Presentation

Planning-oriented Autonomous Driving

Yihan Hu[1,2*], Jiazhi Yang[1*], Li Chen[1*†], Keyu Li[1*], Chonghao Sima[1], Xizhou Zhu[3,1]
Siqi Chai[2], Senyao Du[2], Tianwei Lin[2], Wenhai Wang[1], Lewei Lu[3], Xiaosong Jia[1]
Qiang Liu[2], Jifeng Dai[1], Yu Qiao[1], Hongyang Li[1†]

[1] OpenDriveLab and OpenGVLab, Shanghai AI Laboratory
[2] Wuhan University [3] SenseTime Research
[*]Equal contribution [†]Project lead

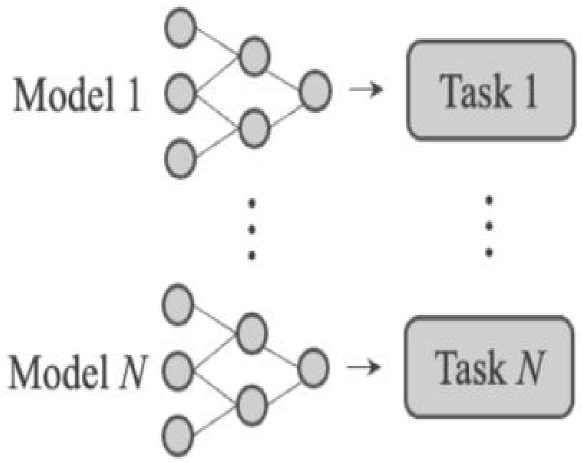
主讲人：陶柏安
日期：2025.11.19

1 Introduction

现代自动驾驶系统通常被描述为按照顺序执行的模块化任务，即感知、预测和规划。现有技术路线如下：

1. 大多数工业方案在车载芯片资源允许的情况下，为每个任务独立部署单独模型

→简化了研发难度，但模块之间的隔离会带来信息损失、误差积累以及特征不对齐的风险



(a) Standalone Models

1 Introduction

现代自动驾驶系统通常被描述为按照顺序执行的模块化任务，即感知、预测和规划。现有技术路线如下：

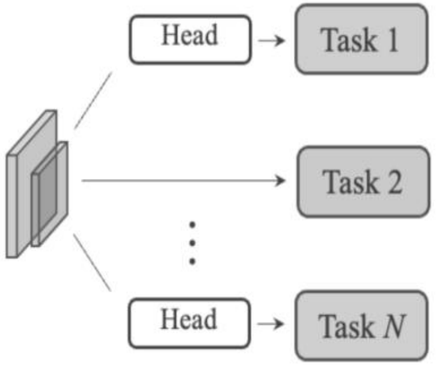
1. 大多数工业方案在车载芯片资源允许的情况下，为每个任务独立部署单独模型

→简化了研发难度，但模块之间的隔离会带来信息损失、误差积累以及特征不对齐的风险

2. 将大量任务纳入一个多任务学习（MTL）范式，通过将多个任务特定头插入共享特征提取器中。这种方式在多个领域应用，包括通用视觉、自动驾驶，如 Transfuser、BEVverse，以及 Mobileye、Tesla、Nvidia 等工业产品

→跨任务联合训练利用共享特征抽象；同时便于扩展到更多任务，并节省车载计算资源，但会导致不良的“负迁移”

Design	Approach	Perception			Prediction		Plan
		Det.	Track	Map	Motion	Occ.	
(b)	NMP [101]	✓			✓		✓
	NEAT [19]			✓			✓
	BEVerse [105]	✓		✓		✓	



