

规划导向的自动驾驶

胡一涵^{1,2*}, 杨家志^{1*}, 陈力^{1*†}, 李可宇^{1*}, 司马崇豪¹, 朱西周^{3,1}
 柴思齐², 杜森尧², 林天威², 王文海¹, 陆乐伟³, 贾晓松¹
 刘强², 戴继峰¹, 乔宇¹, 李宏洋^{1†}

1 OpenDriveLab和OpenGVLab, 上海人工智能实验室2
 武汉大学3 商汤科技研究院

* 同等贡献 †项目负责人

<https://github.com/OpenDriveLab/UniAD>

摘要

现代自动驾驶系统以模块化任务按顺序执行的方式为特征，即感知、预测和规划。为了执行多种多样的任务并实现高级智能，当代方法要么为每个任务部署独立的模型，要么设计一个具有独立头部的多任务范式。然而，这些方法可能会遭受累积误差或任务协调不足的问题。相反，我们认为应该设计并优化一个框架，以追求最终目标，即自动驾驶汽车的规划。基于此，我们重新审视了感知和预测中的关键组件，并优先处理这些任务，以使所有任务都为规划做出贡献。我们引入了统一自动驾驶（UniAD），这是一个迄今为止最全面的框架，将全栈驾驶任务整合到一个网络中。它精心设计，以利用每个模块的优势，并从全局角度为代理交互提供互补的特征抽象。任务通过统一的查询接口进行通信，以相互协作实现规划。我们在具有挑战性的nuScenes基准上实例化了UniAD。通过广泛的消融实验，证明了这种理念的有效性，显著优于之前所有方面的最先进技术。代码和模型已公开。

1. 引言

随着深度学习的成功发展，自动驾驶算法集成了包括感知中的检测、跟踪、映射，以及预测中的运动和占用预测等一系列任务¹。如图1(a)所示，大多数行业解决方案部署了标准的

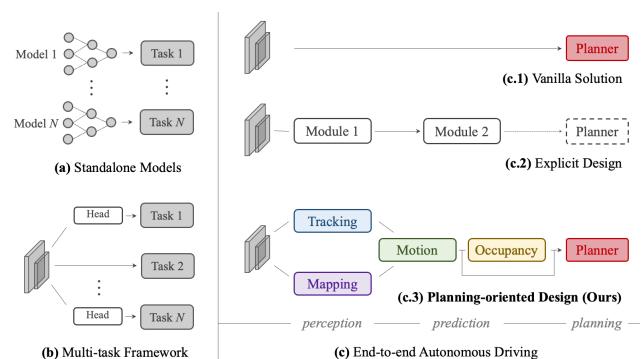


图1. 自主驾驶框架各种设计的比较。(a) 大多数工业解决方案为不同任务部署单独的模型。(b) 多任务学习方案共享一个主干网络，并划分任务头。(c) 端到端范式将感知和预测模块统一。之前的尝试要么直接在(c.1)中对规划进行优化，要么在(c.2)中设计包含部分组件的系统。相反，我们在(c.3)中主张，理想的系统应面向规划，并适当组织前序任务以促进规划。

虽然这种设计简化了跨团队的研发难度，但由于优化目标的孤立，存在模块间信息损失、错误累积和特征不一致的风险。

一个更为优雅的设计是将广泛的²任务整合到多任务学习（MTL）范式中，通过将多个任务特定的头部模块插入到一个共享的特征提取器中，如图1(b)所示。在许多领域，包括通用视觉[46,51,61]、自动驾驶²[8, 34, 57, 59]，如Transfuser[13]和BEV-

¹在以下语境中，我们交替使用任务、模块、组件、单元和节点来表示某一特定任务（例如，检测）。

²在本文中，我们将自动驾驶中的多任务学习（MTL）定义为超越感知任务的范畴。在感知领域内，已有大量关于多任务学习的研究，例如检测、深度、光流等。这类文献不在本文的讨论范围内。

Design	Approach	Perception		Prediction		Plan
		Det.	Track	Map	Motion	
(b)	NMP [57]	✓		✓		✓
	NEAT [12]			✓		✓
	BEVerse [59]	✓		✓	✓	
(c.1)	45 54					✓
(c.2)	PnPNet 32	✓	✓		✓	
	ViP3D 18	✓	✓		✓	
	P3 [47]				✓	✓
	MP3 []			✓	✓	✓
	ST-P3 [23]			✓	✓	✓
(c.3)	LAV []	✓	✓	✓	✓	✓
	UniAD (ours)	✓	✓	✓	✓	✓

表1. 任务对比与分类。“设计”列如图1所示分类。“Det.”表示3D物体检测，“Map”代表在线地图构建，“Occ.”是占用地图预测。
†：这些工作并非直接为规划而提出，但它们仍然体现了联合感知与预测的精神。UniAD执行五项基本驾驶任务以促进规划。

在多任务学习（MTL）中，跨任务的协同训练策略可以利用特征抽象；它可以轻松扩展到其他任务，并节省车载芯片的计算成本。然而，这种方案可能会导致不理想的“负迁移”[16,36]。

相比之下，端到端自动驾驶的兴起[6, 8, 12, 23, 54]将感知、预测和规划的所有节点统一为一个整体。前置任务的选择和优先级应倾向于规划。系统应设计为以规划为导向，精心设计并包含某些组件，从而避免在独立选项中出现的累积误差或在多任务学习方案中出现的负迁移。表1描述了不同框架设计的任务分类。

遵循端到端范式，一种“白板”实践是直接预测规划轨迹，如图1(c.1)所示，无需对感知和预测进行任何显式监督。先驱工作[7,9,14,15,45,53,54,60]在闭环仿真[17]中验证了这种朴素设计。尽管这一方向值得进一步探索，但在安全保证和可解释性方面仍显不足，尤其是在高度动态的城市环境中。本文中，我们倾向于另一种视角，并提出以下问题：面向可靠且以规划为导向的自动驾驶系统，如何设计有利于规划的流程？哪些前置任务是必需的？

一个直观的解决方案是显式地感知周围物体、预测未来行为并规划安全操作，如图1(c.2)所示。当代方法[6,18,23,32,47]提供了良好的见解并取得了显著的性能。然而，我们认为细节中的魔鬼至关重要；以往的工作或多或少未能考虑某些组件（见表1中的区块(c.2)），这让人联想到以规划为导向的精神。我们详细阐述了这一点。

