BEVFormer: 利用时空 Transformer 从多相机图像中学习鸟瞰图表示

李志琦 [1,2*]，**王文海** [2*]，**李弘扬** [2*]，**谢恩泽** [3]，**司马崇昊** [2]，
路通 [1]，**乔宇** [2]，**代季峰** [2✉]

[1] 南京大学　[2] 上海人工智能实验室　[3] 香港大学

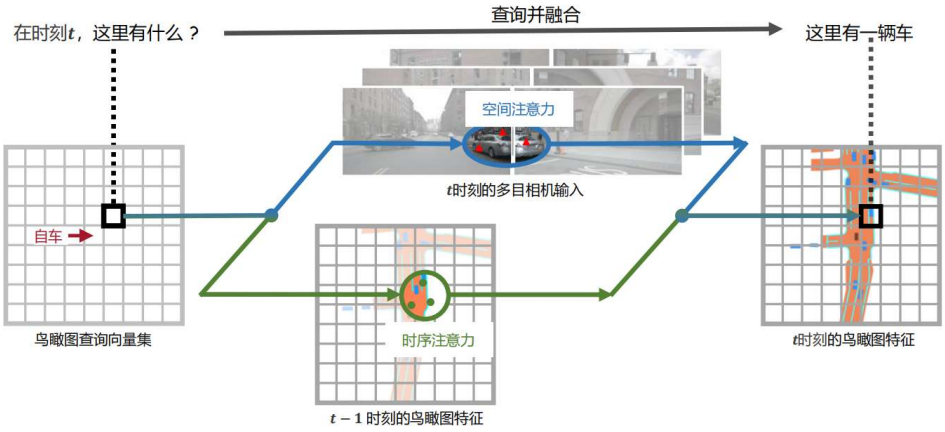

图 1: 我们提出了 **BEVFormer**，这是一种自动驾驶的范式，它应用 Transformer 和时态结构，从多摄像头输入中生成鸟瞰图 (BEV) 特征。BEVFormer 利用询问向量来查找空间/时间域，并相应地聚合时空信息，因此有利于更强的感知任务表征。

摘要

三维视觉感知任务，包括基于多摄像头图像的三维检测和地图分割，是自动驾驶系统的关键。在本研究中，我们提出了一个名为 BEVFormer 的新框架，该框架学习了具有时空 Transformer 的统一 BEV 表征，以支持多个自动驾驶感知任务。简而言之，BEVFormer 利用空间和时间信息，通过预定的网格状 BEV 查询向量与空间和时间域交互。为了聚合空间信息，我们设计了一个空间交叉注意力，每个 BEV 查询向量从跨相机视图的感兴趣区域提取空间特征。对于时间信息，我们提出了一种时间自注意力来递归融合历史 BEV 信息。我们的方法在 nuScenes 测试集上的 NDS 指标达到了最新的 56.9%，比之前的最佳技术高出 9.0 分，与基于 lidar 的基线性能相当。我们进一步表明，BEVFormer 显著提高了低能见度条件下目标速度估计和召回率的准确性。目前代码已开源在 https://github.com/zhiqi-li/BEVFormer。

*: 对等贡献. 这项工作是李志琦在上海人工智能实验室实习时完成的。

✉: 通讯作者。

1 引言

三维空间的感知对于自动驾驶、机器人等各种应用都至关重要。尽管基于 lidar 的方法取得了显著进展 [43, 20, 54, 50, 8]，但基于相机的方法 [45, 32, 47, 30] 近年来引起了广泛关注。除了部署成本低之外，与基于 lidar 的同类方法相比，相机在检测远距离物体和识别基于视觉的道路元素（如交通灯、停车线）方面拥有理想的优势。

自动驾驶中对周围场景的视觉感知，有望从多个摄像头给出的二维线索中预测出 3D 检测框或语义图。最直接的解决方案是基于单目相机框架 [45, 44, 31, 35, 3] 和跨相机后处理。该框架的缺点是它单独处理不同的视图，不能跨相机捕获信息，导致性能和效率低下 [32, 47]。

作为单目相机框架的替代方案，一种更统一的框架是从多目相机图像中提取整体表示。鸟瞰图 (bird’ s eye-view,，BEV) 是一种常用的周围场景表示方法，它能清晰地呈现物体的位置和规模，适用于各种自动驾驶任务，如感知和规划 [29]。尽管之前的地图分割方法证明了 BEV 的有效性 [32, 18, 29]，但基于 BEV 的方法在 3D 目标检测中并没有显示出明显优于其他范式的优势 [47, 31, 34]。其根本原因是，3D 目标检测任务需要强大的 BEV 特征来支持精确的 3D 检测框预测，而从 2D 平面生成 BEV 是不适定的。生成 BEV 特征的流行 BEV 框架基于深度信息 [46, 32, 34]，但这种范式对深度值或深度分布的准确性非常敏感。因此，基于 BEV 的方法的检测性能受到复合误差的影响 [47]，不准确的 BEV 特征会严重影响最终性能。因此，*我们有动力设计一种不依赖深度信息、自适应学习 BEV 特征而不是严格依赖 3D 先验的 BEV 生成方法*。Transformer 使用注意力机制动态地聚合有价值的特性，在概念上满足了我们的需求。

使用 BEV 特征执行感知任务的另一个动机是，BEV 是连接时间和空间域的理想桥梁。对于人类视觉感知系统而言，时间信息在推断物体运动状态和识别遮挡物体方面起着至关重要的作用，视觉领域的许多工作已经证明了利用视频数据的有效性 [2, 27, 26, 33, 19]。然而，现有的多目相机三维检测方法很少利用时间信息。然而，现有最先进的多目相机三维检测方法很少利用时间信息。重要的挑战是，自动驾驶是时间关键的，场景中的物体变化很快，因此简单地叠加跨时间戳的 BEV 特征会带来额外的计算成本和干扰信息，这可能不是理想的。受循环神经网络 (RNN) [17, 10]，*我们利用 BEV 特征将时间信息从过去递归到现在，这与 RNN 模型的隐藏状态具有相同的精神*。

为此，我们提出了一种基于 Transformer 的鸟瞰视角 (BEV) 编码器，称为 **BEVFormer**，它可以有效地聚合环视相机的时空特征和历史 BEV 特征。由 BEVFormer 生成的 BEV 特征可以同时支持三维物体检测和地图分割等多个三维感知任务，对自动驾驶系统具有重要价值。如图1所示，我们的 BEVFormer 包含三个关键设计:(1) 网格型 BEV 查询向量集，通过注意力灵活融合时空特征;(2) 空间交叉注意力模块，从多摄像头图像中聚合空间特征;(3) 时间自注意力模块，从历史 BEV 特征中提取时间信息，有利于运动目标速度估计和重遮挡目标检测。同时带来可以忽略不计的计算开销。利用 BEVFormer 生成的统一特征，该模型可以与不同任务的特定头如可变形 DETR [56] 和掩码解码器 [22] 协作，进行端到端三维物体检测和地图分割。

我们的主要贡献如下:

• 我们提出 BEVFormer，一种基于 Transformer 的编码器，将多目相机和/或时间戳输入投射到 BEV 表示。通过统一的 BEV 特征，我们的模型可以同时支持多种自动驾驶感知任务，包括 3D 检测和地图分割。

• 我们设计了可学习的 BEV 查询向量，并设计了空间交叉注意力层和时间自注意力层，分别从跨摄像头和历史 BEV 中查找空间特征和时间特征，并将其聚合为统一的 BEV 特征。

• 我们在多个具有挑战性的基准上评估拟议的 BEVFormer，包括 [4] 和 Waymo [40]。与现有技术相比，我们的 BEVFormer 始终如一地实现了性能的提高。例如，在可比的参数和计算开销下，BEVFormer 在 nuScenes 测试集上实现了 56.9% 的 NDS，比之前最好的检测方法 DETR3D [47] 高出了 9.0 个百分点 (56.9% *vs.* 47.9%)。对于地图分割任务，我们也实现了最先进的性能，在最具挑战性的车道分割上比 Lift-Splat [32] 高出 5.0 个百分点以上。我们希望这个直观而强大的框架可以作为后续 3D 感知任务的新基线。

2 相关工作

2.1 基于 Transformer 的二维感知

近年来，一种新的趋势是利用 Transformer 来重新构造检测和分割任务 [7, 56, 22]。

DETR [7] 使用一组对象查询向量，通过交叉注意力解码器直接生成检测结果。然而，DETR 的主要缺点是训练时间长。可变形的 DETR [56] 通过提出可变形的注意力来解决这个问题。与传统的全局注意力算法不同，变形注意力算法与局部兴趣区域相互作用，只在每个参考点附近采样 K 个点，计算注意力结果，效率高，训练时间明显缩短。变形注意力的计算公式为:

$$\text{DeformAttn}(q,p,x)=\sum_{i=1}^{N_{\text{head}}}\mathcal{W}_i\sum_{j=1}^{N_{\text{key}}}\mathcal{A}_{ij}\cdot\mathcal{W}_i'x(p+\Delta p_{ij}), \tag{1}$$