(b) Multi-task Framework

1 Introduction

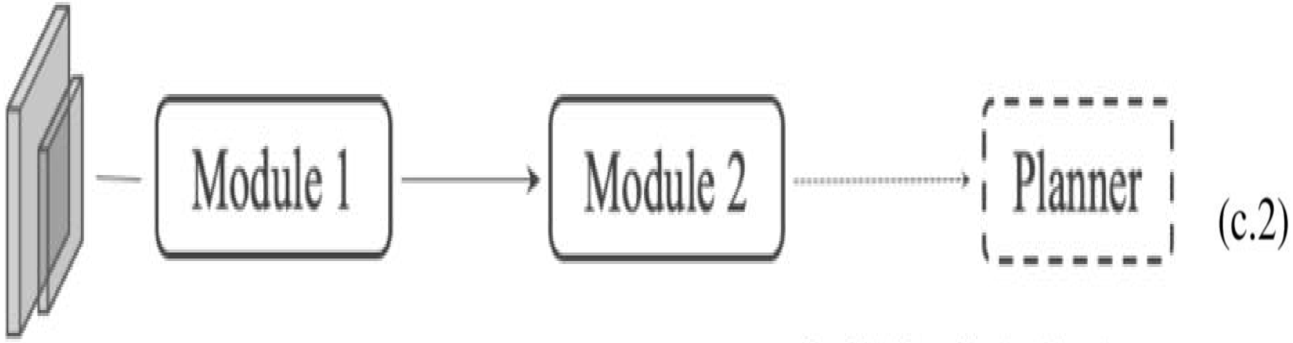
(c.1) Vanilla Solution

Design	Approach	Perception			Prediction		Plan
		Det.	Track	Map	Motion	Occ.	
(b)	NMP [101]	✓			✓		✓
	NEAT [19]			✓			✓
	BEVerse [105]	✓		✓		✓	
(c.1)	[14, 16, 78, 97]						✓

3. 端到端自动驾驶将感知、预测与规划的所有节点整合为整体。前序任务的选择与优先级应当为了规划而设定。

- 一种“白纸式（tabula - rasa）”的做法是直接预测规划轨迹
  → 无需对感知和预测进行显式监督；但在高度动态的城市场景中，在安全保障与可解释性方面仍不足
- 另一种方案是显式地感知周围物体、预测未来行为并规划安全动作
  → 以规划为导向，但已有工作不同程度上忽略了某些组件

1 Introduction



(c.2) Explicit Design

PnPNet† [57]	✓	✓		✓		
ViP3D† [30]	✓	✓		✓		
P3 [82]					✓	✓
MP3 [11]			✓		✓	✓
ST-P3 [38]			✓		✓	✓
LAV [15]	✓		✓	✓		✓

3. 端到端自动驾驶将感知、预测与规划的所有节点整合为整体。前序任务的选择与优先级应当为了规划而设定。

- 一种“白纸式（tabula-rasa）”的做法是直接预测规划轨迹

  →无需对感知和预测进行显式监督；但在高度动态的城市场景中，在安全保障与可解释性方面仍不足

- 另一种方案是显式地感知周围物体、预测未来行为并规划安全动作

  →以规划为导向，但已有工作不同程度上忽略了某些组件

1 Introduction

现代自动驾驶系统通常被描述为**按照顺序执行**的模块化任务，即**感知**、**预测**和**规划**。现有技术路线如下：

1. 大多数工业方案在车载芯片资源允许的情况下，**为每个任务独立部署单独模型**

 →简化了研发难度，但模块之间的隔离会带来**信息损失**、**误差积累**以及**特征不对齐**的风险

2. 将大量任务纳入一个多任务学习（MTL）范式，**通过将多个任务特定头插入共享特征提取器中**。这种方式在多个领域应用，包括通用视觉、自动驾驶，如 Transfuser、BEVverse，以及 Mobileye、Tesla、Nvidia 等工业产品

 →跨任务联合训练利用共享特征抽象；同时便于扩展到更多任务，并节省车载计算资源，但会导致不良的“**负迁移**”

3. 端到端自动驾驶将感知、预测与规划的所有节点整合为整体。前序任务的选择与优先级应当为了规划而设定。

- 一种“白纸式（tabula - rasa）”的做法是**直接预测规划轨迹**

 →无需对感知和预测进行显式监督；但在高度动态的城市场景中，在**安全保障**与**可解释性**方面仍不足

- 另一种方案是**显式地感知周围物体**、**预测未来行为**并**规划安全动作**

 →**以规划为导向**，但已有工作不同程度上**忽略了某些组件**

切入点：为了构建可靠且**以规划为导向**的自动驾驶系统，应**如何设计流程以更有利于规划**？**哪些前置任务是必需的**？

1 Introduction

现代自动驾驶系统通常被描述为**按照顺序执行**的模块化任务，即**感知**、**预测**和**规划**。现有技术路线如下：

1. 大多数工业方案在车载芯片资源允许的情况下，**为每个任务独立部署单独模型**

   →简化了研发难度，但模块之间的隔离会带来**信息损失**、**误差积累**以及**特征不对齐**的风险

2. 将大量任务纳入一个多任务学习（MTL）范式，**通过将多个任务特定头插入共享特征提取器中**。这种方式在多个领域应用，包括通用视觉、自动驾驶，如 Transfuser、BEVverse，以及 Mobileye、Tesla、Nvidia 等工业产品