关于详细定义和术语，这些模块在补充材料中的必要性。

为此，我们引入了UniAD，这是一个统一的自动驾驶算法框架，旨在利用五个基本任务，构建一个安全且稳健的系统，如图1(c.3)和表1(c.3)所示。UniAD的设计理念以规划为导向。我们认为，这不仅仅是一个简单的任务堆叠，而是需要深入的工程努力。关键组成部分是基于查询的设计，以连接所有节点。与经典的边界框表示相比，查询得益于更大的感受野，从而减轻了上游预测中的累积误差。此外，查询具有灵活性，能够建模和编码多种交互，例如多个代理之间的关系。据我们所知，UniAD是首次全面研究自动驾驶领域中感知、预测和规划等多任务联合协作的工作。

贡献总结如下。(a) 我们采用了一种以规划为导向的自动驾驶框架新视角，并展示了有效任务协调的必要性，而非独立设计或简单的多任务学习。(b) 我们提出了UniAD，一个综合的端到端系统，利用广泛的任务范围。其关键组件是作为连接所有节点的接口的查询设计。因此，UniAD具有灵活的中间表示和向规划方向交换多任务知识的能力。(c) 我们在现实场景的具有挑战性的基准上实例化了UniAD。通过广泛的消融实验，我们验证了我们的方法在各个方面优于之前的最先进技术。

我们希望这项工作能够为自动驾驶系统的目标驱动设计提供一些启示，为协调各种驾驶任务提供一个起点。

2. 方法论

概述。如图2所示，UniAD包括四个基于transformer解码器的感知和预测模块，以及一个最终的规划器。查询Q在管道中起到连接作用，用于模拟驾驶场景中不同实体的交互。具体来说，一系列多摄像头图像被输入特征提取器，生成的透视视图特征通过现成的BEVFormer [30]中的BEV编码器转换为统一的鸟瞰视图（BEV）特征B。需要注意的是，UniAD并不局限于特定的BEV编码器，用户可以利用其他替代方案提取具有长期时间融合[19, 43]或多模态融合[33, 36]的更丰富的BEV表示。在TrackFormer中，我们称之为跟踪查询的可学习嵌入从B中查询代理的信息，以检测和跟踪代理。MapFormer将地图查询作为道路元素（如车道和分隔带）的语义抽象，并执行全景分割。

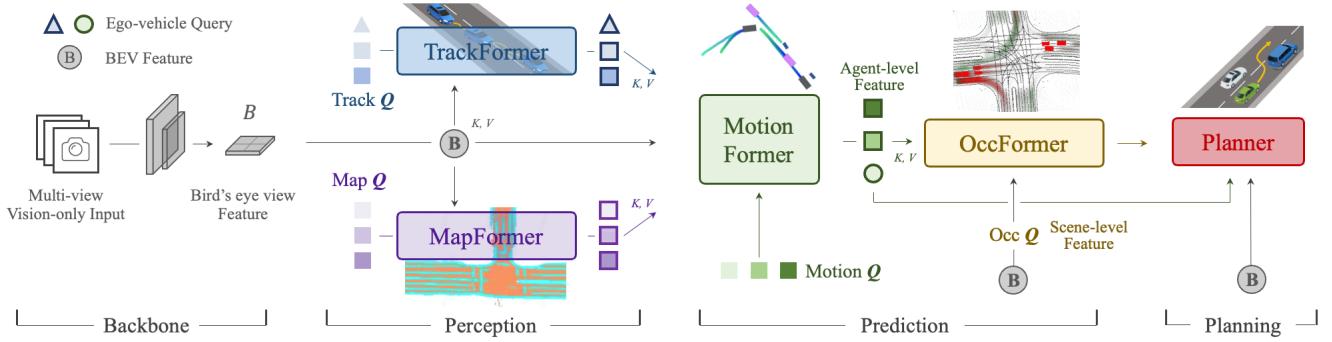


图2. 统一自动驾驶 (UniAD) 的流程。它精巧地设计遵循以规划为导向的哲学。我们不是简单地堆叠任务，而是研究每个模块在感知和预测中的效果，利用从先前节点到最终规划在驾驶场景中的联合优化优势。所有感知和预测模块都设计为transformer解码器结构，任务查询作为接口连接每个节点。最后是一个基于注意力的简单规划器，用于预测自车未来的路径点，考虑从先前节点提取的知识。占用地图仅用于视觉目的。

地图的生成。通过上述查询表示代理和地图，MotionFormer捕捉代理和地图之间的交互，并预测每个代理的未来轨迹。由于每个代理的动作可以显著影响场景中的其他代理，因此该模块对所有考虑的代理进行联合预测。同时，我们设计了一个自我车辆查询，以显式建模自我车辆，并使其能够在这种以场景为中心的范式中与其他代理进行交互。OccFormer使用BEV特征B作为查询，配备代理特定的知识作为键和值，并预测多步未来占用情况，同时保留代理身份。最后，规划器利用MotionFormer中的表达性自我车辆查询来预测规划结果，并使其远离OccFormer预测的占用区域，以避免碰撞。

2.1. 感知：追踪与地图构建

TrackFormer。 它联合执行检测和多目标跟踪 (MOT)，无需非可微的后处理步骤。受[56, 58]的启发，我们采用了类似的查询设计。除了在目标检测中使用的常规检测查询[5, 62]，还引入了额外的跟踪查询来跨帧跟踪代理。具体来说，在每个时间步，初始化的检测查询负责检测首次感知到的新生代理，而跟踪查询则持续建模在前几帧中检测到的代理。检测查询和跟踪查询都通过关注BEV特征B来捕捉代理的抽象信息。随着场景的不断演变，当前帧的跟踪查询通过自注意力模块与先前记录的跟踪查询进行交互，以聚合时间信息，直到相应代理完全消失（在一定时间段内未被跟踪）。与[5]类似，TrackFormer包含N层，最终输出状态QA为下游预测任务提供了Na个有效代理的知识。此外，查询还编码了周围其他代理的信息，

在查询集中，我们引入了一个特定的自我车辆查询，以明确地建模自动驾驶车辆本身，这在规划过程中被进一步使用。

MapFormer。 我们基于2D全景分割方法Panoptic SegFormer [31]设计了它。我们将道路元素稀疏表示为地图查询，以帮助下游的运动预测，其中编码了位置和结构知识。对于驾驶场景，我们将车道、分隔带和交叉口设置为“事物”，将可驾驶区域设置为“背景”[28]。MapFormer还具有N个堆叠层，每个层的输出结果都受到监督，而只有最后一层更新的查询QM被传递到MotionFormer以进行智能体与地图的交互。