其中 q, p, x 分别表示查询向量特征、参考点特征和输入特征。i 表示注意头的索引，N_{head} 表示注意头的总数。j 对采样的键进行索引，N_{key} 为每个头部采样的键总数。$W_i\in\mathbb{R}^{C\times(C/H_{\text{head}})}$ 和 $W_i'\in\mathbb{R}^{(C/H_{\text{head}})\times C}$ 为可学习权重，其中 C 为特征维数。$A_{ij}\in[0,1]$ 为预测注意权重，用 $\sum_{j=1}^{N_{\text{key}}}A_{ij}=1$ 归一化。$\Delta p_{ij}\in\mathbb{R}^2$ 是参考点 p 的预测偏移量.。$x(p+\Delta p_{ij})$ 表示位置 $p+\Delta p_{ij}$ 的特征，采用双线性插值提取，如 Dai [12] 等。在这项工作中，我们将可变形注意力扩展到三维感知任务，以有效地聚合空间和时间信息。

2.2 基于相机的三维感知

以往的三维感知方法通常独立完成三维物体检测或地图分割任务。对于三维物体检测任务，早期的方法类似于二维检测方法 [1, 28, 49, 39, 53]，通常是在二维检测框的基础上预测出三维检测框。Wang 等人 [45] 采用先进的二维检测器 FCOS [41] 直接

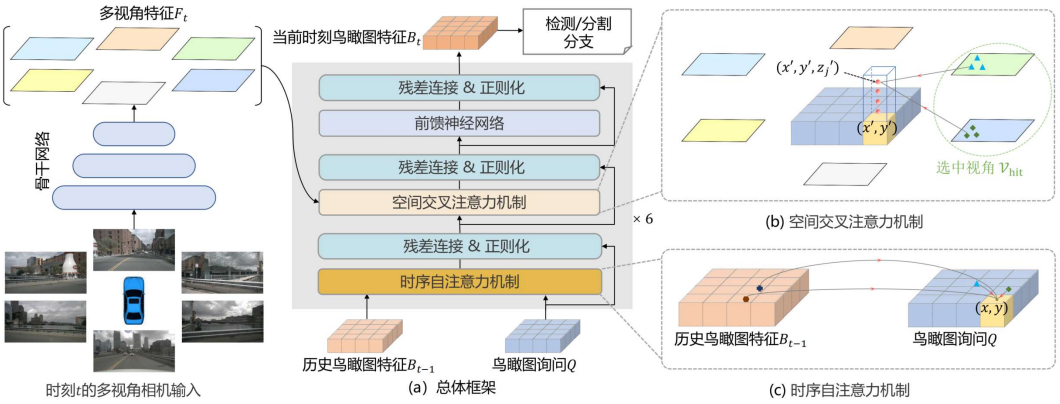

图 2: **BEVFormer 的总体架构**。 (a)BEVFormer 的编码器层包含网格型 BEV 查询向量、时间自注意力和空间交叉注意力。(b) 在空间交叉注意力，每个 BEV 查询向量只与感兴趣区域的图像特征交互。(c) 在时间自注意力中，每个 BEV 查询向量与两个特性交互: 当前时间戳的 BEV 查询和前一个时间戳的 BEV 特性。

预测每个对象的三维检测框。DETR3D [47] 在二维图像中投影可学习的三维查询向量，然后对对应特征进行采样，实现端到端三维检测框预测，无需 NMS 后处理。另一种解决方案是将图像特征转换为 BEV 特征，并从自上而下的视角预测三维检测框。该方法利用深度估计 [46] 或分类深度分布 [34] 得到的深度信息，将图像特征转化为 BEV 特征。OFT [36] 和 ImVoxelNet [37] 将预定义的体素投影到图像特征上，生成场景的体素表示。最近，M^{2}BEV [48] 进一步探索了 BEV 特征同时执行多个感知任务的可能性。

实际上，在地图分割任务中，多目相机特征生成 BEV 特征的研究更为广泛 [32, 30]。一种简单的方法是通过反向透视映射 (IPM) 将透视视图转换为 BEV [35, 5]。此外，Lift-Splat [32] 根据深度分布生成 BEV 特征。方法 [30, 16, 9] 利用多层感知器学习从透视图到 BEV 的转换。PYVA [51] 提出了一种将前视单目图像转换为 BEV 的交叉视角转换器，但由于全局注意力 [42] 的计算量较大，不适合融合多摄像头特征。除了空间信息，以往的研究 [18, 38, 6] 也通过叠加几个时间戳的 BEV 特性考虑了时间信息。BEV 特性限制了固定时间内的可用时间信息，增加了计算量。本文提出的时空 Transformer 同时考虑了时空线索，生成了当前时间的 BEV 特征，而时间信息则是通过 RNN 方式从之前的 BEV 特征中获取的，其计算成本较小。

3 BEVFormer

将多目相机图像特征转换为鸟瞰 (BEV) 特征，可以为各种自动驾驶感知任务提供统一的周边环境表示。在这项工作中，我们提出了一个新的基于 Transformer 的 BEV 生成框架，该框架可以通过注意力有效地聚合来自多视角相机的时空特征和历史 BEV 特征。

3.1 总体架构

如图2所示, BEVFormer 有 6 个编码器层，除 BEV 查询、空间交叉注意力和时间自我注意力三种定制设计外，每个编码器层都遵循转换器 [42] 的常规结构。其中，BEV 查询是一种网格形的可学习参数，通过注意力从多目相机视图中查询 BEV 空间中的特征。空间交叉注意力和时间自注意力是配合 BEV 查询向量的注意层，用于根据 BEV 查询从多相机图像中查找和聚合空间特征和历史 BEV 的时间特征。

在推断过程中，我们在时间戳 t 向骨干网, (如 ResNet-101 [15]) 输入多摄像头图像，得到不同摄像头视图的特征 $F_t = \{F_t^i\}_{i=1}^{N_{\text{view}}}$ 其中 F_t^i 是第 i 个视图的特征, N_{view} 为摄像头视图的总数。同时，我们保留了之前时间戳 $t-1$ 的 BEV 特征 B_{t-1} 。在每个编码器层，我们首先使用 BEV 查询 Q 通过时间自注意力从先验 BEV 特征 B_{t-1} 查询时间信息然后通过空间交叉注意力，利用 BEV 查询向量 Q 从多摄像机特征 F_t 中查询空间信息。前馈网络 [42] 后，编码器层输出细化的 BEV 特征，这是下一编码器层的输入。A 经过 6 层叠加编码器，生成统一的当前时间戳 t 的 BEV 特征 B_t 三维检测头和地图分割头以 BEV 特征 B_t 为输入，预测三维检测框和语义图等感知结果。

3.2 BEV 查询向量

我们预定义了一组网格形状的可学习参数 $Q \in \mathbb{R}^{H \times W \times C}$ 作为 BEVFormer 的查询向量，其中 H, W 为 BEV 平面的空间形状。其中，位于 Q 的 $p = (x, y)$ 查询 $Q_p \in \mathbb{R}^{1 \times C}$ 负责 BEV 平面中对应的网格单元区域。BEV 平面中的每个网格单元对应于真实世界的 s 米大小。BEV 特征的中心默认对应自我车的位置。遵循常用实践 [14], 我们在将 BEV 查询向量 Q 输入到 BEVFormer 之前添加可学习的位置嵌入到 BEV 查询向量 Q 中。

3.3 空间交叉注意力

由于多摄像头三维感知 (包含 N_{view} 摄像头视图) 的输入规模较大，普通的多头关注 [42] 的计算成本极高。因此，我们基于可变形注意力 [56], 开发了空间交叉注意力，这是一个资源高效的注意层，每个 BEV 查询向量 Q_p 只与其跨摄像机视图的兴趣区域交互。但是，可变形注意力原本是为二维感知而设计的，所以对于三维场景需要进行一些调整。

如图2 (b), 我们首先将 BEV 平面上的每个查询向量提升到一个柱状查询向量 [20], 从柱状查询向量中采样 N_{ref} 3D 参考点，然后将这些点投影到二维视图中。对于一个 BEV 查询，投影的二维点只能落在一些视图上，而其他视图不会被命中。在这里，我们将命中视图称为 $\mathcal{V}_{\text{hit}}$. 之后，我们将这些二维点作为查询 Q_p 的参考点，并围绕这些参考点从命中视图 $\mathcal{V}_{\text{hit}}$ 中抽取特征。最后，我们对采样特征进行加权和，作为空间交叉注意力的输出。空间交叉注力意 (SCA) 过程可以表述为:

$$\text{SCA}(Q_p, F_t) = \frac{1}{|\mathcal{V}_{\text{hit}}|} \sum_{i \in \mathcal{V}_{\text{hit}}} \sum_{j=1}^{N_{\text{ref}}} \text{DeformAttn}(Q_p, \mathcal{P}(p, i, j), F_t^i), \tag{2}$$