   →跨任务联合训练利用共享特征抽象；同时便于扩展到更多任务，并节省车载计算资源，但会导致不良的“**负迁移**”

3. 端到端自动驾驶将感知、预测与规划的所有节点整合为整体。前序任务的选择与优先级应当为了规划而设定。

- 一种“白纸式（tabula - rasa）”的做法是**直接预测规划轨迹**

  →无需对感知和预测进行显式监督；但在高度动态的城市场景中，在**安全保障与可解释性**方面仍不足

- 另一种方案是**显式地感知周围物体**、**预测未来行为并规划安全动作**

  →**以规划为导向**，但已有工作不同程度上**忽略了某些组件**

切入点：为了构建可靠且**以规划为导向**的自动驾驶系统，应**如何设计流程**以更有利于规划？**哪些前置任务是必需的**？

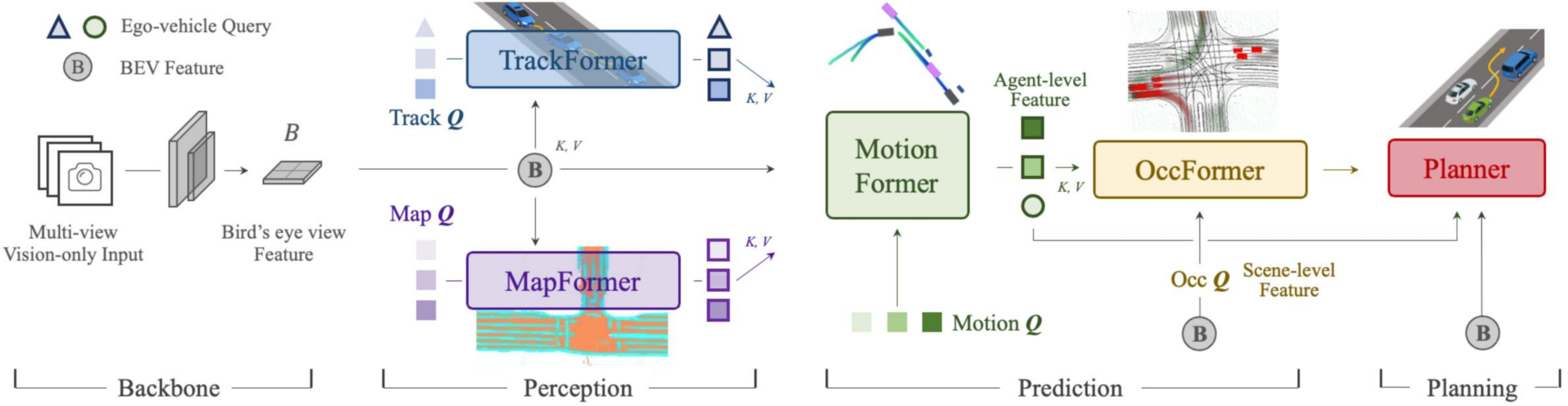
1 Introduction

三个贡献：

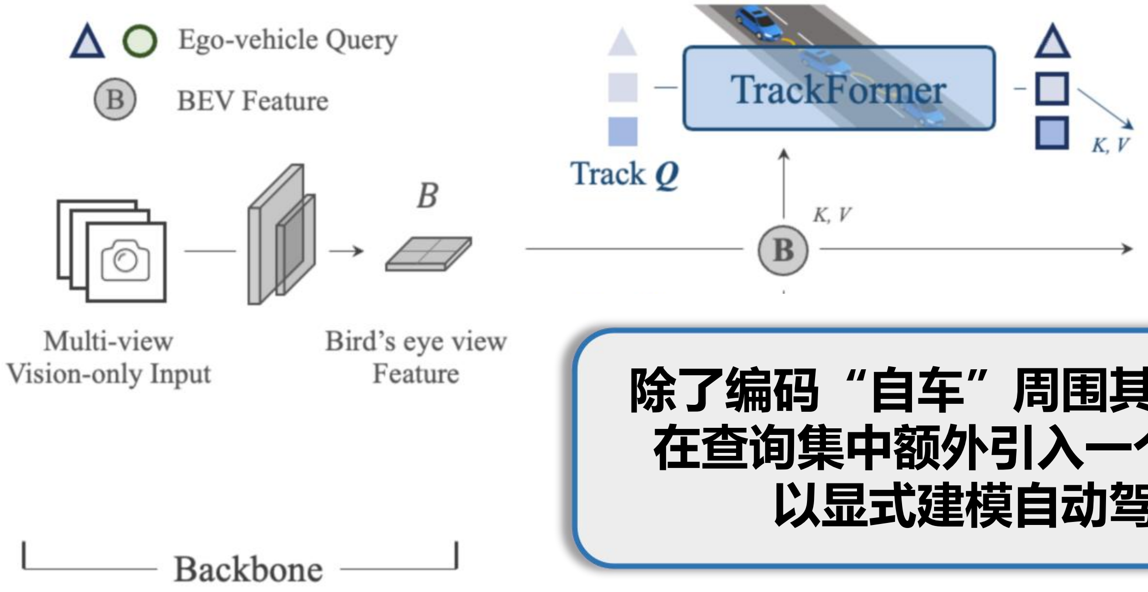
（a）提出了遵循规划导向理念的自动驾驶框架，并展示了有效任务协作的必要性，而非独立设计或简单多任务学习。