2.2. 预测：运动预测

最近的研究已证实了transformer结构在运动任务中的有效性[24, 25, 35, 39, 40, 48, 55]，受此启发，我们最终在端到端设置中提出了MotionFormer。通过从TrackFormer和MapFormer分别获取动态代理QA和静态地图QM的高度抽象查询，MotionFormer以场景为中心的方式预测所有代理的多模态未来运动，即前k个可能的轨迹。这种范式通过单次前向传递在帧内生成多代理轨迹，大大节省了将整个场景对齐到每个代理坐标系的计算成本[27]。同时，我们将来自TrackFormer的自我车辆查询传递给MotionFormer，以使自我车辆与其他代理互动，考虑未来的动态变化。形式上，输出运动被表述为 $\{\hat{x}_{i,k} \in \mathbb{R}^{T \times 2} | i = 1, \dots, N_a; k = 1, \dots, K\}$ ，其中i索引代理，k索引轨迹模态，T为预测时间范围的长度。

MotionFormer。 它由N层组成，每层捕捉三种类型的交互：代理-代理、

agent-map和agent-goal点。对于每个运动查询 $Q_{i,k}$ （稍后定义，为简洁起见，我们在以下上下文中省略了下标 i, k ），它与其他agent QA或地图元素QM的交互可以表示为：

$$Q_{a/m} = \text{MHCA}(\text{MHSA}(Q), Q_A/Q_M),$$

其中，MHCA、MHSA分别表示多头交叉注意力和多头自注意力[50]。由于聚焦于目标位置（即目标点）以优化预测轨迹同样重要，我们设计了一种通过可变形注意力[62]实现的智能体-目标点注意力机制，如下所示：

$$Q_g = \text{DeformAttn}(Q, \hat{x}_T^{l-1}, B),$$

其中 $\hat{x}^{l-1}Tis$ 是前一层预测轨迹的终点。

$\text{DeformAttn}(q, r, x)$ 是一个可变形注意力模块 [62]，它接收查询 q 、参考点 r 和空间特征 x 。该模块在参考点周围的空间特征上执行稀疏注意力。通过这种方式，预测的轨迹进一步细化，以意识到终点周围的环境。所有三种交互都并行建模，生成的 Q_a 、 Q_m 和 Q_g 被连接并通过多层感知器 (MLP) 传递，从而得到查询上下文 $Qctx$ 。然后， $Qctx$ 被发送到后续层进行进一步细化，或在最后一层作为预测结果解码。

运动查询。 MotionFormer 每一层的输入查询被称为运动查询，包含两个组成部分：由前一层产生的查询上下文 $Qctx$ ，以及查询位置 $Qpos$ 。具体来说， $Qpos$ 整合了四方面的位置知识，如公式 (3) 所示：(1) 场景级锚点的位置 I^s ；(2) 代理级锚点的位置 I^a ；(3) 当前代理 i 的位置；(4) 预测的目标点。

$$\begin{aligned} Q_{pos} = & \text{MLP}(\text{PE}(I^s)) + \text{MLP}(\text{PE}(I^a)) \\ & + \text{MLP}(\text{PE}(\hat{x}_0)) + \text{MLP}(\text{PE}(\hat{x}_T^{l-1})). \end{aligned} \quad (3)$$

此处采用正弦位置编码 $\text{PE}(\cdot)$ 后接一个MLP来编码位置点，并将 \hat{x}^0Tis 设为 I^s 在第一层（下标 i, k 省略）。场景级锚点表示全局视角下的先验运动统计信息，而代理级锚点捕捉局部坐标系中的可能意图。它们均通过k-means 算法在真实轨迹的端点上进行聚类，以缩小预测的不确定性。与先验知识相反，起始点为每个代理提供定制的位置嵌入，而预测的端点则作为动态锚点，在由粗到细的方式中逐层优化。

非线性优化。 与那些可以直接访问

地面实况感知结果，即代理的位置及其对应轨迹，我们在端到端范式中考虑了先验模块的预测不确定性。直接从检测不完美位置或航向角回归地面实况航点可能会导致具有大曲率和加速度的不现实轨迹预测。为解决这一问题，我们采用了一个非线性平滑器[4]来调整目标轨迹，使其在给定上游模块预测的不精确起点时物理上可行。具体过程如下：

$$\tilde{x}^* = \arg \min_{\mathbf{x}} c(\mathbf{x}, \tilde{\mathbf{x}}),$$

其中， $\tilde{\mathbf{x}}$ 和 \mathbf{x}^* 分别表示真实轨迹和平滑后的轨迹， \mathbf{x} 由多重射击法生成[2]，成本函数如下：

$$c(\mathbf{x}, \tilde{\mathbf{x}}) = \lambda_{xy} \|\mathbf{x}, \tilde{\mathbf{x}}\|_2 + \lambda_{goal} \|\mathbf{x}_T, \tilde{\mathbf{x}}_T\|_2 + \sum_{\phi \in \Phi} \phi(\mathbf{x}), \quad (5)$$

其中， xy 和 $goal$ 是超参数，运动学函数集 Φ 包含五个项：加加速度、曲率、曲率变化率、加速度和侧向加速度。成本函数使得目标轨迹符合运动学约束。这种目标轨迹优化仅在训练过程中进行，不影响推理。

2.3. 预测：占用预测

占据网格地图是一种离散化的BEV表示，其中每个单元格保存一个信念值，指示其是否被占据，而占据预测任务是发现网格地图在未来如何变化。先前的方法利用RNN结构从观测到的BEV特征中进行时间扩展的未来预测 [20,23,59]。然而，它们依赖于高度手工设计的聚类后处理来生成每个代理的占据地图，因为它们大多是代理无关的，将整个BEV特征压缩为RNN隐藏状态。由于缺乏对代理知识的充分利用，它们难以全局预测所有代理的行为，这对于理解场景如何演变至关重要。为此，我们提出了OccFormer，以在两个方面结合场景级和代理级语义：(1) 密集场景特征通过一个精心设计的注意力模块在展开到未来时获取代理级特征；(2) 我们通过代理级特征和密集场景特征之间的矩阵乘法轻松生成实例化的占据地图，而无需繁重的后处理。

OccFormer由 T_o 个顺序块组成，其中 T_o 表示预测范围。需要注意的是，由于密集表示占用的计算成本较高， T_o 通常小于运动任务中的 T 。每个块以丰富的代理特征 G^t 和前一层的状态（密集特征） F^{t-1} 作为输入，并生成 F^t 。