其中 i 索引摄像机视图，j 索引参考点，N_{ref} 是每次 BEV 查询的总参考点。F_t^i 是第 i 个摄像头视图的特征。对于每一个 BEV 查向量询 Q_p, 我们使用一个投影函数 $\mathcal{P}(p,i,j)$ 来得到第 i 个视图图像上的第 j 个参考点。

接下来，我们介绍如何从投影函数 $\mathcal{P}$ 中获得视图图像上的参考点。我们首先计算出 Q 作为等式 3的位于 $p=(x,y)$ 的查询向量 Q_p 对应的真实世界位置 (x',y')

$$x'=(x-\frac{W}{2})\times s;\quad y'=(y-\frac{H}{2})\times s, \tag{3}$$

其中 H, W 为 BEV 查询向量的空间形状，s 为 BEV 网格的分辨率大小，(x',y') 为自我车位置为原点的坐标。在三维空间中，位于 (x',y') 的物体会出现在 z 轴的高度 z' 处。因此，我们预定义了一组锚点高度 $\{z'_j\}_{j=1}^{N_{\text{ref}}}$ 以确保我们可以捕获出现在不同高度的线索。这样，对于每一个查询向量 Q_p，我们得到一个由 3D 参考点 $(x',y',z'_j)_{j=1}^{N_{\text{ref}}}$ 组成的柱子。最后，我们通过相机的投影矩阵将 3D 参考点投影到不同的图像视图上，可以写成:

$$\begin{gathered}\mathcal{P}(p,i,j)=(x_{ij},y_{ij})\\ \text{where } z_{ij}\cdot\begin{bmatrix}x_{ij} & y_{ij} & 1\end{bmatrix}^T=T_i\cdot\begin{bmatrix}x' & y' & z'_j & 1\end{bmatrix}^T.\end{gathered} \tag{4}$$

这里，$\mathcal{P}(p,i,j)$ 是第 j 个 3D 点 (x',y',z'_j) 投影到第 i 个视图上的 2D 点 $T_i\in\mathbb{R}^{3\times4}$ 是第 i 个相机的已知投影矩阵。

3.4 时间自注意力

除了空间信息外，时间信息对于视觉系统了解周围环境 [27] 也至关重要。例如，在没有时间线索的情况下，从静态图像中推断移动物体的速度或检测高度遮挡的物体是具有挑战性的。为了解决这个问题，我们设计了时间自注意力，它可以通过结合历史 BEV 特征来表示当前环境。

给定当前时间戳 t 的 BEV 查询向量 Q 和保存在时间戳 $t-1$ 的历史 BEV 特征 B_{t-1} 我们首先根据自我运动将 B_{t-1} 与 Q 对齐，使同一网格上的特征对应于相同的现实位置。在这里，我们表示对齐的历史 BEV 特征 B_{t-1} 为 B'_{t-1}。然而，从时间 $t-1$ 到 t, 可移动的物体在现实世界中以各种偏移量移动如何在不同时间的 BEV 特征之间建立相同对象的精确关联是一个难题。因此，我们通过时间自注意力 (temporal self-attention, TSA) 层对特征之间的这种时间联系进行建模，可以写成如下形式:

$$\text{TSA}(Q_p,\{Q,B'_{t-1}\})=\sum_{V\in\{Q,B'_{t-1}\}}\text{DeformAttn}(Q_p,p,V), \tag{5}$$

其中 Q_p 表示位于 $p=(x,y)$ 的 BEV 查询向量。此外，与常规可变形注意力不同的是，时间自注意力的偏移量 Δp 由 Q 和 B'_{t-1} 的连接来预测。特别地，对于每个序列的第一个样本，时间自注意力将退化为没有时间信息的自注意力，其中我们将 BEV 特征 $\{Q,B'_{t-1}\}$ 替换为重复的 BEV 查询向量 $\{Q,Q\}$。

与 [18, 38, 6] 中简单叠加 BEV 相比，我们的时间自注意力可以更有效地模拟长时间依赖性。BEVFormer 从之前的 BEV 特征中提取时间信息，而不是从多个叠加的 BEV 特征中提取时间信息，从而减少了计算量和受到的干扰信息。

3.5 BEV 特性的应用

BEV 的特征是 $B_t \in \mathbb{R}^{H \times W \times C}$ 是一种通用的二维特征地图，可用于各种自动驾驶感知任务，因此，只需稍加修改，就可以基于二维感知方法开发三维物体检测和地图分割任务头 [56, 22]。

对于三维物体检测：我们设计了一种基于二维可变形 DETR [56] 探测器的端到端三维探测头。改进方法包括使用单尺度 BEV 特征 B_t 作为解码器的输入，预测三维检测框和速度而不是二维检测框，仅使用 L_1 损失监督三维检测框回归。利用检测头，我们的模型可以对三维检测框和速度进行端到端预测，而无需进行 NMS 后处理。

对与地图分割：设计了一种基于二维分割方法 Panoptic SegFormer [22] 的地图分割头。由于基于 BEV 的地图分割与普通的语义分割基本相同，我们利用 [22] 的掩码译码器和类固定查询向量来针对每个语义类别，包括汽车、车辆、道路 (可行驶区域) 和车道。

3.6 实施细节

训练阶段：对于时间戳 t 的每个样本，我们从过去 2 秒的连续序列中再随机抽取 3 个样本，这种随机抽样策略可以增加自我运动 [57] 的多样性。我们将这四个样本的时间戳表示为 $t-3, t-2, t-1$ 和 t。对于前三个时间戳的样本，它们负责反复生成 BEV 特征 $\{B_{t-3}, B_{t-2}, B_{t-1}\}$ ，这个阶段不需要梯度。对于时间戳 $t-3$ 的第一个样本，之前没有 BEV 特征，时间上的自注意力退化为自注意力。在时刻 t，模型根据多摄像头输入和先验 BEV 特征 B_{t-1} 生成 BEV 特征 B_t 使 B_t 包含了跨越四个样本的时空线索。最后，我们将 BEV 特征 B_t 输入到检测和分割头部并计算相应的损失函数。

推理阶段：在推断阶段，我们评估每一帧的视频序列的时间顺序。前一个时间戳的 BEV 特性被保存并用于下一个时间戳，这种在线推断策略具有时间效率，并且与实际应用一致。虽然我们利用了时间信息，但我们的推理速度仍然可以与其他方法相媲美 [45, 47]。

4 实验

4.1 数据集

我们在两个具有挑战性的公共自动驾驶数据集上进行实验，即 nuScenes 数据集 [4] Waymo 开放数据集 [40]。

NuScenes 数据集 [4] 包含 1000 个场景，每个场景持续时间大约为 20s，关键样本以 2Hz 的频率进行标注。每个样本由 6 个相机的 RGB 图像组成，具有 360° 水平视场。对于检测任务，有来自 10 个类别的 1.4M 带注释的 3D 边框。我们按照 [32] 中的设置执行 BEV 分段任务。该数据集还提供了检测任务的官方评估指标。nuScenes 的平

均精度 (mAP) 是利用地平面上的中心距离计算的，而不是通过 Union (IoU) 上的 3D Intersection (3D Intersection over Union) 来匹配预测结果和地面真实值。nuScenes 度量还包含 5 种类型的真正度量 (TP 度量)，包括 TE、ASE、AOE、VE 和 AAE，分别用于测量平移、尺度、方向、速度和属性错误。NuScenes 还将 nuScenes 检测得分 (NDS) 定义为 $\text{NDS}=\frac{1}{10}[5\text{mAP}+\sum_{\text{mTP}\in\mathbb{TP}}(1-\min(1,\text{mTP}))]$ 以捕获 nuScenes 检测任务的所有方面。

Waymo 公开数据集 [40] 是一个大型自动驾驶数据集，拥有 798 个训练序列和 202 个验证序列。请注意，Waymo 提供的每帧 5 张图片只有大约 252° 的水平视场，但提供的注释标签是围绕自我车的 360°。我们删除了这些在训练集和验证集的任何图像上都不可见的检测框。由于 Waymo 开放数据集是大规模和高速率的 [34]，我们使用训练分割的子集，每隔 5 帧从训练序列中采样，只检测车辆类别。我们使用 3D IoU 的阈值 0.5 和 0.7 在 Waymo 数据集上计算 mAP。

