式。



(a) 原始图像 (b) 固定查询 (c) 二维检测 (d) 由二维检测生成的查询

图1. MV2D 的动机。采用固定目标查询的三维检测器（固定查询是指查询对于不同输入保持不变）可能会错误定位或忽略某些目标（b），而二维检测器却能成功检测到这些目标（c）。如果根据二维检测器生成目标查询，三维检测器就能给出更精确的位置（d）。

根据融合特征的表示方式，当前的多视图三维目标检测方法主要分为两类：稠密三维方法和稀疏查询方法。具体而言，稠密三维方法会将多视图特征映射到三维空间中，例如鸟瞰图（BEV）特征空间 [16, 22, 23, 5] 或体素特征空间 [35, 21]。然而，由于计算成本与三维空间的范围成正比，因此这类方法不可避免地难以扩展到大规模场景 [9]。另一类基于查询的方法 [41, 26] 采用可学习的目标查询，从多视图图像中聚合特征，并基于查询特征预测目标边界框。虽然固定数量的目标查询避免了计算成本随三维空间增大而激增，但依赖经验先验设定查询数量和位置，可能会导致动态场景中出现误检或漏检目标。

本文旨在探索一种更可靠的目标定位方式。其动机源自二维目标检测方法的快速发展 [12, 34, 24, 37] ，这些方法能够在图像空间中生成高质量的二维边界框来定位目标。一个自然的想法是将每个二维检测结果用作后续三维检测任务的一个参考。基于此，我们设计了多视图二维目标引导的三维目标检测器（MV2D），它能够将任意二维检测器提升至三维检测，并使三维检测器能够充分利用二维检测领域的所有进展。

给定输入的多视角图像，我们首先从二维检测器获取二维检测结果，然后为每个二维边界框生成一个动态目标查询。不同于从多视角输入中的所有区域聚合特征，我们强制每个目标查询关注一个特定目标。为此，我们提出了一种基于二维检测结果和相机配置的高效相关特征选择方法。随后，将动态生成的目标查询及其嵌入三维位置的相关特征输入带有稀疏交叉注意力层的Transformer解码器。最后，更新后的目标查询预测最终的三维边界框。

得益于强大的二维检测性能，由二维检测结果生成的动态查询能够覆盖图像中出现的大多数目标。与固定查询相比，这带来了更高的精确率和召回率，尤其是对于尺寸较小且距离较远的目标。如图1所示，基于固定查询的三维检测器可能漏检的目标，在二维目标检测器的帮助下可以通过MV2D找回。从理论上讲，由于三维目标查询源自二维检测结果，我们的方法能够受益于所有现成的优秀二维检测器的改进。我们的贡献可总结如下：

- 我们提出了一个名为 MV2D 的框架，能够将任意二维目标检测器提升为多视角三维目标检测器。
- 我们证明，基于二维目标检测，从多视图图像中的特定相关区域进行动态目标查询和聚合，可以提升三维检测性能。
- 我们在标准 nuScenes 数据集上对 MV2D 进行了评估，其取得了当前最先进的性能。

2. 相关工作

2.1. 二维目标检测

二维目标检测旨在预测并定位二维图像中的物体，这是计算机视觉中的一项基础任务。RCNN 系列 [12, 11, 34, 24] 提出了一种用于二维目标检测的两阶段流程。这些方法首先生成一组可能包含图像中物体的区域提议，然后在第二阶段对物体进行分类和边界框回归。另一类研究者致力于在单阶段流程中检测物体 [33, 37, 25], , 旨在提供更快、便于部署的检测器。最近，DETR [3] 将Transformer编码器-解码器架构引入二维目标检测，并将目标检测构建为一个集合预测问题。在DETR中，采用一组可学习的目标查询与图像特征交互，并负责检测物体。Deformable DETR [47] 通过引入多尺度可变形注意力改进了DETR。此外，它将原版DETR转变为两阶段框架：首先根据编码器特征生成区域提议，然后将其作为目标查询发送至解码器进行进一步细化。Efficient DETR [43]研究了二维检测中的查询初始化，但并未探究如何将这种初始化迁移并调整至三维空间。

2.2. 基于视觉的三维目标检测

基于视觉的三维目标检测旨在从相机图像中预测三维边界框。许多早期工作 [39, 18, 1, 31, 36, 40] 基于单视图图像设置执行这项任务。这些方法大多侧重于实例深度估计，并使用额外的深度预测模块，为二维检测器补充深度信息。然而，在处理环绕自车的多视图图像时，这些方法需要从每个视角检测目标，再将其投影到同一三维空间，并通过繁琐的非极大值抑制后处理合并结果。

近期，一些工作尝试直接从多视图图像中检测三维空间内的目标。其中一类方法将二维图像提升至三维空间，再基于三维表示执行检测。ImVoxelNet [35] 构建三维体素化空间，并从多视角图像中采样图像特征，以获取体素表示。BEVFormer [23]利用密集的鸟瞰视图查询，通过可变形注意力 [47]投影并聚合多视图图像中的特征。PolarFormer [17] 引入极坐标表示来建模鸟瞰空间。BEVDet 和 BEVDepth [16, 22] 采用 Lift-Splat模块 [32] ，根据预测的深度分布将多视图图像特征转换为鸟瞰表示。尽管此类三维空间表示有助于统一多视图图像，但随着三维空间中检测范围的扩大，内存消耗和计算成本也会随之增加 [9]。

另一类工作采用可学习的目标查询来聚合图像特征，并遵循 DETR [3]预测目标。DETR3D [41] 根据目标查询生成三维参考点，并利用相机参数将其投影到多摄像头图像上。随后，从图像中采样对应的点特征，以此优化查询特征。与通过三维到二维查询投影建立固定映射不同，一些方法通过注意力机制学习灵活的

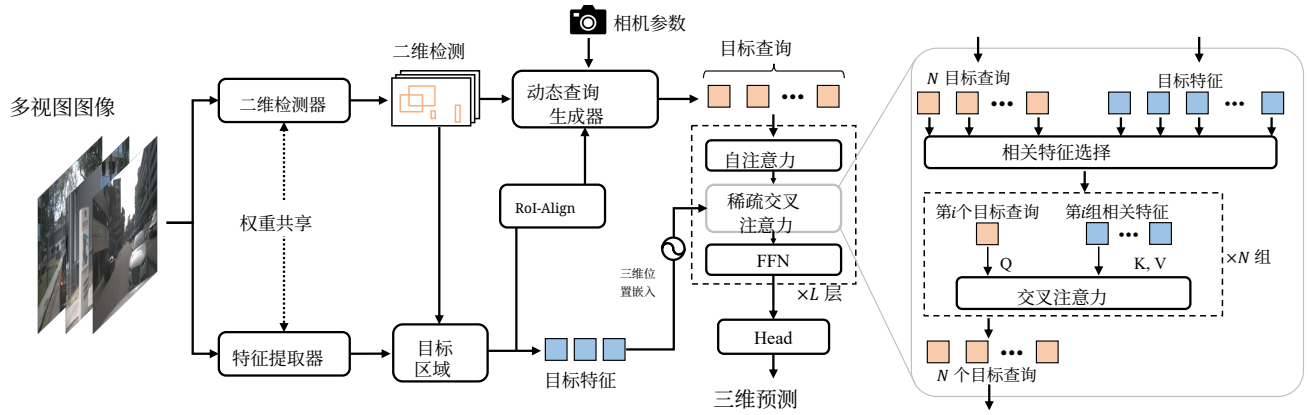


图 2. 所提出的 MV2D 框架。给定输入的多视图图像，特征提取器会提取图像特征图。同时，使用二维检测器获取各视角的二维检测结果。动态查询生成器以目标特征、二维检测框和相机参数为输入，初始化一组目标查询。对目标区域应用 RoI-Align，以获取供查询生成器使用的固定长度目标特征。目标区域内的所有特征均添加三维位置嵌入（三维位置嵌入） [26], 随后，将目标查询和目标特征输入解码器，以更新查询特征。与普通的 Transformer 解码器相比，MV2D 中的解码器采用稀疏交叉注意力，每个目标查询仅与其相关特征交互。最后，通过预测头处理更新后的目标查询，生成三维检测结果。