考虑实例级和场景级信息的时间步长 t 。为了获取具有动态和空间先验的代理特征 G^t ，我们通过在模态维度上对来自 MotionFormer 的运动查询进行最大池化，记为 $Q_X \in \mathbb{R}^{Na \times D}$ ，其中 D 表示特征维度。然后，我们通过一个特定于时间的 MLP 将其与上游的跟踪查询 QA 和当前位置嵌入 PA 进行融合：

$$G^t = \text{MLP}_t([Q_A, P_A, Q_X]), t = 1, \dots, T_o,$$

其中 $[\cdot]$ 表示连接。对于场景级别的知识，BEV 特征 B 被下采样到 $1/4$ 分辨率以提高训练效率，并作为第一个块的输入 F^0 。为了进一步节省训练内存，每个块采用下采样-上采样的方式，并在中间加入一个注意力模块，以在 $1/8$ 下采样的特征上进行像素-代理交互，记为 $F^t ds$ 。

像素-代理交互旨在统一场景级和代理级理解，以预测未来的占用情况。我们以密集特征 $F^t ds$ 作为查询，实例级特征作为键和值，随时间更新密集特征。具体而言， $F^t ds$ 通过自注意力层来建模远距离网格间的响应，然后交叉注意力层建模代理特征 G^t 与每个网格特征之间的交互。此外，为对齐像素-代理对应关系，我们通过注意力掩码来约束交叉注意力，该掩码限制每个像素仅在时间步 t 时查看占据它的代理，受 [10] 启发。密集特征的更新过程表述为：

$$D_{ds}^t = \text{MHCA}(F_{ds}^t, G^t, \text{attn_mask} = O_m^t).$$

注意力掩码 O_m^t 在语义上类似于占用情况，并通过乘以一个额外的代理级别特征和密集特征 $F^t ds$ 生成，我们将这里的代理级别特征命名为掩码特征 $M^t = \text{MLP}(G^t)$ 。在公式(7)的交互过程之后， $D^t ds$ 被上采样到 B 的 $1/4$ 大小。我们进一步将 $D^t ds$ 与块输入 $F^{t-1} as$ 进行残差连接，并将生成的特征 F^t 传递到下一个块。

实例级占用。 它表示保留了每个代理身份的占用情况。可以通过矩阵乘法简单绘制，如最近的基于查询的分割工作 [11, 29] 所示。正式地，为了获得 BEV 特征 B 的原始尺寸 $H \times W$ 的占用预测，场景级特征 F^t 通过卷积解码器上采样到 $F^{t dec} \in \mathbb{R}^{C \times H \times W}$ ，其中 C 是通道维度。对于代理级特征，我们进一步将粗略掩码特征 M^t 更新为占用特征 $U^t \in \mathbb{R}^{Na \times C}$ ，通过另一个 MLP。我们经验性地发现，从掩码特征 M^t 生成 U^t 而不是原始代理特征 G^t 能带来更好的性能。最终的时间步 t 的实例级占用为：

$$\hat{O}_A^t = U^t \cdot F_{dec}^t.$$

2.4. 规划

在没有高清 (HD) 地图或预定义路线的情况下进行规划，通常需要高层指令来指示行进方向 [6, 23]。接着，我们将原始导航信号（即左转、右转和直行）转换为三种可学习的嵌入，称为指令嵌入。由于 MotionFormer 中的自行车查询已经表达了其多模态意图，我们将其与指令嵌入结合，形成一个“规划查询”。我们将规划查询与 BEV 特征 B 进行注意力处理，使其感知周围环境，然后将其解码为未来的路径点 $\hat{\tau}$ 。为进一步避免碰撞，我们仅在推理过程中基于牛顿法对 $\hat{\tau}$ 进行优化，具体如下：

$$\tau^* = \arg \min_{\tau} f(\tau, \hat{\tau}, \hat{O}),$$

其中， $\hat{\tau}$ 是原始规划预测， τ^* 表示优化后的规划，该规划从多重射击 [2] 轨迹 τ 中选出，以最小化成本函数 $f(\cdot)$ 。 \hat{O} 是一个经典的二进制占用图，由 OccFormer 的实例占用预测合并而成。成本函数 $f(\cdot)$ 的计算公式为：

$$f(\tau, \hat{\tau}, \hat{O}) = \lambda_{\text{coord}} \|\tau, \hat{\tau}\|_2 + \lambda_{\text{obs}} \sum_t \mathcal{D}(\tau_t, \hat{O}^t), \quad (11)$$

$$\mathcal{D}(\tau_t, \hat{O}^t) = \sum_{(x,y) \in \mathcal{S}} \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{\|\tau_t - (x,y)\|_2^2}{2\sigma^2}\right).$$

这里， coord 、 obs 和 \mathcal{S} 是超参数， t 索引未来的时间步长。 $\|\cdot\|_2$ 成本将轨迹拉向原始预测的轨迹，而碰撞项 \mathcal{D} 则将其从被占用的网格中推开，考虑到周围位置被限制在 $\mathcal{S} = \{(x, y) | \|(\mathbf{x}, \mathbf{y}) - \tau_t\|_2 \leq \hat{O}^t \mathbf{x}, \mathbf{y} = 1\}$ 。

2.5. 学习

UniAD 的训练分为两个阶段。我们首先共同训练感知部分，即跟踪和映射模块，进行几个周期（在我们的实验中为 6 个周期），然后端到端地训练模型，包括所有的感知、预测和规划模块，共 20 个周期。经验上发现，两阶段的训练更为稳定。我们建议读者参考补充材料以获取每个损失的详细信息。

共享匹配。 由于 UniAD 涉及实例级别的建模，因此在感知和预测任务中需要将预测结果与真实数据集配对。类似于 DETR [5, 31]，在跟踪和在线映射阶段采用了二分匹配算法。对于跟踪任务，来自检测查询的候选对象与新生成的真实对象配对，而来自跟踪查询的预测结果则继承了前一帧的分配。跟踪模块中的匹配结果在运动和占用节点中被复用，以便在端到端框架中持续地从历史轨迹建模到未来的运动。

ID	Modules				Tracking			Mapping		Motion Forecasting			Occupancy Prediction			Planning			
	Track	Map	Motion	Occ.	Plan	AMOTA	AMOTP	IDS	IoU-lane	IoU-road	minADE	minFDE	MR	IoU-n.	IoU-f.	VPQ-n.	VPQ-f.	avg.L2	avg.Col.
						0.356	1.328	893	0.302	0.675	0.858	1.270	0.186	55.9	34.6	47.8	26.4	1.154	0.941
						0.348	1.333	791	0.305	0.674									
						0.355	1.336	785	0.301	0.671									
						0.360	1.350	919	0.303	0.672	0.815	1.224	0.182						
						0.354	1.339	820	0.303	0.672	0.751	1.109	0.162						
						0.360	1.322	809	0.304	0.675	0.736(-9.7%)	1.066(-12.9%)	0.158						
						0.359	1.359	1057			0.710(-3.5%)	1.005(-5.8%)	0.146	60.5	37.0	52.4	29.8		
10						0.366	1.337	889	0.303	0.672	0.741	1.077	0.157	62.1	38.4	52.2	32.1		
11						0.358	1.334	641	0.302	0.672	0.728	1.054	0.154	62.3	39.4	53.1	32.2		
12																	1.131	0.773	
																	1.014	0.717	
																	1.004	0.430	