4.2 实验设置

按照之前的方法 [45, 47, 31]，我们采用了两种类型的骨干：ResNet101-DCN [15, 12] that initialized 从 FCOS3D [45] 检查点初始化的 ResNet101-DCN [15, 12] 和从 DD3D [31] 检查点初始化的 VoVnet99 [21]。我们默认使用 FPN [23] 输出的多尺度特征，其大小分别为$1/16$, $1/32$, $1/64$ 维数 $C=256$ 。在 nuScenes 上的实验中，BEV 查向量询的默认大小为 200×200, 感知范围为 [−51.2m, 51.2m] 的 X 轴和 Y 轴，BEV 网格分辨率 s 大小为 0.512m. 我们对 BEV 查询向量采用了可学习的位置嵌入。BEV 编码器包含 6 个编码器层，并不断细化每一层的 BEV 查询向量。每个编码器层的输入 BEV 特性 B_{t-1} 相同，不需要梯度。对于每一个局部查询向量，在变形注意力实现的空间交叉注意力模块中，对应于三维空间中 $N_{\text{ref}}=4$ 个不同高度的目标点，预定义的高度锚点在 -5 米到 3 米范围内均匀采样。对于 2D 视图特征上的每个参考点，我们在每个头部参考点周围使用四个采样点。默认情况下，我们用 24 个 epoch 训练模型，学习率为 2×10^{-4}。

对于 Waymo 上的实验，我们改变了一些设置。由于 Waymo 的摄像系统无法捕捉到自我车 [40] 周围的整个场景，BEV 查询向量的默认空间形状为 300×220, 感知范围为 X 轴 [−35.0m, 75.0m]，Y 轴 [−75.0m, 75.0m]。每个网格分辨率 s 的大小为 0.5m。这款自负型汽车是 BEV 的 (70,150) 倍。

基线： 为了消除任务头的影响，公平地比较其他 BEV 生成方法，我们使用 VPN [30] 和 Lift-Splat [32] 替换我们的 BEVFormer，保持任务头和其他设置相同。我们还通过在不使用历史 BEV 特性的情况下将临时自我注意力调整为普通自我注意力，将 BEVFormer 调整为一个静态模型 **BEVFormer-S**。

4.3 三维目标检测结果

我们用检测头训练我们的模型进行检测任务，只为了与之前最先进的三维目标检测方法进行相当的比较。在表1和表2中，我们报告了 nuScenes 测试集和验证集拆分的

表 1: **NuScenes 测试集上的三维检测结果**。 $*$ 注意到 VoVNet-99 (V2-99) [21] 在深度估计任务上预先训练了额外的数据 [31]。“BEVFormer-S” 没有利用 BEV 编码器中的时间信息。“L” 和 “C” 分别表示 LiDAR 和 Camera。

方法	模态	骨干	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
SSN [55]	L	-	0.569	0.463	-	-	-	-	-
CenterPoint-Voxel [52]	L	-	0.655	0.580	-	-	-	-	-
PointPainting [43]	L&C	-	0.581	0.464	0.388	0.271	0.496	0.247	0.111
FCOS3D [45]	C	R101	0.428	0.358	0.690	0.249	0.452	1.434	**0.124**
PGD [44]	C	R101	0.448	0.386	**0.626**	**0.245**	0.451	1.509	0.127
BEVFormer-S	C	R101	0.462	0.409	0.650	0.261	0.439	0.925	0.147
BEVFormer	C	R101	**0.535**	**0.445**	0.631	0.257	**0.405**	**0.435**	0.143
DD3D [31]	C	V2-99*	0.477	0.418	**0.572**	**0.249**	**0.368**	1.014	**0.124**
DETR3D [47]	C	V2-99*	0.479	0.412	0.641	0.255	0.394	0.845	0.133
BEVFormer-S	C	V2-99*	0.495	0.435	0.589	0.254	0.402	0.842	0.131
BEVFormer	C	V2-99*	**0.569**	**0.481**	0.582	0.256	0.375	**0.378**	0.126

表 2: **NuScenes `val` 集的三维检测结果** “C” 表示相机。

方法	模态	骨干	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
FCOS3D [45]	C	R101	0.415	0.343	0.725	0.263	0.422	1.292	**0.153**
PGD [44]	C	R101	0.428	0.369	0.683	**0.260**	0.439	1.268	0.185
DETR3D [47]	C	R101	0.425	0.346	0.773	0.268	0.383	0.842	0.216
BEVFormer-S	C	R101	0.448	0.375	0.725	0.272	0.391	0.802	0.200
BEVFormer	C	R101	**0.517**	**0.416**	**0.673**	0.274	**0.372**	**0.394**	0.198

主要结果。在公平的训练策略和可比较的模型量表下，我们的方法优于以前最好的方法 DETR3D [47] 在 `val` 集上超过 9.2 个百分点 (51.7% NDS *vs.* 42.5% NDS)。在测试集上, 我们的模型在没有花里胡哨的情况下实现了 56.9% NDS，比 DETR3D (47.9% NDS) 高了 9.0 个百分点。我们的方法甚至可以达到与一些基于激光雷达的基线 (SSN (56.9% NDS) [55] 和 PointPainting (58.1% NDS) [43]) 相当的性能。

以往的基于摄像机的方法 [47, 31, 45] 几乎无法估计速度，而我们的方法证明了时间信息在多镜头检测的速度估计中起着至关重要的作用。BEVFormer 在测试集上的平均 V 速度误差 (mA VE) 为 0.378 米 / 秒大大超过其他基于摄像头的方法，接近基于激光雷达的方法 [43] 的性能。

我们还在 Waymo 上进行了实验, 如表3所示。在 [34] 之后, 我们用 0.7 和 0.5 的 IoU 标准来评估车辆类别。此外，我们也采用 nuScenes 指标来评估结果，因为基于 iou 的指标对于基于摄像头的方法太有挑战性了。由于有少数基于相机的作品在 Waymo 上报道

表 3: **在 Waymo 评价指标和 nuScenes 评价指标下对 Waymo** 验证集**进行三维检测结果**。“L1”和 “L2” 是指 Waymo [40]。* 只使用前置摄像头，只考虑前置摄像头视场 (50.4°) 内的物体标签。†: 我们通过设置 ATE 和 AAE 为 1 来计算 NDS 评分。“L” 和 “C” 分别表示 LiDAR 和 Camera。

方法	模态	Waymo 度量				Nuscenes 度量				
		IoU=0.5		IoU=0.7						
		L1/APH	L2/APH	L1/APH	L2/APH	NDS†↑	AP↑	ATE↓	ASE↓	AOE↓
PointPillars [20]	L	0.866	0.801	0.638	0.557	0.685	0.838	0.143	0.132	0.070
DETR3D [47]	C	0.220	0.216	0.055	0.051	0.394	0.388	0.741	**0.156**	0.108
BEVFormer	C	**0.280**	**0.241**	**0.061**	**0.052**	**0.426**	**0.440**	**0.679**	0.157	**0.101**
CaDNN* [34]	C	0.175	0.165	0.050	0.045	-	-	-	-	-
BEVFormer*	C	0.308	0.277	0.077	0.069	-	-	-	-	-

表 4: **在 nuScenes** 验证集**上的三维目标检测和地图分割结果**。训练分割与检测任务是否联合的比较。*: 我们使用 VPN [30] 和 Lift-Splat [32] 替换我们的 BEV 编码器进行比较，任务头是相同的。†: 这是他们论文的结果。

方法	任务头		三维目标检测		BEV 分割 (IoU)			
	Det	Seg	NDS↑	mAP↑	汽车	交通工具	道路	车道线
Lift-Splat† [32]	✗	✓	-	-	32.1	32.1	72.9	20.0
FIERY† [18]	✗	✓	-	-	-	38.2	-	-
VPN* [30]	✓	✗	0.333	0.253	-	-	-	-
VPN*	✗	✓	-	-	31.0	31.8	76.9	19.4
VPN*	✓	✓	0.334	0.257	36.6	37.3	76.0	18.0
Lift-Splat*	✓	✗	0.397	0.348	-	-	-	-
Lift-Splat*	✗	✓	-	-	42.1	41.7	77.7	20.0
Lift-Splat*	✓	✓	0.410	0.344	43.0	42.8	73.9	18.3
BEVFormer-S	✓	✗	0.448	0.375	-	-	-	-
BEVFormer-S	✗	✓	-	-	43.1	43.2	**80.7**	21.3
BEVFormer-S	✓	✓	0.453	0.380	44.3	44.4	77.6	19.8
BEVFormer	✓	✗	0.517	**0.416**	-	-	-	-
BEVFormer	✗	✓	-	-	44.8	44.8	80.1	**25.7**
BEVFormer	✓	✓	**0.520**	0.412	**46.8**	**46.7**	77.5	23.9

结果，我们也使用 DETR3D 官方代码在 Waymo 上进行实验进行比较。我们可以看到，在 IoU 标准为 0.5 的情况下，BEVFormer 在 LEVEL_1 和 LEVEL_2 困难度上的平均精度 (带有航向信息的平均精度, APH) [40] 分别为 6.0% 和 2.5% ，优于 DETR3D。在 nuScenes 指标上，BEV 以 3.2% NDS 和 5.2% AP 领先于 DETR3D。我们还在前置摄像头上进行实验，比较 BEVFormer 和 CaDNN [34]，这是一种单目相机三维目标检