映射 [44, 26]。PETR 系列 [26, 27] 引入三维位置感知图像特征，并通过全局交叉注意力学习查询与图像特征之间的灵活映射。尽管如此，这些方法通常需要在三维空间中密集分布的目标查询，以确保目标具有足够的召回率。

本文提出了一种用于多视图三维目标检测的二维目标引导框架。基于二维检测器，我们的方法能够利用动态查询召回目标，并消除噪声和干扰项的影响。

3. 方法

3.1. 概述

MV2D 的整体流程如图 2 所示。给定 N 多视图图像 $\mathcal{I} = \{\mathbf{I}_v \mid 0 \le v < N\}$，首先使用骨干网络从输入图像中提取图像特征图 $\mathcal{F} = \{\mathbf{F}_v \mid 0 \le v < N\}$，其中 $\mathbf{F}_v \in \mathbb{R}^{H^f \times W^f \times C}$是从第 v 张图像中提取的特征图。为了进行二维目标检测，我们将二维目标检测器 Faster R-CNN [34], 应用于所有输入图像，从而得到一组二维目标边界框$\mathcal{B} = \{\mathbf{B}_v \mid 0 \le v < N\}$，其中 $\mathbf{B}_v \in \mathbb{R}^{M_v \times 4}$ 表示第 v 张图像中的预测二维边界框，而 M_v表示检测框的数量。实际上，二维检测器骨干网络的权重也可用于后续的特征提取。

不同于多视角三维目标检测中常见的、在整个数据集上采用固定目标查询的方法， [41, 26], MV2D 会根据输入的多视图图像生成目标查询。给定预测的二维边界框和提取的图像特征后，MV2D 使用 RoI-Align [13] 提取定长目标特征。随后，将目标特征与相应的边界框和相机参数一同输入动态目标查询生成器，以生成目标查询。对于每个目标查询，相关特征选择模块会根据二维检测结果和相机参数，从多视图图像中选取相关图像特征。然后，目标查询彼此交互，并通过 Transformer 解码器层 [38] 迭代整合相关目标特征中的信息。最后，使用简单的前馈网络（FFN）头，基于更新后的特征生成三维目标预测。

3.2. 动态目标查询生成

借助高效的二维目标检测器，可以准确呈现目标的存在，并将目标位置限定在图像的特定区域内，从而为在三维空间中定位目标提供有价值的先验信息。

为了从二维检测结果生成目标查询，我们提出了一种动态查询生成器，如图 3所示。动态查询生成器会在三维世界空间中为每个感兴趣区域推导出一个三维参考点 $\mathbf{p}_{ref} \in \mathbb{R}^3$。随后，通过位置编码层 $\mathbf{p}^{ref}$ 生成目标查询 [38]。

具体而言，给定所有图像的二维目标检测结果 $\mathcal{B}$ 和图像特征图 $\mathcal{F}$，我们首先通过 RoI-Align 提取目标 RoI特征 $\mathcal{O}$，其中 $\mathbf{O}_v \in \mathbb{R}^{M_v \times H^{roi} \times W^{roi} \times C}$ 是与二维目标边界框$\mathbf{B}_v$对应的 RoI特征：

$$\mathbf{O}_v = \text{RoI-Align}(\mathbf{F}_v, \mathbf{B}_v), \quad 0 \le v < N. \tag{1}$$

RoI特征$\mathbf{O}_v$ 包含足够的物体外观信息，可用于推断图像中的物体中心位置

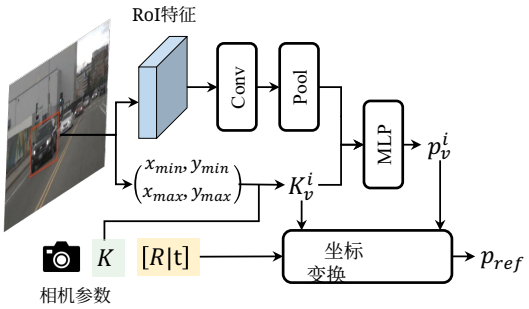


图 3. 动态对象查询生成器。

。然而，要直接从RoI特征进一步估计物体深度仍然很困难。由于原始图像空间中的物体区域会被缩放为固定大小的RoI，原始图像中包含的几何信息也随之丢失。

考虑到这一点，我们对每个RoI应用等效相机内参变换，使得对不同RoI执行的缩放操作等价于使用不同相机参数进行透视投影。因此，RoI坐标系中的点可以通过等效相机内参转换到三维世界空间：

$$\mathbf{p}_{ref} = [\mathbf{R}|\mathbf{t}]^{-1}(\mathbf{K}_v^i)^{-1}\,\mathbf{p}_v^i, \tag{2}$$

其中，$\mathbf{p}_v^i \in \mathbb{R}^4$ （齐次坐标）表示RoI坐标系中的2.5D点，$\mathbf{p}_{ref} \in \mathbb{R}^4$ 表示三维世界空间中的点，$\mathbf{K}_v^i$ 是该RoI的等效相机内参，且 $[\mathbf{R}|\mathbf{t}]$ 是相机外参。

设RoI尺寸为 $H^{roi} \times W^{roi}$ （例如， 7×7），第 i个视角中的第 v个二维对象边界框为$\mathbf{B}_v^i$ $=(x_{min}^i, y_{min}^i, x_{max}^i, y_{max}^i)$，原始相机内参矩阵为：

$$\mathbf{K}_v = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

第 i个RoI的等效相机内参可表示为：

$$\mathbf{K}_v^i = \begin{bmatrix} f_x * r_x & 0 & (o_x - x_{min}^i) * r_x & 0 \\ 0 & f_y * r_y & (o_y - y_{min}^i) * r_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

其中 $r_x = W^{roi}/(x_{max}^i - x_{min}^i)$ $r_y = H^{roi}/(y_{max}^i - y_{min}^i)$。

由于等效相机内参矩阵包含相机和物体的几何属性，我们采用一个小型网络，根据 RoI特征 $\mathbf{p}_v^i \in \mathbb{R}^4$ 和等效相机内参 $\mathbf{o}_v^i$ $\mathbf{K}_v^i$对物体位置进行隐式编码：

$$\mathbf{p}_v^i = \mathcal{H}(\text{MLP}(\text{Pool}(\text{Conv}(\mathbf{o}_v^i)); \mathbf{K}_v^i)), \tag{5}$$