表2. 各任务效果的详细消融分析。我们可以得出结论，两个感知子任务极大地促进了运动预测，而统一两个预测模块也有助于提升预测性能。在所有先验表示下，我们的目标规划显著提升，以确保安全性。UniAD在预测和规划任务上大幅超越了简单的多任务学习（MTL）解决方案，并且具有感知性能无显著下降的优势。为简洁起见，仅展示主要指标。“avg.L2”和“avg.Col”是规划时间范围内的平均值。*：ID-0是每个任务具有独立头的MTL方案。

3. 实验

我们在具有挑战性的nuScenes数据集[3]上进行了实验。在本节中，我们验证了设计在三个方面的有效性：联合结果揭示了任务协调的优势及其对规划的影响，各任务的模块化结果与之前的方法相比，以及对特定模块设计空间的消融研究。由于篇幅限制，完整的协议套件、部分消融研究和可视化结果在补充材料中提供。

3.1. 联合结果

我们进行了广泛的消融实验，如表2所示，以证明在端到端管道中前置任务的有效性和必要性。表中的每一行显示了在引入第二列“模块”中列出的任务模块时的模型性能。第一行（ID-0）作为基准的多任务模型，具有单独的任务头，用于比较。每个指标的最佳结果用粗体标记，每个列中的次优结果用下划线标记。

安全规划路线图。由于预测相较于感知更接近规划，我们首先研究了框架中的两种预测任务，即运动预测和占用预测。在实验10-12中，只有当这两个任务同时引入时（实验12），规划的L2指标和碰撞率均达到最佳结果，相比没有任何中间任务的直接端到端规划（实验10，图1(c.1)）。因此我们得出结论，这两种预测任务对于实现安全规划目标都是必要的。退一步说，在实验7-9中，我们展示了两种预测类型的协同效应。当这两项任务紧密结合时（实验9，-3.5% minADE，-5.8% minFDE，-1.3 MR%，+2.4 IoU-f.%，+2.4 VPQ-f.%），两项任务的性能都得到了提升，这证明了包含代理和场景表示的必要性。同时，为了实现更优的运动预测，

通过一系列实验，我们探讨了感知模块在实验4-6中如何贡献于性能提升。值得注意的是，结合跟踪和映射节点显著改善了预测结果（-9.7% minADE，-12.9% minFDE，-2.3 MR%）。我们还展示了实验1-3，这些实验表明，同时训练感知子任务能获得与单一任务相当的结果。此外，与简单的多任务学习（实验0，图1(b)）相比，实验12在所有关键指标上显著优于前者（-15.2% minADE，-17.0% minFDE，-3.2 MR%），+4.9 IoU-f.%，+5.9 VPQ-f.%，-0.15m avg.L2，-0.51 avg.Col%），展示了我们以规划为导向设计的优越性。

3.2. 模块化结果

按照感知-预测-规划的顺序，我们报告了每个任务模块在nuScenes验证集上与先前最先进的性能对比。需要注意的是，UniAD通过单一训练的网络共同执行所有这些任务。每个任务的主要指标在表格中以灰色背景标记。

感知结果。在表3的多目标跟踪中，UniAD相较于MUTR3D[58]和ViP3D[18]分别提升了+6.5和+14.2的AMOTA%。此外，UniAD实现了最低的ID切换分数，显示出其在每个轨迹片段中的时间一致性。在表4的在线地图构建中，UniAD在车道分割方面表现出色（与BEVFormer相比，IoU%提升了+7.4），这对运动模块中的下游代理-道路交互至关重要。由于我们的跟踪模块遵循端到端范式，它在复杂关联的跟踪检测方法（如Immortal Tracker[52]）面前仍显不足，且在特定类别上的地图构建结果落后于之前以感知为导向的方法。我们认为，UniAD旨在通过感知信息来优化最终规划，而非以全模型容量来优化感知。

Method	AMOTA	AMOTP	Recall	IDS
Immortal Tracker [52]	0.378	1.119	0.478	936
ViP3D [18]	0.217	1.625	0.363	
QD3DT [21]	0.242	1.518	0.399	
MUTR3D [58]	0.294	1.498	0.427	3822
UniAD	0.359	1.320	0.467	906

表3. 多目标跟踪。UniAD 优于先前的所有基于图像输入的端到端多目标跟踪技术（仅限图像输入）指标。†：基于检测的跟踪方法，带有后关联，使用BEVFormer进行了重新实现，以进行公平比较。

Method	Lanes	Drivable	Divider	Crossing
VPN [42]	18.0	76.0		
LSS [44]	18.3	73.9		
BEVFormer [30]	23.9	77.5		
BEVerse [59]		30.6	17.2	
UniAD	31.3	69.1	25.7	13.8

表4. 在线映射。UniAD在与最先进的感知导向方法的竞争中表现出色，综合道路语义。我们报告分割IoU (%)。†: 使用BEVFormer重新实现。

Method	minADE(m)	minFDE(m)	MR	EPA
PnPNet [32]	1.15	1.95	0.226	0.222
ViP3D [18]	2.05	2.84	0.246	0.226
Constant Pos.	5.80	10.27	0.347	
Constant Vel.	2.13	4.01	0.318	
UniAD	0.71	1.02	0.151	0.456

表5. 运动预测。UniAD显著优于先前的基于视觉的端到端方法。我们还报告了两种将车辆建模为常位置或常速度的设置作为比较。†: 使用BEVFormer重新实现。

预测结果。运动预测结果如表5所示，其中UniAD显著优于以往基于视觉的端到端方法。与PnPNet-vision [32]和ViP3D [18]相比，UniAD分别减少了38.3%和65.4%的minADE预测误差。在表6报告的占用预测方面，UniAD在附近区域取得了显著进展，相对于FIERY [20]和BEVerse [59]（后者采用了大量数据增强），UniAD在IoU-near(%)上分别提高了+4.0和+2.0。

规划结果。得益于自车查询和占用中丰富的时空信息，UniAD在规划视界的平均值方面，将规划L2误差和碰撞率分别降低了51.2%和56.3%，优于ST-P3[23]。此外，它显著超越了多个基于激光雷达的同类系统，这在感知任务中通常被认为具有挑战性。

3.3. 定性结果

图3展示了在一个复杂场景中所有任务的结果。自车在行驶时注意到了潜在的

Method	IoU-n.	IoU-f.	VPQ-n.	VPQ-f.
FIERY [20]	59.4	36.7	50.2	29.9
StretchBEV []	55.5	37.1	46.0	29.0
ST-P3 [23]		38.9		32.1
BEVerse [59]	61.4	40.9	54.3	36.1
UniAD	63.4	40.2	54.7	33.5

表6. 占用预测。UniAD在邻近区域取得了显著改进，这些区域对规划更为关键。“n.”