表 5: **采用不同的 BEV 编码器对 nuScenes 验证集进行不同的检测结果**。“内存” 是指训练过程中消耗的 GPU 内存。*: 我们使用 VPN [30] 和 LiftSplat [32] 替换我们模型的 BEV 编码器进行比较。†: 我们在空间交叉注意力中使用全局注意力来训练 BEVFormer-S，模型使用 fp16 权值来训练。此外，我们仅采用骨干中的单尺度特征，并将 BEV 查询向量的空间形状设置为 100×100 以节省内存。‡: 我们仅通过去除预测的偏移量和权重来降低可变形注意的交互目标从局部区域到参考点。

方法	注意力	NDS↑	mAP↑	mATE↓	mAOE↓	#Param.	FLOPs	Memory
VPN* [30]	-	0.334	0.252	0.926	0.598	111.2M	924.5G	~20G
List-Splat* [32]	-	0.397	0.348	0.784	0.537	74.0M	1087.7G	~20G
BEVFormer-S†	Global	0.404	0.325	0.837	0.442	62.1M	1245.1G	~36G
BEVFormer-S‡	Points	0.423	0.351	0.753	0.442	68.1M	1264.3G	~20G
BEVFormer-S	Local	**0.448**	**0.375**	**0.725**	**0.391**	68.7M	1303.5G	~20G

测方法，在 Waymo 数据集上报告了他们的结果。在 IoU 标准为 0.5 的 LEVEL_1 和 LEVE_2 难度上，BEVFormer 优于 CaDNN, APH 分别为 13.3% 和 11.2%。

4.4 多任务感知结果

我们同时使用检测头和分割头对模型进行训练，以验证模型对多任务的学习能力，结果如表4所示。比较相同设置下的不同 BEV 编码器时，除了道路分割结果与 BEVFormer- s 比较外，BEVFormer 在所有任务中都实现了更高的性能。例如，通过联合训练，BEVFormer 比 Lift-Splat* [32] 在闪避任务上高出 11.0 个百分点 (52.0% NDS *v.s.* 41.0% NDS) 在车道分割上高出 5.6 个百分点 (23.9% *v.s.* 18.3%)。与单独训练任务相比，多任务学习通过共享更多的模块来节省计算成本和减少推理时间，包括骨干和 BEV 编码器。在本文中，我们证明了由 BEV 编码器生成的 BEV 特征能够很好地适应不同的任务，并且使用多任务头进行模型训练在检测任务和车辆分割上表现更好。然而，联合训练的模型在道路和车道分割方面不如单独训练的模型，这是多任务学习中常见 负迁移现象 [11, 13]。

4.5 消融实验

为了深入研究不同模块的影响，我们对带探测头的 nuScenes val 装置进行了消融实验。更多消融研究见附录。

空间交叉注意力的有效性：为了验证空间交叉注意力的效果，我们使用 BEVFormer-S 进行消融实验，排除时间信息的干扰，结果如表5所示。默认的空间交叉注意力是基于可变形注意力的。为了进行比较，我们还构建了两种不同注意力的基线:(1) 用全局注意力替代可变形注意力;(2) 每次查询向量只与自己的参考点交互，而不是与周围的局部区域交互，类似于之前的方法 [36, 37]. 为了更广泛的比较，我们还将 BEVFormer 替换为 VPN [30] Lift-Spalt [32] 提出的 BEV 生成方法。我们可以观察到，在可比的模型规模下，可变形注意力明显优于其他注意力。全局注意力消耗过多的 GPU 内存，点交互

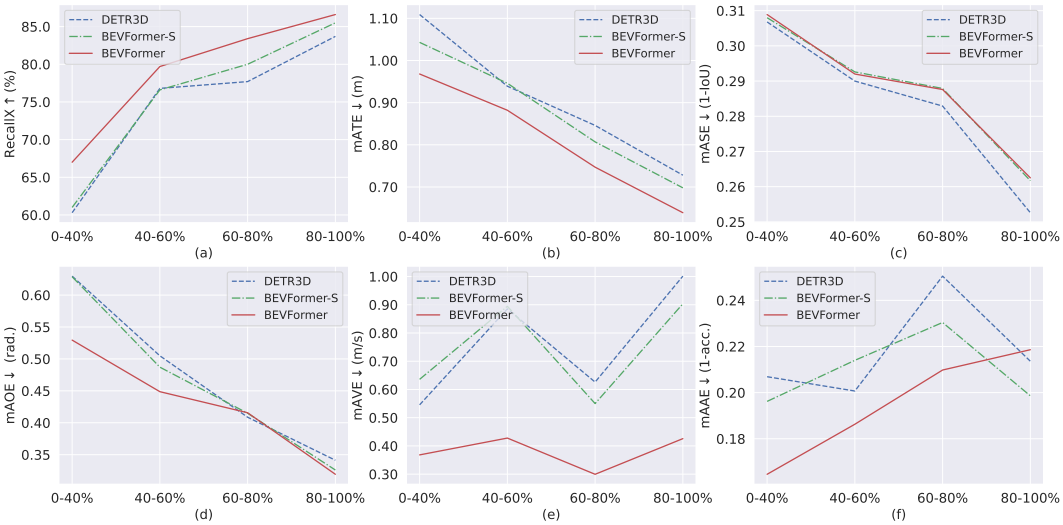

图 3: **不同可见性子集的检测结果**。我们根据对象的可见性 {0-40%, 40-60%, 60-80%, 80-100%} 将 nuScenes `val` 集划分为四个子集。(a): 通过时间信息的增强，BEVFormer 在所有子集上都有较高的召回率，尤其是在能见度最低的子集上 (0-40%)。(b)、(d) 和 (e): 时间信息有利于翻译、定向和速度精度。(c) 和 (f): 不同方法之间的尺度和属性误差差距最小。时间信息对对象尺度预测没有帮助。

的感受野有限。稀疏注意力实现了更好的性能，因为它与先验确定的感兴趣区域交互，平衡了感受野和 GPU 消耗。

时间自注意力的有效性: 从表1和表4中我们可以看到，在相同的设置下，BEVFormer 的性能要优于 BEVFormer-s，有显著的提高，尤其是在挑战性的检测任务上。时间信息的影响主要体现在以下几个方面:(1) 时间信息的引入大大提高了速度估计的准确性;(2) 利用时间信息预测的目标位置和方向更准确;(3) 由于时间信息中包含了过去物体的线索，我们对严重遮挡的物体获得了更高的回忆率，如图3所示。T 为了评估 BEVFormer 在具有不同遮挡级别的物体上的性能，我们根据 nuScenes 提供的官方可见性标签将 nuScenes 的验证集划分为四个子集。在每个子集中，我们在匹配时以 2 米为中心距离阈值，计算所有类别的平均召回率。所有方法的最大预测框数为 300，以便公平地比较召回率。在只有 0-40% 的对象可见的子集上，BEVFormer 的平均召回率优于 BEVFormer-S 和 DETR3D，余量超过 6.0%。

模型规模和延迟: 我们在表6中比较了不同配置的性能和延迟。我们从是否使用多尺度视图特性、BEV 查询的形状和层数三个方面对 BEVFormer 的尺度进行了削弱，以验证性能和推断延迟之间的权衡。我们可以观察到，在 BEVFormer 中使用一个编码器层的配置 C 实现了 50.1% 的 NDS，并将 BEVFormer 的延迟从原来的 130ms 减少到 25ms。配置 D，具有单尺度视图特征，较小的 BEV 大小，只有一个编码器层，在推断期间只消耗 7ms，尽管它比默认配置损失 3.9 个点。但由于环视图像输入，限制效率的瓶颈在于骨干，高效的自动驾驶骨干值得深入研究。总的来说，我们的架构可以适应各种模型规模，并灵活地权衡性能和效率。

表 6: **在 nuScenes `val` 集上不同模型配置的延迟和性能**。延迟是在 V100 GPU 上测量的，骨干是 R101-DCN。输入图像形状为 900×1600。“MS” 注释了多尺度视图功能。

方法	BEVFormer 规模			延迟 (ms)			FPS	NDS↑	mAP↑
	MS	BEV	# 层	骨干	BEVFormer	头			
BEVFormer	✓	200×200	6	391	130	19	1.7	**0.517**	**0.416**
A	✗	200×200	6	387	87	19	1.9	0.511	0.406
B	✓	100×100	6	391	53	18	2.0	0.504	0.402
C	✓	200×200	1	391	25	19	2.1	0.501	0.396
D	✗	100×100	1	387	**7**	18	**2.3**	0.478	0.374