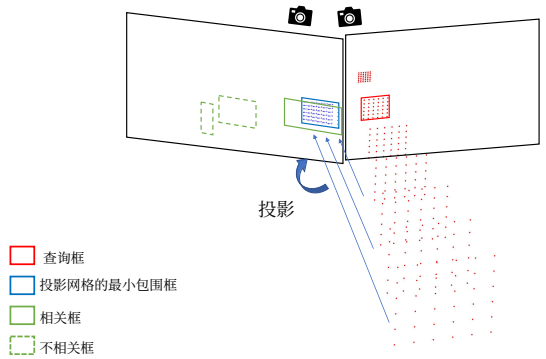


图4. 相关区域选择示意图。每个查询框根据三维网格生成一个离散化相机视锥。然后，将该相机视锥投影到另一视角的像素坐标上，以计算最小包围框。接着，根据相关区域与最小包围框的重叠情况选出相关区域。

其中，(;)表示特征拼接， $\mathcal{H}(\cdot)$表示齐次变换。

隐式编码的对象位置 $\mathbf{p}_v^i$可视为 RoI 坐标系中的三维对象位置，随后使用等效相机内参和相机外参将其变换到三维世界空间，如公式2所示。因此，变换后的三维位置充当参考点 $\mathbf{p}^{ref}$，用于在三维世界空间中定位对象。

通过这一变换，MV2D 基于二维检测结果和相机配置生成一组动态参考点 $\mathcal{P}_{ref} = \{\mathbf{P}_{ref,\,v} \in \mathbb{R}^{M_v \times 3} \mid 0 \leq v < N\}$ 随后，这些参考点经过位置编码层 [38]，以初始化一组目标查询 $\mathcal{Q} = \{\mathbf{Q}_v \in \mathbb{R}^{M_v \times C} \mid 0 \leq v < N\}$。具体而言，每个目标查询 $\mathbf{q}$ 由以下方式获得：

$$\mathbf{q} = \text{Linear}(PE), \tag{6}$$

$$PE_{[6i:6i+3]} = \sin(\mathbf{p}_{ref}/10000^{2i/C}), 0 \leq i < C/2, \tag{7}$$

$$PE_{[6i+3:6i+6]} = \cos(\mathbf{p}_{ref}/10000^{2i/C}), 0 \leq i < C/2. \tag{8}$$

3.3. 相关目标特征选择

每个目标仅出现在多视图图像中的一个子区域内。通过聚焦于与目标相关的区域，我们可以消除可能妨碍目标定位性能的干扰项和噪声。

为此，我们提出为每个目标查询的更新选择相关特征。由于二维目标检测器能够预测可靠的二维目标提议，检测到的目标边界框表明了包含目标最具区分性信息的区域。因此，我们将目标的相关特征分为两部分：(1)生成目标查询的二维边界框$\mathbf{b}_v^i$；(2)其他视图中与$\mathbf{b}_v^i$对应的边界框。

在本文中，我们开发了一种高效的方法，用于为给定的查询关联其他视角中的边界框

。如图4所示，我们为每个感兴趣区域创建一个三维网格$\mathbf{G} \in \mathbb{R}^{W^{roi} \times H^{roi} \times D \times 4}$。我们将网格中的每个点记为$\mathbf{g}_{x,\ y,\ z} = (x \times d_z,\ y \times d_z,\ d_z,\ 1)$，其中$(x,\ y)$是感兴趣区域中的坐标，$d_z \in \{d_0, d_1, \ldots, d_{D-1}\}$是一组预定义的深度值。感兴趣区域$\mathbf{b}_v^i$的网格被转换到第$w$个视角的坐标系中：

$$\mathbf{g}_{v\to w;x,y,z}^{i} = \mathbf{K}_w \mathbf{T}_{v\to w} (\mathbf{K}_v^i)^{-1} \mathbf{g}_{x,y,z}, \tag{9}$$

其中，$\mathbf{g}_{v\to w;x,y,z}^{i}$是第$w$个视图中的变换后网格点，而$\mathbf{T}_{v\to w}$是从第$v$个视图到第$w$个视图的坐标变换矩阵。随后，变换后网格$\mathbf{G}_{v\to w}^{i}$的最小包围框$\mathbf{b}_{v\to w}^{i}$可作为指引，用于在其他视图中为$\mathbf{b}_v^i$寻找相关前景区域。本文考虑两种规则——“Top1 IoU”和“所有重叠区域”——从其他视图中为$\mathbf{b}_v^i$选择相关前景区域$\mathcal{R}_v^i$。

Top1 IoU：选择与$\mathbf{b}_{v\to w}^{i}$的交并比（IoU）最高（若大于 0）的框，作为$\mathbf{b}_v^i$的相关区域：

$$\mathcal{R}_v^i = \left\{ \underset{\mathbf{b}_w^j \in \mathbf{B}_w}{\arg\max}\, \mathrm{IoU}(\mathbf{b}_{v\to w}^{i}, \mathbf{b}_w^j) | v \neq w \right\}. \tag{10}$$

所有重叠区域：选择所有与$\mathbf{b}_{v\to w}^{i}$重叠的检测框，作为相关框：

$$\mathcal{R}_v^i = \left\{ \mathbf{b}_w^j | \mathrm{IoU}(\mathbf{b}_{v\to w}^{i}, \mathbf{b}_w^j) > 0, v \neq w \right\}. \tag{11}$$

随后，将落在$\mathbf{b}_v^i$和$\mathcal{R}_v^i$区域内的所有图像特征视为相关特征，并用于更新目标查询$\mathbf{q}_v^i$。

3.4. 使用稀疏交叉注意力的解码器

生成的对象查询通过类似 DETR 的 Transformer解码器与其相关特征进行交互 [3, 26]。遵循 PETR 的做法，为相关特征提供三维位置嵌入 [26]。如图所示，我们的解码器在稀疏交叉注意力层上有所不同2。MV2D 不使用完整的多视角特征图来构建所有查询共享的键和值，而是仅使用相关特征，为每个目标查询分别构建其自身的键和值。此设计不仅形成了一组紧凑的键和值，还能避免目标查询受到背景噪声和干扰项的影响。最后，我们将由多层感知机构成的分类头和回归头应用于更新后的目标查询，以预测最终的三维目标检测结果。请注意，在三维目标检测中，预测的目标中心$\hat{\mathbf{b}}_{center}$由以下部分组成：$\mathbf{p}^{ref}$以及查询预测的偏移量$\hat{\mathbf{b}}^{offset}$，即$\mathbf{b}_{center} = \mathbf{p}_{ref} + \mathbf{b}_{offset}$。

3.5. 损失函数

在我们的实现中，MV2D 中的二维检测器和三维检测器采用共享的骨干网络权重进行联合训练，供两个检测器共用。二维目标检测损失$\mathcal{L}_{2d}$直接沿用二维检测器中的损失。对于三维目标检测损失，我们遵循以往工作 [41, 26]，使用匈牙利算法 [19] 进行标签分配。分类采用焦点损失 [25]，而 L_1 损失则用于边界框回归。三维目标检测损失可概括如下：

$$\mathcal{L}_{3d} = \lambda_{cls3d} \cdot \mathcal{L}_{cls3d} + \mathcal{L}_{reg3d}. \tag{12}$$