“f.” 分别表示近（ 30×30 米）和远（ 50×50 米）的评估范围。
†：经过大量数据增强训练。

Method	L2(m)				Col. Rate(%)			
	Avg.		Avg.		Avg.		Avg.	
NMP [57]		2.31				1.92		
SA-NMP [57]		2.05				1.59		
FF [22]	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43
EO [26]	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33
ST-P3 [23]	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31

表7. 规划。UniAD在所有时间间隔内实现了最低的L2误差和碰撞率，甚至优于基于LiDAR的方案。

方法 (†) 在大多数情况下，验证我们系统的安全性。

ID	Scene-l. Anch.	Goal Inter.	Ego Q	NLO.	Col. Rate(%)			
					minADE	minFDE	MR	minFDE -mAP
					0.844	1.336	0.177	0.246
					0.768	1.159	0.164	0.267
					0.755	1.130	0.168	0.264
					0.747	1.096	0.156	0.266
					0.710	1.004	0.146	0.273

表8. 运动预测模块中的设计消融实验。所有组件都对最终性能有所贡献。“Scene-l. Anch.” 表示旋转的场景级锚点。“Goal Inter.” 表示智能体与目标点的交互。“Ego Q” 代表自我车辆查询，“NLO.” 是非线性优化策略。*：一种同时考虑检测和预测准确性的指标，详细信息见补充材料。

前车和车道的运动。在补充材料中，我们展示了更多具有挑战性的场景的可视化，以及一个面向规划设计的潜在成功案例，即在前模块结果不准确的情况下，后续任务仍能恢复，例如，尽管物体有较大的航向角偏差或在跟踪结果中未能被检测到，规划的轨迹仍然合理。此外，我们还分析了UniAD的失败案例，主要集中在一些长尾场景，如大型卡车和拖车，这些也在补充材料中展示。

3.4. 消融研究

设计在MotionFormer中的效果。表8显示，我们在第2.2节中描述的所有提出的组件都对minADE、minFDE、Miss Rate和minFDE-mAP指标的最终性能有所贡献。值得注意的是，旋转场景级锚点显示出显著的性能提升（-15.8% minADE，-11.2% minFDE，+1.9 minFDE-mAP(%)）。

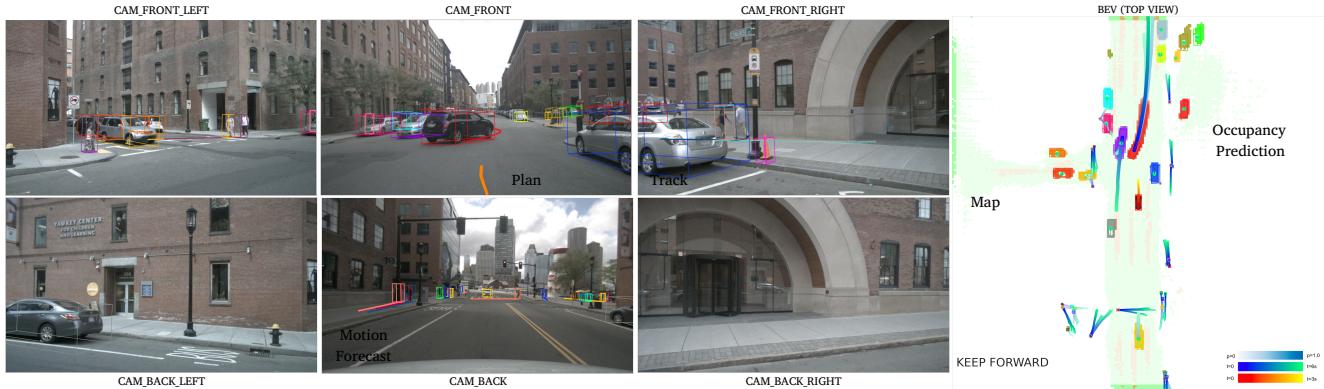


图3. 可视化结果。我们在环绕视图图像和BEV中展示了所有任务的结果。运动和占用模块的预测是一致的，在此情况下，自行车正在让行前方黑色车辆。每个代理以独特颜色表示。仅top-1和top-3轨迹分别在图像视图和BEV上进行可视化。

ID	Cross. Attn.	Attn. Mask	Mask Feat.	IoU-n.	IoU-f.	VPQ-n.	VPQ-f.
				61.2	39.7	51.5	31.8
✓				61.3	39.4	51.0	31.8
✓	✓			62.3	39.7	52.4	32.5
✓	✓	✓		62.6	39.5	53.2	32.8

表9. 占用预测模块中设计的消融实验。带有掩码的交叉注意力和掩码特征的重用有助于提高预测效果。“交叉注意”和“注意掩码”

表示像素-代理中的交叉注意力和注意力掩码

交互分别。“Mask Feat.”表示重用

实例级占用的掩码特征。

ID	BEV Att.	Col. Loss	Occ. Optim.	L2			Col. Rate
				0.44	0.99	1.71	0.56 0.88 1.64
✓				0.44	1.04	1.81	0.35 0.71 1.58
✓	✓			0.44	1.02	1.76	0.30 0.51 1.39
✓	✓	✓		0.54	1.09	1.81	0.13 0.42 1.05

表10. 规划模块中设计的消融实验。结果展示了每个前置任务的必要性。“BEV Att.”表示关注BEV特征。“Col. Loss”表示碰撞损失。“Occ. Optim.”是基于占位的优化策略。

表明在以场景为中心的方式下进行运动预测是至关重要的。代理-目标点交互增强了运动查询与面向规划的视觉特征，而周围代理通过考虑自行车的意图可以进一步受益。此外，非线性优化策略通过在端到端场景中考虑感知不确定性，提升了性能（-5.0% minADE，-8.4% minFDE，-1.0 MR%，+0.7 minFDE-mAP%）。

OccFormer中设计的影响。如表9所示，对每个像素进行无局部约束的全局注意力（实验2）相比无注意力的基线（实验1），性能略有下降。占用-

引导注意力掩码解决了问题并带来了增益，特别是在邻近区域（Exp.3, +1.0 IoU-n.%, +1.4 VPQ-n.%）。此外，重用掩码特征M^t而不是代理特征来获取占用特征进一步提升了性能。