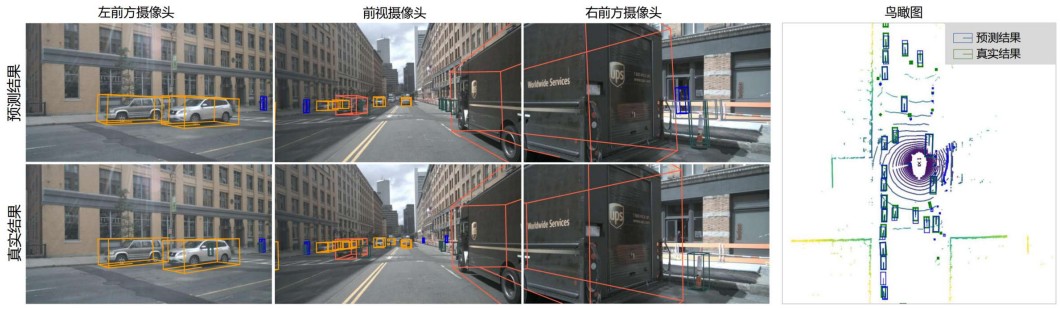

图 4: **BEVFormer 在 nuScenes 验证集上的可视化结果**。我们展示了在多目相机图像和鸟瞰图中的三维检测框预测。

4.6 可视化结果

我们将复杂场景的检测结果如图4所示。BEVFormer 产生了令人印象深刻的结果，除了在小型和远程对象中有一些错误。更多的定性结果在附录中提供。

5 讨论和结论

在这项工作中，我们提出了 BEVFormer 从多镜头输入生成鸟瞰图特征。BEVFormer 可以高效聚合时空信息，生成强大的 BEV 特征，同时支持三维检测和地图分割任务。

局限性：目前，基于摄像机的方法在效果和效率上仍与基于激光雷达的方法有一定差距。对基于摄像机的方法来说，从二维信息准确推断出三维位置仍然是一个长期的挑战。

更广泛的影响： BEVFormer 实验表明，利用多镜头输入的时空信息可以显著提高视觉感知模型的性能。BEVFormer 所展示的优势，如更准确的速度估计和对低可见物体更高的召回率，对构建更好、更安全的自动驾驶系统至关重要。我们认为 BEVFormer 只是以下更强大的视觉感知方法的基础，基于视觉的感知系统仍有巨大的潜力有待开发。

参考文献

[1] Brazil, G., Liu, X.: M3d-rpn: Monocular 3d region proposal network for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9287–9296 (2019)

[2] Brazil, G., Pons-Moll, G., Liu, X., Schiele, B.: Kinematic 3d object detection in monocular video. In: European Conference on Computer Vision. pp. 135–152. Springer (2020)

[3] Bruls, T., Porav, H., Kunze, L., Newman, P.: The right (angled) perspective: Improving the understanding of road scenes using boosted inverse perspective mapping. In: 2019 IEEE Intelligent Vehicles Symposium (IV). pp. 302–309. IEEE (2019)

[4] Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)

[5] Can, Y.B., Liniger, A., Paudel, D.P., Van Gool, L.: Structured bird's-eye-view traffic scene understanding from onboard images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15661–15670 (2021)

[6] Can, Y.B., Liniger, A., Unal, O., Paudel, D., Van Gool, L.: Understanding bird's-eye view semantic hd-maps using an onboard monocular camera. arXiv preprint arXiv:2012.03040 (2020)

[7] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)

[8] Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1907–1915 (2017)

[9] Chitta, K., Prakash, A., Geiger, A.: Neat: Neural attention fields for end-to-end autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15793–15803 (2021)

[10] Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)

[11] Crawshaw, M.: Multi-task learning with deep neural networks: A survey. arXiv preprint arXiv:2009.09796 (2020)

[12] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)

[13] Fifty, C., Amid, E., Zhao, Z., Yu, T., Anil, R., Finn, C.: Efficiently identifying task groupings for multi-task learning. Advances in Neural Information Processing Systems **34** (2021)

[14] Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: International Conference on Machine Learning. pp. 1243–1252. PMLR (2017)

[15] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[16] Hendy, N., Sloan, C., Tian, F., Duan, P., Charchut, N., Xie, Y., Wang, C., Philbin, J.: Fishing net: Future inference of semantic heatmaps in grids. arXiv preprint arXiv:2006.09917 (2020)

[17] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)

[18] Hu, A., Murez, Z., Mohan, N., Dudas, S., Hawke, J., Badrinarayanan, V., Cipolla, R., Kendall, A.: Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15273–15282 (2021)

[19] Kang, K., Ouyang, W., Li, H., Wang, X.: Object detection from video tubelets with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 817–825 (2016)

[20] Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019)

[21] Lee, Y., Hwang, J.w., Lee, S., Bae, Y., Park, J.: An energy and gpu-computation efficient backbone network for real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)

[22] Li, Z., Wang, W., Xie, E., Yu, Z., Anandkumar, A., Alvarez, J.M., Lu, T., Luo, P.: Panoptic segformer: Delving deeper into panoptic segmentation with transformers. arXiv preprint arXiv:2109.03814 (2021)

[23] Lin, T.Y., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 936–944 (2017)

[24] Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv: Learning (2017)

[25] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)

[26] Luo, W., Yang, B., Urtasun, R.: Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 3569–3577 (2018)

[27] Ma, X., Ouyang, W., Simonelli, A., Ricci, E.: 3d object detection from images for autonomous driving: A survey. arXiv preprint arXiv:2202.02980 (2022)

[28] Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3d bounding box estimation using deep learning and geometry. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7074–7082 (2017)

[29] Ng, M.H., Radia, K., Chen, J., Wang, D., Gog, I., Gonzalez, J.E.: Bev-seg: Bird's eye view semantic segmentation using geometry and semantic point cloud. arXiv preprint arXiv:2006.11436 (2020)

[30] Pan, B., Sun, J., Leung, H.Y.T., Andonian, A., Zhou, B.: Cross-view semantic segmentation for sensing surroundings. IEEE Robotics and Automation Letters **5**(3), 4867–4873 (2020)

[31] Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is pseudo-lidar needed for monocular 3d object detection? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3142–3152 (2021)

[32] Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: European Conference on Computer Vision. pp. 194–210. Springer (2020)

[33] Qi, C.R., Zhou, Y., Najibi, M., Sun, P., Vo, K., Deng, B., Anguelov, D.: Offboard 3d object detection from point cloud sequences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6134–6144 (2021)

[34] Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical depth distribution network for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8555–8564 (2021)

[35] Reiher, L., Lampe, B., Eckstein, L.: A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird' s eye view. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). pp. 1–7. IEEE (2020)

[36] Roddick, T., Kendall, A., Cipolla, R.: Orthographic feature transform for monocular 3d object detection. In: BMVC (2019)

[37] Rukhovich, D., Vorontsova, A., Konushin, A.: Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2397–2406 (2022)

[38] Saha, A., Maldonado, O.M., Russell, C., Bowden, R.: Translating images into maps. arXiv preprint arXiv:2110.00966 (2021)

[39] Simonelli, A., Bulo, S.R., Porzi, L., Lopez-Antequera, M., Kontschieder, P.: Disentangling monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)

[40] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020)

[41] Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9627–9636 (2019)

[42] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[43] Vora, S., Lang, A.H., Helou, B., Beijbom, O.: Pointpainting: Sequential fusion for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4604–4612 (2020)

[44] Wang, T., Xinge, Z., Pang, J., Lin, D.: Probabilistic and geometric depth: Detecting objects in perspective. In: Conference on Robot Learning. pp. 1475–1485. PMLR (2022)

[45] Wang, T., Zhu, X., Pang, J., Lin, D.: Fcos3d: Fully convolutional one-stage monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 913–922 (2021)

[46] Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8445–8453 (2019)

[47] Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning. pp. 180–191. PMLR (2022)

[48] Xie, E., Yu, Z., Zhou, D., Philion, J., Anandkumar, A., Fidler, S., Luo, P., Alvarez, J.M.: Mˆ 2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. arXiv preprint arXiv:2204.05088 (2022)

[49] Xu, B., Chen, Z.: Multi-level fusion based 3d object detection from monocular images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2345–2353 (2018)

[50] Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)

[51] Yang, W., Li, Q., Liu, W., Yu, Y., Ma, Y., He, S., Pan, J.: Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15536–15545 (2021)

[52] Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11784–11793 (2021)

[53] Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)

[54] Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4490–4499 (2018)

[55] Zhu, X., Ma, Y., Wang, T., Xu, Y., Shi, J., Lin, D.: Ssn: Shape signature networks for multi-class object detection from point clouds. In: European Conference on Computer Vision. pp. 581–597. Springer (2020)

[56] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (2020)

[57] Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2349–2358 (2017)

Appendix

A 实现细节

在本节中，我们提供了所提出方法的更多实现细节和实验。

A.1 训练策略

按照之前的方法 [47, 56]，我们用 24 个 epoch 训练所有模型，每个 GPU 的批大小为 1(包含 6 个视图图像)，学习率为 2×10^{-4}，骨干的学习率乘数为 0.1，我们用余弦退火 [24] 衰减学习率。我们使用权重衰减为 1×10^{-2} 的 AdamW [25] 来优化我们的模型。