MV2D 的整体损失函数为：

$$\mathcal{L} = \mathcal{L}_{2d} + \lambda_{3d} \cdot \mathcal{L}_{3d}, \tag{13}$$

其中，λ_{3d}是用于平衡二维检测和三维检测损失的权重项，在我们的实验中设为0.1。

3.6. 多帧输入

将所有输入图像（无论是否来自同一时间戳）视为具有不同外参矩阵的相机视角后，MV2D 无需修改流程即可处理多帧输入。目标查询不仅会与来自同一时间戳的相关目标特征交互，也会与来自其他时间戳的相关目标特征交互。相关目标特征选择与第3.3节所述相同。

4. 实验

4.1. 数据集与指标

我们在大规模 nuScenes 数据集上验证了 MV2D 的有效性 [2]。nuScenes 包含 1000 个驾驶序列，分别划分为 700 个训练样本、150 个验证样本和 150 个测试样本。每个序列约持续 20 秒，其中包含来自采样关键帧的 10 个类别的标注三维边界框。每个样本由6 张环视图像组成，具有 1600×900 分辨率，总计提供360° 水平视场。我们将测试集结果提交至在线服务器进行官方评估。其他实验均基于验证集。

4.2. 实现细节

训练与评估。 所有模型均使用 AdamW [29] 优化器进行训练，权重衰减为 0.01。采用余弦退火策略 [28]，初始学习率设为 2×10^{-4}。由于 MV2D 根据二维检测结果生成目标查询，我们在 nuImages [2]上预训练二维检测器，以提供合适的初始化。由于 nuScenes 数据集不提供 2D 边界框标注，我们按照 [42]根据三维边界框生成框标签。我们采用图像空间增强（例如 Gridmask [4]，随机翻转、裁剪和调整尺寸）以及

方法	骨干网络	分辨率	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	平均属性误差↓
DETR3D [41]	ResNet-50	1600 × 900	0.373	0.302	0.811	0.282	0.493	0.979	0.212
PETR [26]	ResNet-50	1408 × 512	0.403	0.339	0.748	0.273	0.539	0.907	0.203
PETRv2 [27]	ResNet-50	1600 × 640	0.494	0.398	0.690	0.273	0.467	0.424	0.195
MV2D-S‡	ResNet-50	1408 × 512	0.440	0.398	0.665	0.269	0.507	0.946	0.203
MV2D-T‡	ResNet-50	1600 × 640	**0.546**	**0.459**	**0.613**	**0.265**	**0.388**	**0.385**	**0.179**
FCOS3D† [39]	ResNet-101	1600 × 900	0.415	0.343	0.725	0.263	0.422	1.292	**0.153**
PGD† [40]	ResNet-101	1600 × 900	0.428	0.369	0.683	**0.260**	0.439	1.268	0.185
DETR3D† [41]	ResNet-101	1600 × 900	0.425	0.346	0.773	0.268	0.383	0.842	0.216
BEVFormer-S† [23]	ResNet-101	1600 × 900	0.448	0.375	0.725	0.272	0.391	0.802	0.200
PETR† [26]	ResNet-101	1600 × 900	0.442	0.370	0.711	0.267	0.383	0.865	0.201
BEVFormer [23]†	ResNet-101	1600 × 900	0.517	0.416	0.673	0.274	0.372	0.394	0.198
PolarFormer [17]†	ResNet-101	1600 × 900	0.528	0.432	0.648	0.270	0.348	0.409	0.201
PETRv2† [27]	ResNet-101	1600 × 640	0.524	0.421	0.681	0.267	0.357	0.377	0.186
MV2D-S‡	ResNet-101	1600 × 640	0.470	0.424	0.654	0.267	0.416	0.888	0.200
MV2D-T‡	ResNet-101	1600 × 640	**0.561**	**0.471**	**0.593**	0.262	**0.340**	**0.368**	0.184

表 1. **nuScenes 上的三维目标检测结果验证集**。 †：模型使用 FCOS3D 初始化。 ‡：模型在 nuImages 上预训练。

方法	骨干网络	NDS↑	平均精度均值↑	平均平移误差↓	平均尺度误差↓	平均方向误差↓	平均速度误差↓	平均属性误差↓
FCOS3D† [39]	ResNet-101	0.428	0.358	0.690	0.249	0.452	1.434	0.124
PGD† [40]	ResNet-101	0.448	0.386	0.626	**0.245**	0.451	1.509	0.127
BEVFormer† [23]	ResNet-101	0.535	0.445	0.631	0.257	0.405	0.435	0.143
PolarFormer† [17]	ResNet-101	0.543	0.457	0.612	0.257	0.392	0.467	0.129
PETRv2† [27]	ResNet-101	0.553	0.456	0.601	0.249	0.391	**0.382**	0.123
MV2D-T ‡	ResNet-101	**0.573**	**0.483**	**0.567**	0.249	**0.359**	0.395	**0.116**
BEVDet [16]	Swin-Base	0.488	0.424	0.524	0.242	0.373	0.950	0.148
M2BEV‡ [42]	ResNeXt-101	0.474	0.429	0.583	0.254	0.376	1.053	0.190
DETR3D§ [41]	V2-99	0.479	0.412	0.641	0.255	0.394	0.845	0.133
BEVDet4D[15]	Swin-Base	0.569	0.451	**0.511**	**0.241**	0.386	**0.301**	0.121
BEVFormer§ [23]	V2-99	0.569	0.481	0.582	0.256	0.375	0.378	0.126
PolarFormer§ [17]	V2-99	0.572	0.493	0.556	0.256	0.364	0.440	0.127
PETRv2§ [26]	V2-99	0.582	0.490	0.561	0.243	0.361	0.343	**0.120**
MV2D-T§	V2-99	**0.596**	**0.511**	0.525	0.243	**0.357**	0.357	**0.120**

表2。 **三维检测结果在 nuScenes 测试集**。 †：模型初始化自 FCOS3D。 ‡：模型在 nuImages 上进行了预训练。 §：模型初始化自 DD3D。

训练期间进行 BEV空间增强 [16]。与最先进方法相比，模型训练 72 个 epoch，且不使用 CBGS [46] 。推理期间不使用测试时增强。我们将不使用和使用多帧输入的模型分别记为 MV2D-S 和 MV2D-T。对于 MV2D-T，推理时我们从不同时间戳采样两帧，间隔为 $15T(T \approx 0.083s)$。为公平比较，我们按照 PETRv2 [27]对 MV2D-T 采用去噪训练。我们的实现基于 MMDetection3D [6]。

网络架构。 在我们的实验中，Faster-RCNN [34]用作二维检测器。二维分数阈值和非极大值抑制（NMS）IoU阈值分别设为 0.05 和 0.6。我们使用 ResNet-50、ResNet-101 [14]和 VoVNetV2 [20]作为骨干网络。ResNet-50 和 ResNet-101 在第 3 和第 4 阶段采用可变形卷积 [7]。VoVNetV2 使用在额外数据 [31]上训练的 DD3D检查点进行初始化。我们采用 FPN [24] 中的 P4 阶段来进行动态目标查询生成和相关特征选择。解码器包含 6 个 Transformer 解码器层，后接一个用于分类和回归的 MLP头。