设计在规划器中的影响。我们在表10中对规划器中提出的设计进行了消融实验，即关注BEV特征、使用碰撞损失进行训练以及采用占用优化策略。与先前的研究[22,23]类似，相比于单纯的轨迹模仿（L2度量），更低的碰撞率更符合安全性的需求，并且在UniAD中应用所有部分后，碰撞率有所降低。

4. 结论与未来工作

我们讨论了自动驾驶算法框架的系统级设计。提出了一个面向规划的流水线，即UniAD，以追求最终的规划目标。我们对感知和预测中每个模块的必要性进行了详细分析。为了统一任务，提出了基于查询的设计，以连接UniAD中的所有节点，得益于环境中更丰富的代理交互表示。广泛的实验验证了所提出方法在各个方面的有效性。

局限性与未来工作。协调这样一个包含多任务的综合系统并非易事，尤其在需要时间历史数据训练的情况下，它需要大量的计算资源。如何设计和优化系统以实现轻量级部署，值得未来探索。此外，是否应纳入更多任务，如深度估计、行为预测，以及如何将它们嵌入系统中，也是值得探讨的未来方向。

致谢。本工作得到中国国家重点研发计划（2022ZD0160100）、上海市科学技术委员会（21DZ1100100）及国家自然科学基金（62206172）的部分资助。

参考文献

- [1] Adil Kaan Akan 和 Fatma Güney。StretchBEV：在空间和时间上延伸未来实例预测。发表于ECCV，2022年。7
- [2] Hans Georg Bock 和 Karl-Josef Plitt。一种用于直接求解最优控制问题的多重打靶算法。IFAC Proceedings Volumes，1984年。4, 5
- [3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liang, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, 和 Oscar Beijbom。nuscenes：一个用于自动驾驶的多模态数据集。发表于CVPR，2020年。6
- [4] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wol , Alex Lang, Luke Fletcher, Oscar Beijbom, 和 Sammy Omari。nuplan：一个基于ML的闭环规划基准，用于自动驾驶车辆。arXiv预印本arXiv:2106.11810 , 2021年。4
- [5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, 和 Sergey Zagoruyko。使用Transformer的端到端目标检测。发表于ECCV，2020年。3, 5
- [6] Sergio Casas, Abbas Sadat, 和 Raquel Urtasun。Mp3：一个统一的模型，用于地图、感知、预测和规划。发表于CVPR，2021年。2, 5
- [7] Dian Chen, Vladlen Koltun, 和 Philipp Krähenbühl。从世界的轨道中学习驾驶。发表于ICCV，2021年。2
- [8] Dian Chen 和 Philipp Krähenbühl。从所有车辆中学习。发表于CVPR，2022年。1, 2
- [9] Dian Chen, Brady Zhou, Vladlen Koltun, 和 Philipp Krähenbühl。通过作弊学习。发表于CoRL，2020年。2
- [10] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, 和 Rohit Girdhar。用于通用图像分割的掩码注意力掩码Transformer。发表于CVPR，2022年。5
- [11] Bowen Cheng, Alex Schwing, 和 Alexander Kirillov。逐像素分类并不是语义分割的全部。发表于NeurIPS，2021年。5
- [12] Kashyap Chitta, Aditya Prakash, 和 Andreas Geiger。NEAT：用于端到端自动驾驶的神经注意力场。发表于ICCV，2021年。2
- [13] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, 和 Andreas Geiger。Transfuser：基于Transformer的传感器融合用于自动驾驶的模仿学习。IEEE TPAMI，2022年。1
- [14] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, 和 Alexey Dosovitskiy。通过条件模仿学习实现端到端驾驶。发表于ICRA，2018年。2
- [15] Felipe Codevilla, Eder Santana, Antonio M López, 和 Adrien Gaidon。探索行为克隆在自动驾驶中的局限性。发表于ICCV，2019年。2
- [16] Michael Crawshaw。多任务学习与深度神经网络：一项调查。arXiv预印本arXiv:2009.09796 , 2020年。2
- [17] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, 和 Vladlen Koltun。CARLA：一个开放的城市驾驶模拟器。发表于CoRL，2017年。2
- [18] 顾俊儒, 胡晨旭, 张天元, 陈晓耀, 王一伦, 王越, 赵航。ViP3D: 通过3D代理查询实现端到端视觉轨迹预测. 在CVPR, 2023. 2, 6, 7
- [19] 韩春瑞, 孙建建, 葛铮, 杨金荣, 董润培, 周宏宇, 毛伟新, 彭光, 张翔宇. 探索多视角3D感知中长期时间融合的递归融合. arXiv预印本arXiv:2303.05970, 2023. 2
- [20] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Je rey Hawke, Vijay Badrinarayanan, Roberto Cipolla, 和Alex Kendall. FIERY: 从环绕单目相机进行鸟瞰图的未来实例预测. 在ICCV, 2021. 4, 7
- [21] Hou-Ning Hu, Yung-Hsu Yang, Tobias Fischer, Trevor Darrell, Fisher Yu, 和Min Sun. 单目准密集3D物体追踪. IEEE TPAMI, 2022. 7
- [22] 胡培云, Aaron Huang, John Dolan, David Held, 和Deva Ramanan. 通过自监督自由空间预测实现安全本地运动规划. 在CVPR, 2021. 7, 8
- [23] 胡胜超, 陈力, 吴鹏浩, 李宏洋, 严俊驰, 和陶大程。ST-P3: 通过时空特征学习实现端到端基于视觉的自动驾驶. 在ECCV, 2022. 2, 4, 5, 7, 8
- [24] 贾晓松, 孙立廷, 赵航, Masayoshi Tomizuka, 和魏展. 通过结合自我中心和客观视角实现多智能体轨迹预测. 在CoRL, 2021. 3
- [25] 贾晓松, 吴鹏浩, 陈力, 李宏洋, 刘宇, 和严俊驰. HDGT: 通过场景编码实现多智能体轨迹预测的异构驾驶图变换器. arXiv预印本arXiv:2205.09753, 2022. 3
- [26] Tarasha Khurana, 胡培云, Achal Dave, Jason Ziglar, David Held, 和Deva Ramanan. 通过可微分光线投射实现自监督占用预测. 在ECCV, 2022. 7
- [27] Jinkyu Kim, Reza Mahjourian, Scott Ettinger, Mayank Bansal, Brandy White, Ben Sapp, 和Dragomir Anguelov. Stopnet: 城市自动驾驶的可扩展轨迹和占用预测. arXiv预印本arXiv:2206.00991, 2022. 3
- [28] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, 和Piotr Dollár. 全景分割. 在CVPR, 2019. 3
- [29] 李峰, 张浩, 刘世龙, 张磊, Lionel M Ni, 和沈向洋。Mask DINO: 面向物体检测和分割的统一基于变换器的框架. 在CVPR, 2023. 5
- [30] 李志奇, 王文海, 李宏洋, 谢恩泽, 司马崇豪, 卢彤, 乔桥, 和戴继峰。BEVFormer: 通过时空变换器从多相机图像学习鸟瞰图表示. 在ECCV, 2022. 2, 7
- [31] 李志奇, 王文海, 谢恩泽, 余志定, Anima Anandkumar, Jose M Alvarez, Ping Luo, 和卢彤. 全景分割变换器: 深入全景分割与变换器. 在CVPR, 2022. 3, 5