A.2 VPN 和 Lift-Splat

在本工作中，我们使用 [30] 和 Lift-Splat [32] 作为两个基线。为了公平比较，骨干和任务头与 BEVFomer 是相同的。

VPN：我们在这项工作中使用官方代码[1] 。受 MLP 参数量的限制，VPN 很难生成高分辨率的 BEV (如 200× 200)。为了与 VPN 进行比较，我们通过两个视图转换层将单尺度视图特征转换为低分辨率 50×50 的 BEV。

Lift-Splat：我们用两个额外的卷积层增强了 Lift-Splat[2]的摄像机编码器，以便在可比较的参数下与我们的 BEVFormer 进行公平的比较，其他设置保持不变。

A.3 任务头

检测头：我们预测了每个三维检测框的 10 个参数，包括 3 个参数 (l, w, h) 用于每个盒子的尺度，3 个参数 (x_o, y_o, z_o) 用于中心位置，2 个参数 $(\cos(\theta), \sin(\theta))$ 用于物体的偏航 θ，2 个参数 (v_x, v_y) 用于速度。训练阶段只使用 L_1 损耗和 L_1 成本。在 [47] 之后，我们使用 900 个对象查询向量，并在推断期间保持 300 个置信度最高的检测框。

分离头：如图5所示，对于语义映射的每个类，我们遵循 [22] 中的掩码译码器，使用一个可学习的查询向量来表示这个类，并根据常规多头注意力的注意图生成最终的分割掩码。

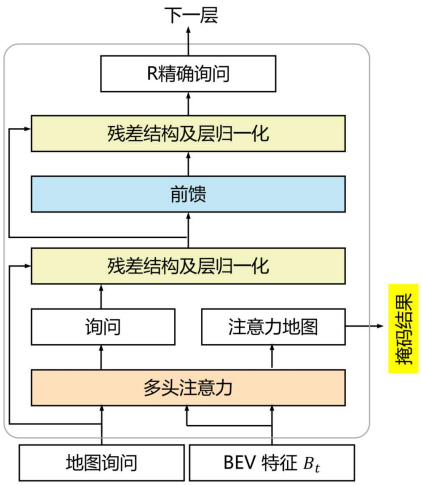

图 5: **BEVFormer 的分割头（掩码解码器）。**

A.4 空间交叉注意力

全局注意力：除了可变形注意力 [56]，我们的空间交叉注意力还可以通过全局注意力 (即常规多头注意力) [42]。使用全局注意力的最直接方法是让每个 BEV 查询向量与所

[1] https://github.com/pbw-Berwin/View-Parsing-Network

[2] https://github.com/nv-tlabs/lift-splat-shoot

有多摄像头功能交互，而且这种概念性实现不需要相机校准。然而，这种简单的方法的计算成本是无法承受的。因此，我们仍然利用相机的内在和外在来决定一个 BEV 查询向量应该交互的命中视图。该策略使得一个 BEV 查询向量通常只与一个或两个视图交互，而不是与所有视图交互，使得在空间交叉注意力中使用全局注意力成为可能。值得注意的是，与其他依赖于相机内在和外在的精确注意机制相比，全局注意力对相机校准更稳健。

B 相机外在的鲁棒性

BEVFormer 利用相机的内在和外在特性来获取二维视图上的参考点。在自动驾驶系统的部署阶段，由于各种原因，如校准误差、相机偏移等，外部因素可能会产生偏差。如图6所示，我们展示了不同相机外部噪声水平下的模型结果。与 BEVFormer-S(点) 相比，BEVFormer-S 利用了参考点周围的可变注意力 [56] 和样本特征的空间交叉注意力，而不是仅仅与参考点相互作用。在可变行注意力的情况下，BEVFormer-S 的鲁棒性比 BEVFormer-S(点) 强。例如，噪声等级为 4 时，BEVFormer-S 的 NDS 下降了 15.2% (按 $1-\frac{0.380}{0.448}$) 计算，而 BEVFormer-S(点) 的 NDS 下降了 17.3%. 与 BEVFormer-s 相比，BEVFormer 只降低了 14.3% 的 NDS，这表明时间信息也可以提高相机外部特征的鲁棒性。在 [32] 之后，我们表明当带有噪声的外部因素训练 BEVFormer 时，BEVFormer(噪声) 具有更强的鲁棒性 (仅下降 8.9% 的 NDS)。基于全局注意力的空间交叉注意力，BEVFormer(全局) 在 4 级相机外部噪声下仍具有较强的抗干扰能力 (NDS 下降 4.0%)。原因是我们不使用相机外在来选择 BEV 查询的 RoIs。

值得注意的是，在最苛刻的噪声下，我们看到 BEVFormer-S(全局) 的表现甚至超过 bevformer (38.8% NDS *vs.* 38.0% NDS)。

C 消融实验

训练中帧数的影响： 表7显示了帧号在训练中的作用。我们看到 nuScenes 验证集上的 NDS 随着帧数的增加而增加，并在帧数 ≥ 4 时开始趋于平稳。因此，我们在实验中默认设置训练时的帧数为 4。

一些设计的效果： 表8显示了几项消融实验的结果。对比 #1 和 #4, 我们发现将历史 BEV 特征与自我运动对齐对于表示与当前 BEV 查询向量相同的几何场景非常重要 (51.0% NDS *vs.* 51.7% NDS)。对比 #2 and #4, 从 5 帧中随机抽取 4 帧是一个有效的数据增强策略，可以提高性能 (51.3% NDS *vs.* 51.7% NDS)。与在时间自我注意力模块中仅使用 BEV 查询向量预测偏移量和权重相比 (见 #3)，同时使用 BEV 查询向量和历史 BEV 特征 (见 #4) 包含更多关于过去 BEV 特征和利益位置预测的线索 (51.3% NDS *vs.* 51.7% NDS)。

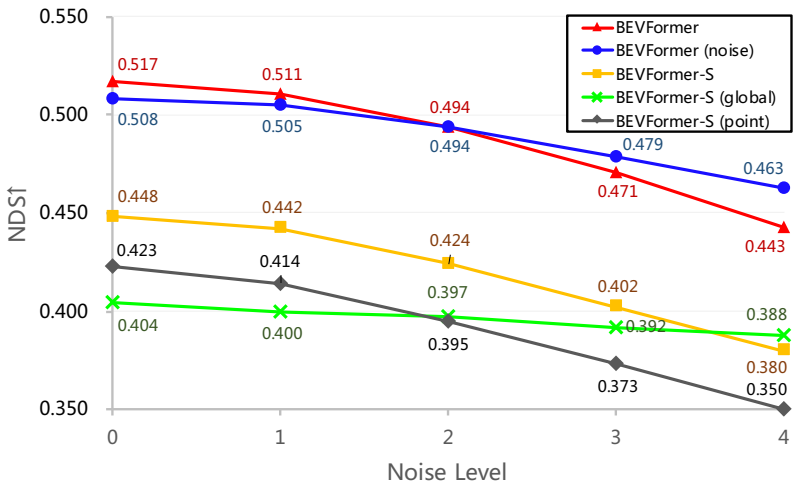

图 6: **在不同程度相机外部噪声作用下的图像处理方法**。对于第 i 级噪声旋转噪声采样自均值为 0、方差为 i 的正态分布 (旋转噪声以度为单位，各轴噪声相互独立)，平移噪声采样自均值为 0、方差为 $5i$ 的正态分布 (平移噪声以厘米为单位，各方向噪声相互独立)。“BEVFormer” 是我们的默认版本。“BEVFormer (噪声)” 使用噪声外部性 (噪声等级 =1) 进行训练。“BEVFormer-S” 是我们的静态版本的 BEVFormer，空间交叉注意由可变注意力 [56] 实现。“BEVFormer-S (全局)” 是 BEVFormer-S 与空间交叉注意力实现的全局注意力 (即常规多头注意力) [42]。“BEVFormer-S (点)” 是具有点空间交叉注意力的 BEVFormer-S，我们将可变形注意力的交互目标从局部区域降低到参考点，仅通过去除预测偏移和权重。

表 7: **在训练过程中使用不同的帧号对 nuScenes 验证集合上的模型进行 NDS 分析**。“#Frame” 表示训练时的帧号。

# 框架	NDS↑	mAP↑	mAVE↓
1	0.448	0.375	0.802
2	0.490	0.388	0.467
3	0.510	0.410	0.423
4	**0.517**	**0.416**	0.394
5	**0.517**	0.412	**0.387**

表 8: **nuScenes 验证的消融实验**。 “A.” 表示将历史 BEV 特征与自我运动相结合。“R.” 表示从 5 个连续帧中随机抽取 4 帧。“B.” 表示同时使用 BEV 查询向量和历史 BEV 特性来预测偏移量和权重。