4.3. 与最先进方法的比较

我们将 MV2D 在 nuScenes 验证 集和 测试 集上的性能与最先进方法进行了比较。结果如表 1 和表 2所示。

表1 展示了在 nuScenes 验证集上的比较结果。根据

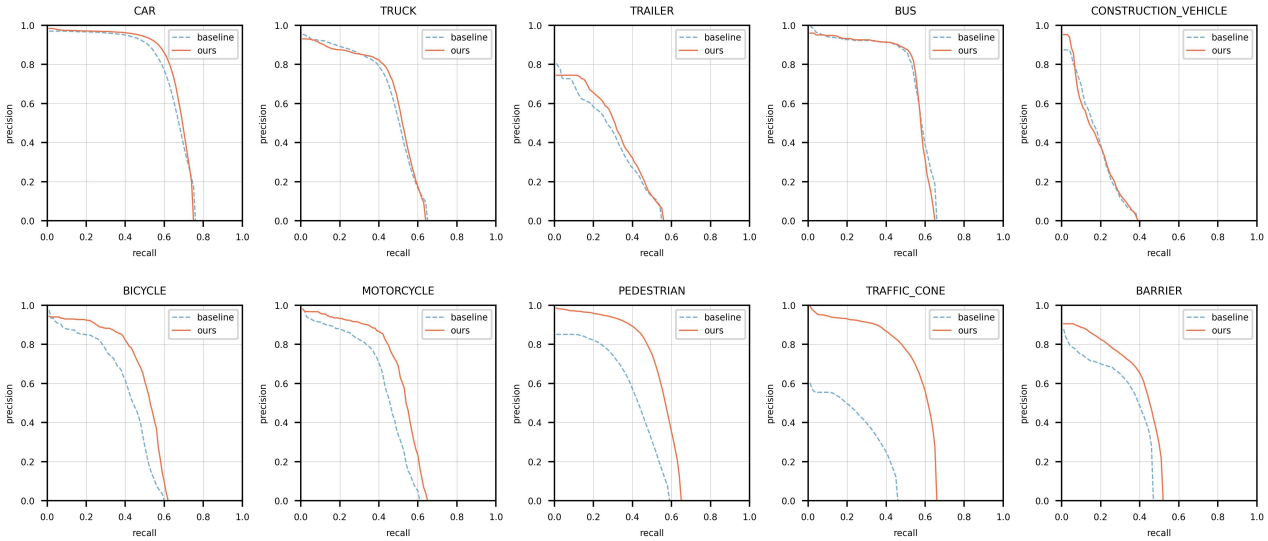


图5. 二维目标检测设置下不同类别的精确率-召回率曲线。模型仍然生成三维目标预测，而每张图像中的二维边界框则通过使用相机参数将三维边界框投影得到。在 nuScenes 验证 集上，使用 0.5 的二维交并比阈值计算精确率和召回率。蓝色虚线表示基线 方法（固定对象查询）。红色实线表示我们的方法（生成的对象查询）。

#	查询	当前键和值	历史键和值	mAP↑	NDS↑
1	固定（900）	all	-	36.2	39.9
2	固定（300）	all	-	36.1	39.4
3	固定 (1500)	all	-	36.6	40.6
4	动态	all	-	38.0	41.2
5	动态	Top1	-	38.2	42.4
6	动态	AO	-	38.4	42.6
7	固定 (900)	all	all	38.6	47.4
8	动态	AO	AO	41.4	51.1

表3。对交叉注意力层中的目标查询和键和值进行消融研究。生成的查询数量按平均值计算。*curr.* 表示当前帧，*hist.* 表示历史帧，*Top1* 表示 Top1 IoU，*AO* 表示全部重叠。

表中结果，MV2D 使用 ResNet-50 和 ResNet-101 时均取得了更高的性能。与多视角三维目标检测方法相比，使用 ResNet-101 的 MV2D-T 达到47.1% 平均精度均值和56.1% NDS，优于基于 BEV 的方法 PolarFormer [17]，高出3.9% 平均精度均值和3.3% NDS。与基于查询的方法相比，MV2D-T 的平均精度均值和 NDS 分别比 PETRv 2 [26] 高出5.0% 和3.7% 。与其他同样使用单帧输入的方法相比，MV2D-S 在平均精度均值上始终表现出优势。平均精度均值的显著提升表明，引入二维检测可以增强三维目标定位能力。

表2 展示了 nuScenes 测试集上的比较结果。如表所示，采用 ResNet-101 的 MV2D 达到了 48.3% 平均精度均值和 57.3% NDS，性能比其他方法高出 2.6%平均精度均值和 2.0% NDS。采用 VoVNetV2 的 MV2D 达到了 51.1% 平均精度均值和 59.6% NDS，性能优于现有方法。

#	对象查询生成	mAP↑	NDS↑
1	均匀采样	37.1	40.1
2	基于尺度的深度	37.3	40.8
3	查询生成器	38.4	42.6

表4。对对象查询生成方式进行消融研究。我们将查询生成器与两种替代方案“均匀采样”和“基于尺度的深度”进行比较。

4.4. 消融研究

本节将详细分析 MV2D 的核心设计选择。所有实验均基于 ResNet-50 骨干网络， 1408×512 输入分辨率。模型训练 24 个 epoch。为公平比较，我们设计了一种基线方法，同样使用二维检测损失进行训练，并以 nuImages 预训练检查点初始化。该方法使用固定数量的可学习目标查询，从整个多视角特征图中聚合信息，与 PETR 中的做法相同 [26]。如表3所示，我们将遵循 PETR、采用 900 个目标查询的设计#1 称为基线。

生成的对象查询与固定对象查询。我们首先比较生成的对象查询与固定对象查询。从#1 到#3 在表3中可以看出，对于固定查询，增加查询数量可以提升性能，因为基于固定查询的方法依赖密集放置的目标查询来定位目标。根据#4，将固定查询替换为动态生成的查询后，平均精度均值和 NDS 分别提升了1.8%和1.3%。这表明，由二维检测器生成的目标查询能够产生质量更高的目标位置假设，从而有利于三维检测。

分辨率	骨干网络	二维检测器	mAP↑	NDS↑
1408×512	R50	YOLOX	41.3	50.0
1408×512	R50	RetinaNet	41.1	50.9
1408×512	R50	Faster R-CNN	41.4	51.1

表5. 不同二维检测器的实验。

方法	交叉注意力模式	mAP	NDS
MV2D	可变形注意力	39.7	48.4
MV2D	相关目标区域	41.4	51.1

表6. 交叉注意力模式比较。

方法	预训练	mAP	NDS	预训练	mAP	NDS
基线	COCO	37.0	45.6	nuImages	38.6	47.4
MV2D	COCO	39.3	49.8	nuImages	41.4	51.1

表7. 预训练权重对比。

相关特征的有效性。我们还验证了仅使用相关特征进行目标查询更新的有效性。在表3中，#4 表示对每个目标查询使用整个多视角特征图，#5 和#6 则表示如第3.3节所述，对每个目标查询仅使用相关目标特征。结果表明，采用“Top1 IoU”规则选出的相关目标特征进行目标查询更新，可使平均精度均值和 NDS 分别提升0.2% 和1.2% 。“所有重叠”规则还能使平均精度均值和 NDS 分别额外提升0.2% 。实验中，鉴于“所有重叠”规则表现更好，我们基于该规则实现相关目标特征选择。这一结果验证了目标查询能够在指定前景区域内聚合信息的有效性。