- [32] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, 和 Raquel Urtasun. Pnpnet: 端到端的感知与预测框架 , 结合了跟踪功能。发表于CVPR , 2020年。1, 2, 7
- [33] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, 和 Zhi Tang. BEVFusion: 一个简单且强大的激光雷达-摄像头融合框架。发表于NeurIPS , 2022年。2
- [34] Xiwen Liang, Yangxin Wu, Jianhua Han, Hang Xu, Chun-jing Xu, 和 Xiaodan Liang. 在多任务协同训练中实现统一自动驾驶的有效适应。发表于NeurIPS , 2022年。1
- [35] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, 和 Bolei Zhou. 使用堆叠Transformer的多模态运动预测。发表于CVPR , 2021年。3
- [36] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huiyi Mao, Daniela Rus, 和 Song Han. BEVFusion: 使用统一鸟瞰图表示的多任务多传感器融合。发表于ICRA , 2023年。2
- [37] Wenjie Luo, Bin Yang, 和 Raquel Urtasun. 快速且狂暴 : 使用单一卷积网络实现实时3D检测、跟踪和运动预测。发表于CVPR , 2018年。1
- [38] Mobileye. Mobileye 内部揭秘。<https://www.mobileye.com/ces-2022/> , 2022年。1, 2
- [39] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, 和 Benjamin Sapp. Wayformer: 通过简单且高效的注意力网络实现运动预测。arXiv预印本 arXiv:2207.05844 , 2022年。3
- [40] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Je rey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, David Weiss, Ben Sapp, Zhifeng Chen, 和 Jonathon Shlens. 场景Transformer : 用于行为预测和规划的统一多任务模型。发表于ICLR , 2022年。3
- [41] Nvidia. NVIDIA DRIVE 端到端解决方案 , 适用于自动驾驶车辆。<https://developer.nvidia.com/drive> , 2022年。1, 2
- [42] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, 和 Bolei Zhou. 跨视图语义分割 , 用于感知周围环境。IEEE RA-L , 2020年。7
- [43] Jinyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, 和 Wei Zhan. 时间将揭示 : 对时间多视图3D物体检测的新视角及基线。arXiv预印本 arXiv:2210.02443 , 2022年。2
- [44] Jonah Philion 和 Sanja Fidler. Lift, splat, shoot: 通过隐式反投影到3D空间来编码任意相机设置的图像。发表于ECCV , 2020年。7
- [45] Aditya Prakash, Kashyap Chitta, 和 Andreas Geiger. 用于端到端自动驾驶的多模态融合Transformer。发表于CVPR , 2021年。2
- [46] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, 和 Nando de Freitas. 一个通用智能体。arXiv预印本 arXiv:2205.06175 , 2022年。1
- [47] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, 和 Raquel Urtasun. 感知、预测、规划 : 通过可解释的语义表示实现安全运动规划。在 ECCV , 2020. 1, 2
- [48] Shaoshuai Shi, Li Jiang, Dengxin Dai, 和 Bernt Schiele. 运动变换器与全局意图定位和局部运动细化。在 NeurIPS , 2022. 3
- [49] Tesla. Tesla AI Day. https://www.youtube.com/watch?v=ODSJsviD_SU , 2022. 2
- [50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko-reit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, 和 Illia Polosukhin. 注意力就是你所需要的。在 NeurIPS , 2017. 4
- [51] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhihang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, 和 Hongxia Yang. OFA: 通过简单的序列到序列学习框架统一架构、任务和模态。在 ICML , 2022. 1
- [52] Qitai Wang, Yuntao Chen, Ziqi Pang, Naiyan Wang, 和 Zhaoxiang Zhang. Immortal Tracker: Tracklet永不死。arXiv 预印本 arXiv:2111.13672 , 2021. 6, 7
- [53] Penghao Wu, Li Chen, Hongyang Li, Xiaosong Jia, Junchi Yan, 和 Yu Qiao. 通过自监督几何建模进行自动驾驶的策略预训练。在 ICLR , 2023. 2
- [54] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, 和 Yu Qiao. 轨迹引导的控制预测用于端到端自动驾驶 : 一个简单但强大的基线。在 NeurIPS , 2022. 2
- [55] Ye Yuan, Xinshuo Weng, Yanglan Ou, 和 Kris M Kitani. Agentformer: 用于社会时空多智能体预测的智能体感知变换器。在 ICCV , 2021. 3
- [56] Fangao Zeng, Bin Dong, Tiancai Wang, Xiangyu Zhang, 和 Yichen Wei. MOTR: 使用变换器的端到端多目标跟踪。在 ECCV , 2021. 3
- [57] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, 和 Raquel Urtasun. 端到端可解释神经运动规划器。在 CVPR , 2019. 1, 2, 7
- [58] Tianyuan Zhang, Xuanyao Chen, Yue Wang, Yilun Wang, 和 Hang Zhao. MUTR3D: 通过3D到2D查询的多相机跟踪框架。在 CVPR Workshop , 2022. 3, 6, 7
- [59] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, 和 Jiwen Lu. BEVerse: 鸟瞰视角下的统一感知和预测 , 用于以视觉为中心的自动驾驶。arXiv 预印本 arXiv:2205.09743 , 2022. 1, 2, 4, 7
- [60] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, 和 Luc Van Gool. 通过模仿强化学习教练的端到端城市驾驶。在 ICCV , 2021. 2
- [61] Jinguo Zhu, Xizhou Zhu, Wenhui Wang, Xiaohua Wang, Hongsheng Li, Xiaogang Wang, 和 Jifeng Dai. Uni-Perceiver-MoE: 通过条件MoEs学习稀疏的通用模型。在 NeurIPS , 2022. 1
- [62] 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变换器。在 ICLR , 2020. 3, 4