#	A.	R.	B.	NDS↑	mAP↑
1	✗	✓	✓	0.510	0.410
2	✓	✗	✓	0.513	0.410
3	✓	✓	✗	0.513	0.404
4	✓	✓	✓	**0.517**	**0.416**

D 可视化

如图7所示, 我们比较 BEVFormer 和 BEVFormer- s。利用时间信息，BEVFormer 成功检测出被板卡遮挡的两根总线。我们还在图8中展示了物体检测和地图分割的结果，我们可以看到检测结果和分割结果是高度一致的。我们在图9中提供了更多的地图分割结果，我们看到，通过 BEVFormer 生成的强大的 BEV 特征，通过一个简单的掩码解码器可以很好地预测语义地图。

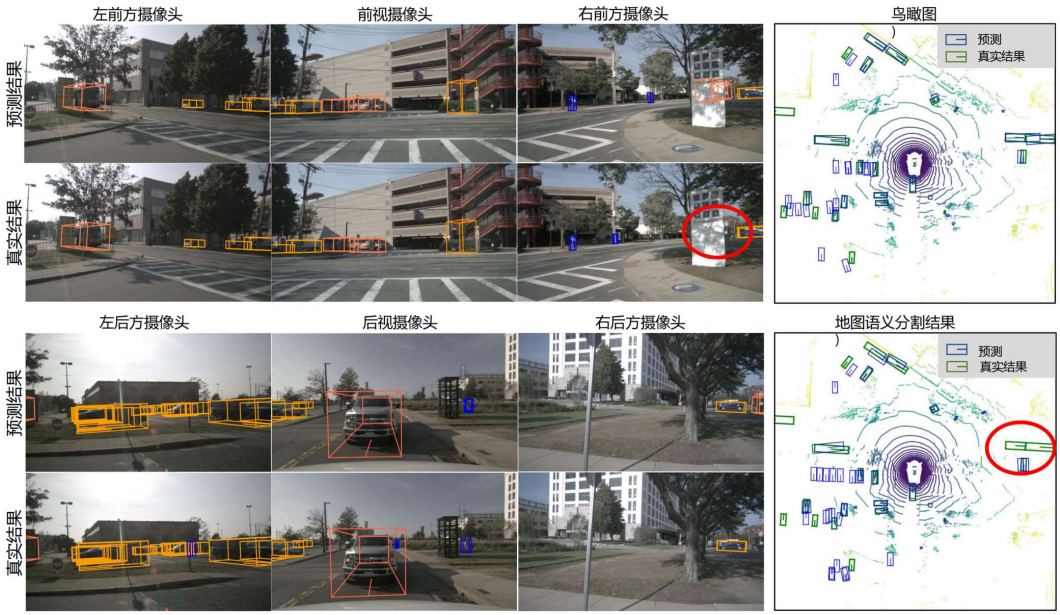

图 7: **比较 BEVFormer 和 BEVFormer- s 的 nuScenes 值集**。我们可以观察到 BEVFormer 可以检测到高度遮挡的物体，而这些物体在 BEVFormer- s 的预测结果中被遗漏了 (红圈)。

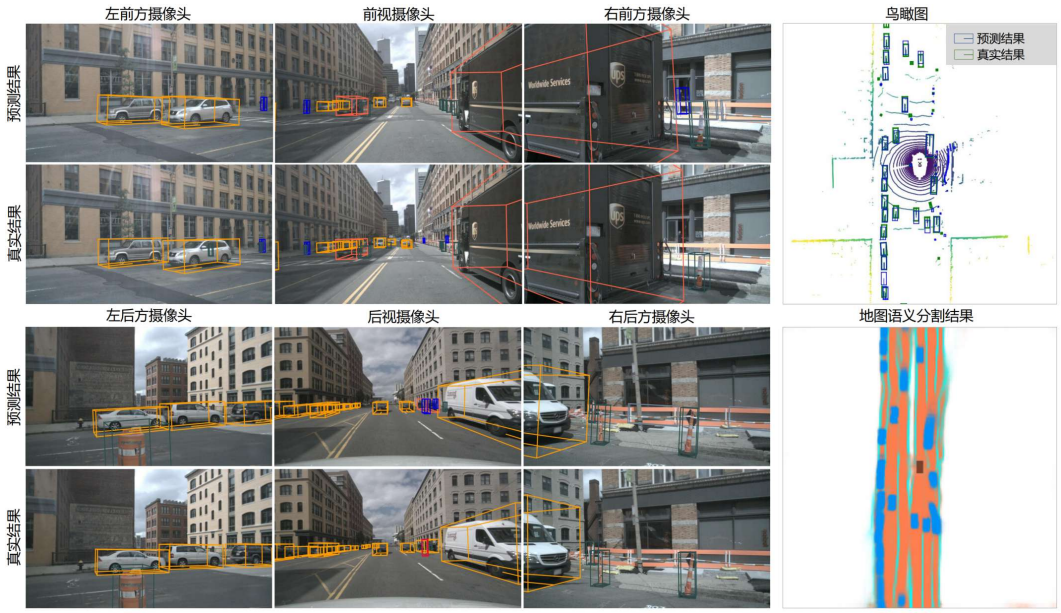

图 8: **对象检测和地图分割任务的可视化结果**。我们分别用蓝色、橙色和绿色表示车辆、道路和车道分割。

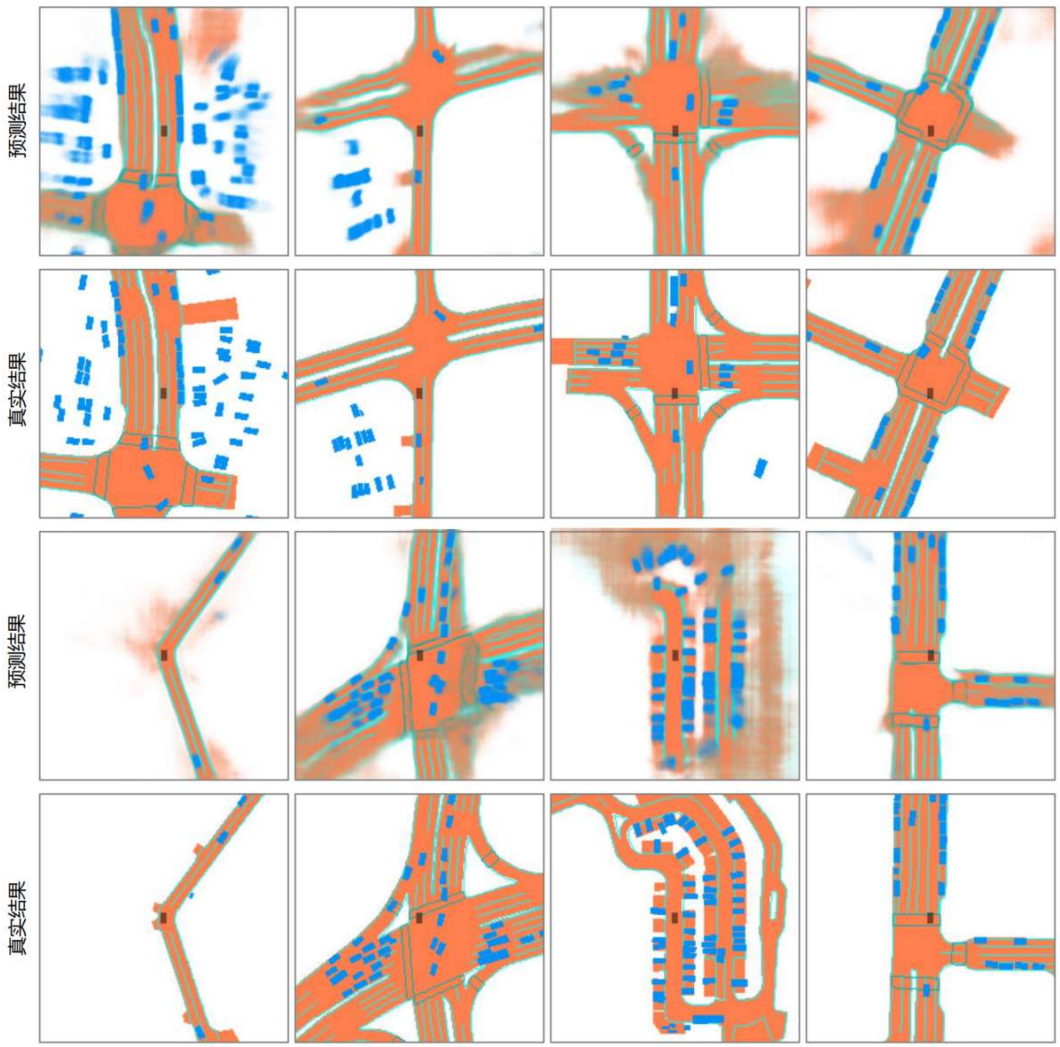

图 9: **地图分割任务的可视化结果**。我们分别用蓝色、橙色、青色和绿色表示车辆、道路、行人过路和车道分割。