历史信息。我们额外采样一帧相机扫描帧，为 MV2D 提供历史信息。如表3所示，在采用多帧输入时，#8 比单帧版本#6 提升了3.0% 平均精度均值和8.5% NDS。为进行公平比较，#8 也使带有历史信息的基线#7 提升了2.8% 平均精度均值和3.7% NDS。

目标查询生成器的设计。除了第 3.2 节所述的设计外，我们还尝试了目标查询生成器的其他方案。一种方案是从 [0.5m, 65m] 中为每个感兴趣区域均匀采样 10 个深度值，称为“均匀采样”。另一种方案是根据图像中的表观目标尺度推导深度值 [8]，称为“基于尺度的深度”。实验采用单帧设置，结果列于表 4。这两种替代方案“均匀采样”和“基于尺度的深度”都假设目标深度值服从较简单的分布，从而削弱了三维目标定位能力。

与可变形注意力的比较。可变形注意力 [47] 也将注意力计算限制在
相关目标特征的局部范围。然而，可变形注意力无法保证其生成的采样点能够覆盖对应的目标区域。我们

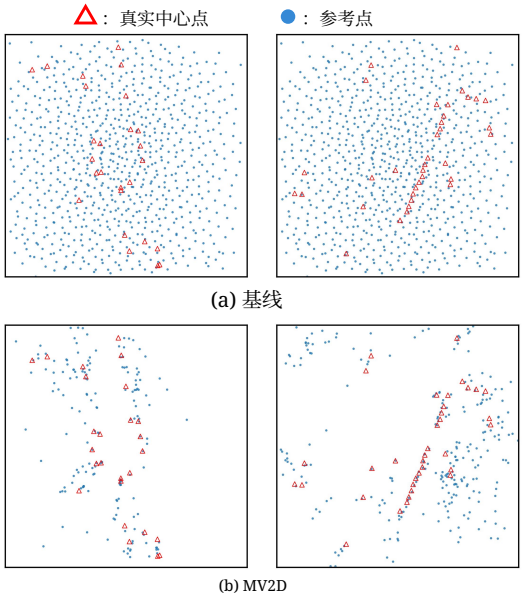


图6. 不同类型目标查询的示意图。上排 (a) 表示基线方法（固定对象查询），下排 (b) 表示我们的方法（生成的对象查询）。我们在鸟瞰空间中展示真实标注中心位置和参考点。这些样本取自 nuScenes 验证集。建议读者放大查看此图。

方法明确施加了这一约束，因此取得了更优的结果，如表6所示。

预训练权重消融实验：为评估预训练的影响，我们在表7中给出了对比结果。基线对应表3中的设置 #7。使用不同的预训练权重时，我们的方法均能稳定提升性能。

4.5. 不同二维检测器上的泛化能力

我们使用不同的二维检测器进行实验，以验证 MV2D 的泛化能力。我们选取了三种二维检测器，包括两阶段检测器 Faster R-CNN、单阶段基于锚框的检测器 RetinaNet [25]以及单阶段无锚框检测器 YOLOX [10]。所有模型均训练 24 个周期。详细的实验设置见补充材料。

如表5所示，使用更强的目标检测器 Faster R-CNN 时，得益于更高的目标候选框质量，MV2D 能够取得更好的性能。即使搭配 RetinaNet 和 YOLOX 等轻量级目标检测器，MV2D 也表现不俗，展现了其泛化能力。

4.6 定性分析

本节分析二维检测器如何提升多视角三维目标检测的性能。为了验证借助图像语义，二维检测器生成的目标查询是否能更准确地定位目标，我们

通过透视投影比较基线方法（固定对象查询）与我们的方法（生成的对象查询）的二维目标检测性能。具体而言，我们将三维边界框投影到每张图像上，然后计算最小二维边界框作为二维检测结果。我们通过比较预测框投影得到的二维边界框与真实标注来评估性能。我们将真阳性分配的交并比阈值设为 0.5。

在图5中，我们绘制了 nuScenes 中 10 个类别的精确率-召回率曲线。从图中可以看出，与基线方法相比，我们的方法取得了更高的精确率和召回率，尤其是在尺寸相对较小的类别上。这表明，二维目标检测器能够充分利用图像语义，从图像空间中发现物体，并为三维检测器提供可利用的物体存在性的有力证据。作为补充，我们在图6中展示了不同类型查询的可视化结果。尽管生成的查询比固定查询稀疏得多，但它们大多分布在物体周围。更多可视化结果和案例研究见补充材料。

5. 结论

本文提出了用于多视图三维目标检测的多视图二维目标引导的三维目标检测器（MV2D）。在我们的框架中，我们将二维目标用作稀疏查询，并采用稀疏交叉注意力模块来约束信息聚合。在实验中，我们展示了所提出的 MV2D框架在 nuScenes 数据集上取得了令人鼓舞的结果。MV2D 可以将任意二维检测器提升为三维检测器；我们相信，利用二维目标作为引导这一思路，能够进一步启发多视角3D目标检测方法的设计。

局限性。 由于 MV2D 根据二维检测生成目标查询，如果二维检测器在所有摄像头视图中都未能检测到某个目标，该目标就可能被遗漏。

致谢

本研究部分得到了国家重点研发计划项目（编号：2022ZD0115502）、国家自然科学基金项目（编号：62122010）以及中央高校基本科研业务费的资助。

参考文献

[1] Garrick Brazil 和 Xiaoming Liu。M3D-RPN：用于目标检测的单目三维区域提议网络。载于 *ICCV*，2019年。1，2[2] Holger Caesar、Varun Bankiti、Alex H. Lang、Sourabh Vora、Venice Erin Liong、Qiang Xu、Anush Krishnan、Yu Pan、Giancarlo Baldan 和 Oscar Beijbom。nuScenes：用于自动驾驶的多模态数据集。载于 *CVPR*，2020年。5，11

[3] Nicolas Carion、Francisco Massa、Gabriel Synnaeve、Nicolas Usunier、Alexander Kirillov 和 Sergey Zagoruyko。基于 Transformer 的端到端目标检测。载于 ECCV，2020年。2，5[4] Pengguang Chen、Shu Liu、Hengshuang Zhao 和 Jiaya Jia。Gridmask 数据增强。arXiv 预印本 arXiv:2001.04086，2020年。5[5] Shaoyu Chen、Xinggang Wang、Tianheng Cheng、Qian Zhang、Chang Huang 和 Wenyu Liu。面向基于视觉的环视三维目标检测的极坐标参数化。arXiv 预印本 arXiv:2206.10965，2022年。1[6] MMDetection3D贡献者。MMDetection3D：OpenMMLab 新一代通用三维目标检测平台。https://github.com/open-mmlab/mmdetection3d，2020年。6[7] Jifeng Dai、Haozhi Qi、Yuwen Xiong、Yi Li、Guodong Zhang、Han Hu 和 Yichen Wei。可变形卷积网络。载于 ICCV，2017年。6[8] Tom van Dijk 和 Guido de Croon。神经网络如何从单幅图像中感知深度？载于 IEEE/CVF国际计算机视觉大会论文集，第2183–2191页，2019年。8[9] Lue Fan、Feng Wang、Naiyan Wang 和 Zhaoxiang Zhang。全稀疏三维目标检测。载于 NeurIPS，2022年。1，2[10] Zheng Ge、Songtao Liu、Feng Wang、Zeming Li 和 Jian Sun。YOLOX：2021年超越 YOLO 系列。arXiv 预印本 arXiv:2107.08430，2021年。8，11[11] Ross Girshick。Fast R-CNN。载于 ICCV，2015年。2[12] Ross Girshick、Jeff Donahue、Trevor Darrell 和 Jitendra Malik。用于精准目标检测和语义分割的丰富特征层次结构。载于 CVPR，2014年。2[13] Kaiming He、Georgia Gkioxari、Piotr Doll´ar 和 Ross Girshick。Mask R-CNN。载于 ICCV，2017年。3[14] Kaiming He、Xiangyu Zhang、Shaoqing Ren 和 Jian Sun。用于图像识别的深度残差学习。载于 CVPR，2016年。6[15] Junjie Huang 和 Guan Huang。BEVDet4D：利用多摄像头三维目标检测中的时序线索。arXiv 预印本 arXiv:2203.17054，2022年。6[16] Junjie Huang、Guan Huang、Zheng Zhu 和 Dalong Du。BEVDet：鸟瞰视角下的高性能多摄像头三维目标检测。arXiv 预印本 arXiv:2112 11790，2021 1，2，6 . .[17] Yanqin Jiang、Li Zhang、Zhenwei Miao、Xiatian Zhu、Jin Gao、Weiming Hu 和 Yu-Gang Jiang。PolarFormer：采用极坐标Transformer的多摄像头三维目标检测。arXiv 预印本 arXiv:2206.15398，2022 年。2，6，7[18] Jason Ku 、 Alex D. Pon 和 Steven L. Waslander。利用精准候选框和形状重建进行单目三维目标检测。载于 CVPR，2019年。1，2[19] Harold W Kuhn。用于指派问题的匈牙利算法。《海军研究后勤季刊》，1955年。5[20] Youngwan Lee、Joong-Won Hwang、Sangrok Lee、Yuseok Bae 和 Jongyoul Park。一种节能且图形处理器计算高效、用于实时目标检测的骨干网络。载于计算机视觉与模式识别会议研讨会，2019年。6

[21] Yanwei Li、Yilun Chen、Xiaojuan Qi、Zeming Li、Jian Sun 和 Jiaya Jia。将基于体素的表示与 Transformer 统一用于三维目标检测。载于 NeurIPS，2022年。1[22] Yinhao Li、Zheng Ge、Guanyi Yu、Jinrong Yang、Zengran Wang、Yukang Shi、Jianjian Sun 和 Zeming Li。BEVDepth：为多视角三维目标检测获取可靠深度。arXiv 预印本 arXiv:2206.10092，2022年。1，2[23] Zhiqi Li、Wenhai Wang、Hongyang Li、Enze Xie、Chonghao Si ma、Tong Lu、Qiao Yu 和 Jifeng Dai。BEVFormer：通过时空变换器从多摄像头图像中学习鸟瞰视角表示。载于 ECCV，2022年。1，2，6[24] Tsung-Yi Lin、Piotr Doll´ar、Ross Girshick、Kaiming He、Bharath Hariharan 和 Serge Belongie。用于目标检测的特征金字塔网络。载于 CVPR，2017年。2，6[25] Tsung-Yi Lin、Priya Goyal、Ross Girshick、Kaiming He 和 Piotr Doll´ar。用于密集目标检测的焦点损失。载于 ICCV，2017年。2，5，8，11[26] Yingfei Liu、Tiancai Wang、Xiangyu Zhang 和 Jian Sun。PETR：用于多视图三维目标检测的位置嵌入变换。载于 ECCV，2022年。1，3，5，6，7[27] Yingfei Liu、Junjie Yan、Fan Jia、Shuailin Li、Qi Gao、Tian- cai Wang、Xiangyu Zhang 和 Jian Sun。PETRv2：用于多摄像头图像三维感知的统一框架。arXiv 预印本 arXiv:2206.01256，2022年。3，6[28] Ilya Loshchilov 和 Frank Hutter。SGDR：带热重启的随机梯度下降。载于 ICLR，2017年。5[29] Ilya Loshchilov 和 Frank Hutter。解耦权重衰减正则化。载于 ICLR，2019年。5[30] Arsalan Mousavian、Dragomir Anguelov、John Flynn 和 Jana Kosecka。利用深度学习和几何进行三维边界框估计。载于 CVPR，2017年。1[31] Dennis Park、Rares Ambrus、Vitor Guizilini、Jie Li 和 Adrien Gaidon。单目三维目标检测是否需要伪激光雷达？载于 ICCV，2021年。2，6[32] Jonah Philion 和 Sanja Fidler。Lift, Splat, Shoot：通过隐式反投影至三维，对来自任意相机组的图像进行编码。载于 ECCV，2020年。2[33] Joseph Redmon、Santosh Divvala、Ross Girshick 和 Ali Farhadi。You Only Look Once：统一的实时目标检测。载于 CVPR，2016年。2[34] Shaoqing Ren、Kaiming He、Ross Girshick 和 Jian Sun。Faster R-CNN：通过区域候选网络实现实时目标检测。载于 NeurIPS，2015年。2，3，6，11[35] Danila Rukhovich、Anna Vorontsova 和 Anton Konushin。ImVoxelNet：用于单目和多视角通用三维目标检测的图像到体素投影。载于 WACV，2022年。1，2[36] Andrea Simonelli、Samuel Rota Bul`o、Lorenzo Porzi、Manuel Lopez-Antequera 和 Peter Kontschieder。解耦单目三维目标检测。载于 ICCV，2019年。1，2[37] Zhi Tian、Chunhua Shen、Hao Chen 和 Tong He。FCOS：全卷积单阶段目标检测。载于 ICCV，2019年。2[38] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszko-reit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。《Attention is all you need》。发表于 NeurIPS，2017年。3, 4[39] Tai Wang、Xinge Zhu、Jiangmiao Pang 和 Dahua Lin。FCOS3D：全卷积单阶段单目三维目标检测。发表于 ICCVW，2021年。1, 2, 6[40] Tai Wang、Xinge Zhu、Jiangmiao Pang 和 Dahua Lin。概率与几何深度：在透视场景中检测目标。发表于 CoRL，2021年。1, 2, 6[41] Yue Wang、Vitor Guizilini、Tianyuan Zhang、Yilun Wang、Hang Zhao 和 Justin Solomon。DETR3D：通过三维到二维查询从多视图图像进行三维目标检测。发表于 CoRL，2021年。1, 2, 3, 5, 6[42] Enze Xie、Zhiding Yu、Daquan Zhou、Jonah Philion、Anima Anandkumar、Sanja Fidler、Ping Luo 和 Jose M Alvarez。M^2BEV：采用统一鸟瞰图表示的多摄像头联合三维检测与分割。arXiv 预印本 arXiv:2204.05088，2022年。5, 6[43] Zhuyu Yao、Jiangbo Ai、Boxun Li 和 Chi Zhang。高效 DETR：利用密集先验改进端到端目标检测器。arXiv 预印本 arXiv:2104.01318，2021年。2[44] Brady Zhou 和 Philipp Kr¨ahenb¨uhl。用于实时地图视图语义分割的跨视图变换器。发表于 CVPR，2022年。3[45] Xingyi Zhou、Dequan Wang 和 Philipp Kr¨ahenb¨uhl。以点为目标。CoRR，abs/1904.07850，2019年。1[46] Benjin Zhu、Zhengkai Jiang、Xiangxin Zhou、Zeming Li 和 Gang Yu。面向点云 3D 目标检测的类别平衡分组与采样。arXiv 预印本 arXiv: . . 3 09492，2019 1908 6[47] Xizhou Zhu、Weijie Su、Lewei Lu、Bin Li、Xiaogang Wang 和 Jifeng Dai。Deformable DETR：用于端到端目标检测的可变形 Transformer。发表于 ICLR，2021年。2, 8

A. 使用更多架构进行实验

本节将通过不同的二维检测器和特征提取器进行更多实验。MV2D 的二维检测器部分在 nuImages 上进行预训练 [2], , 随后在 nuScenes 上联合训练二维检测器部分和三维检测器部分训练集。我们在 nuScenes 上评估三维目标检测性能和模型延迟验证集 [2]。所有模型均训练 24 个周期，且不使用 CBGS。对于模型延迟，我们仅考虑网络前向传播的延迟，不计入预处理和后处理耗时（例如图像加载和格式转换）。延迟是在单张 NVIDIA RTX 3090 图形处理器上以批量大小 1 测得。下面将详细介绍模型架构。

A.1. 模型架构

二维检测器 不失一般性，我们选择了 3 种二维检测器，包括 Faster R-CNN [34], 、单阶段基于锚框的二维检测器 RetinaNet [25]以及单阶段无锚框二维检测器 YOLOX [10]。

特征金字塔 对于采用 ResNet-50 骨干网络和 Faster R-CNN 检测器的模型，我们构建特征金字塔，以生成下采样步幅为$\{4, 8, 16, 32, 64\}$的特征图。对于采用 ResNet-50 骨干网络和 RetinaNet/YOLOX 检测器的模型，我们构建特征金字塔，以生成下采样步幅为$\{8, 16, 32, 64, 128\}$的特征图。

解码器层 MV2D 中的解码器默认包含 6 个解码器层。我们还对不同数量的解码器层进行了实验。

A.2. 性能比较

我们为 MV2D 配备了不同的二维检测器和特征提取器，并在 nuScenes 验证集上评估其延迟和性能。结果列于表 8。如结果所示，在 1408×512 输入分辨率下，采用 Faster R-CNN 作为二维检测器的 MV2D 性能最高，平均精度均值为 41.4%、NDS 为 51.1%，推理延迟为 380ms。采用 RetinaNet 作为二维检测器时，性能略有下降，平均精度均值为 41.1%、NDS 为 50.9%。采用速度更快的单阶段检测器 YOLOX 时，推理延迟降至 246ms，同时仍保持较好的性能，平均精度均值为 41.3%、NDS 为 50.0%。这些结果表明，MV2D 能够适配不同的二维检测器，即使是非常轻量的二维检测器也能发挥作用。在较小的输入分辨率 800×320 下，采用 YOLOX 的 MV2D 平均精度均值为 37.7% 、NDS 为 47.0%，延迟则降至 115ms。这些实验表明，MV2D 能够很好地泛化至其他架构。

在图 7中，我们评估了不同解码器层数下的性能。MV2D 基于 Faster R-CNN 作为二维检测器，采用 1408×512 输入分辨率；基线方法则基于固定对象查询，并采用相同的输入分辨率。使用 1 个解码器层时，MV2D 达到 36.4% 的平均精度均值和 44.8% 的 NDS。使用 2 个解码器层时，平均精度均值和 NDS 分别提升 2.7% 和 3.9%。随着解码器层数增加，平均精度均值和 NDS 也随之提升。可以看出，在平均精度均值和 NDS 两项指标上，使用 2 个解码器层的 MV2D 优于使用 6 个解码器层的基线方法。

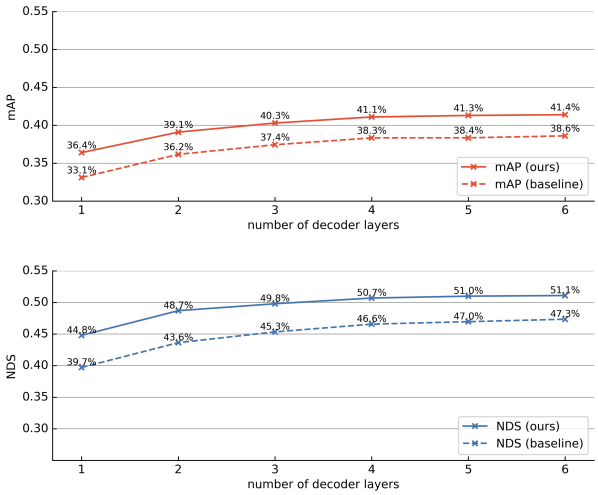


图7. 不同解码器层的比较。

B. 更多可视化

B.1. 定性结果

我们将 MV2D 与基线方法（使用固定目标查询并从所有图像区域收集信息）进行比较，并展示定性结果。三维物体检测结果如图8所示。第1至第3行取自不同的数据样本。第1至第3行分别表示真实标注、MV2D 检测结果和基线检测结果。基线方法可能无法检测到某些物体（三维空间中第1行的自行车），或无法准确定位某些物体（三维空间中第2行的卡车和第3行的人）。然而，二维检测器能够在图像空间中检测到其中大多数物体。因此，二维检测能够为物体的存在性和位置提供丰富的证据。利用这些信息，MV2D 能够生成更准确的三维检测结果。

B.2. 失败案例分析

我们还分析了 MV2D 的失败案例。图中展示了一些示例9。从第 1 行到第 2 行，MV2D 有时会将一个“卡车”目标拆分成一个“卡车”目标和一个“拖车”目标。从第 3 行到第 4 行，如果目标被严重遮挡，二维检测器可能无法成功检测到它们，从而导致 MV2D 出现假阴性。从第 5 行到第 6 行，如果光照条件极端或运动幅度较大

分辨率	骨干网络	二维检测器	延迟	mAP	NDS	mATE	mASE	mAOE	mAVE	mAAE
800×320	ResNet-50	YOLOX	115毫秒	0.377	0.470	0.737	0.280	0.532	0.427	0.213
1408×512	ResNet-50	YOLOX	246毫秒	0.413	0.500	0.697	0.271	0.496	0.402	0.203
1408×512	ResNet-50	RetinaNet	368毫秒	0.411	0.509	0.696	0.272	0.418	0.387	0.185
1408×512	ResNet-50	Faster R-CNN	380毫秒	0.414	0.511	0.694	0.272	0.427	0.396	0.172

表 8. nuScenes验证集上的性能比较 et.



图8。一些定性结果。

出现模糊，二维检测器也可能无法检测到某些目标，从而影响 MV2D 的性能。



图9. 一些失败案例的示意图。