（b）提出 UniAD，一个综合性的端到端系统，涵盖广泛任务。其核心在于 query 作为连接所有节点的接口，使 UniAD 能够使用灵活的中间表示，并促进多任务之间的知识交换以服务规划。

（c）在真实场景的挑战性基准上实例化 UniAD，并通过大量消融实验验证其在各方面均优于既有最先进方法。



2 Method



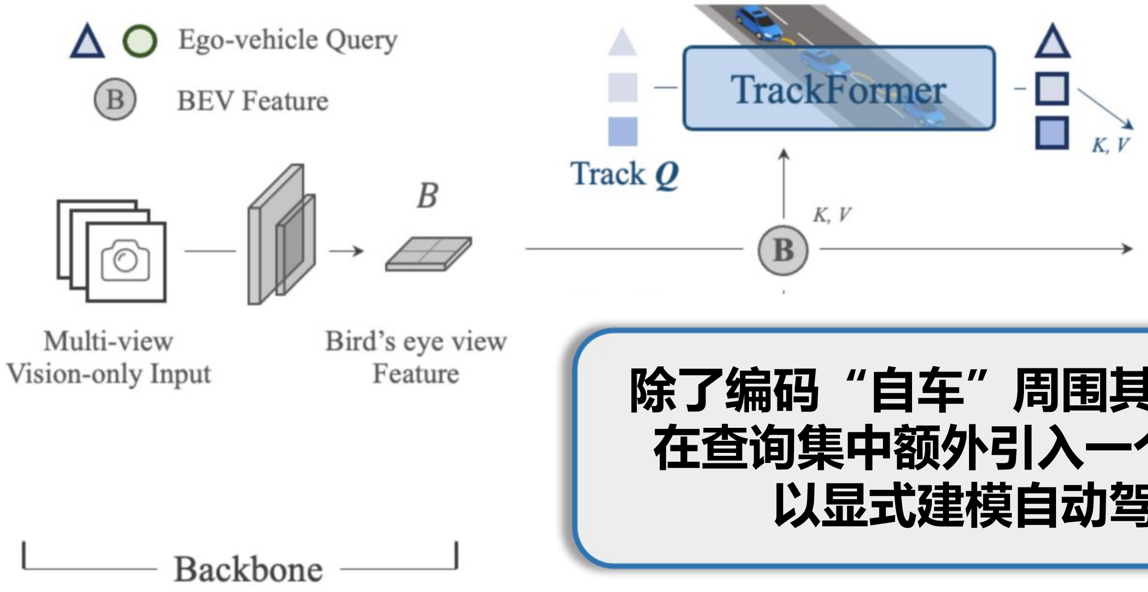
除了编码"自车"周围其他智能体的查询外在查询集中额外引入一个"自车查询"，以显式建模自动驾驶车辆本身

TrackFormer

1. detection queries（Deformable DETR）负责检测首次被感知到的智能体
2. track queries（MOTR）持续建模前几帧中检测到的智能体
3. detection queries 和 track queries 均通过 BEV 特征 B 来捕获智能体特征。
4. 当前帧的 track queries 会与先前记录的 track queries 在自注意力模块中交互，以聚合时序信息，直至对应的智能体完全消失（即在一段时间内未被跟踪）。

TrackFormer 包含 N 层，其最终输出状态 Q_A 为下游预测任务提供 N_a 个有效智能体的知识。

2 Method



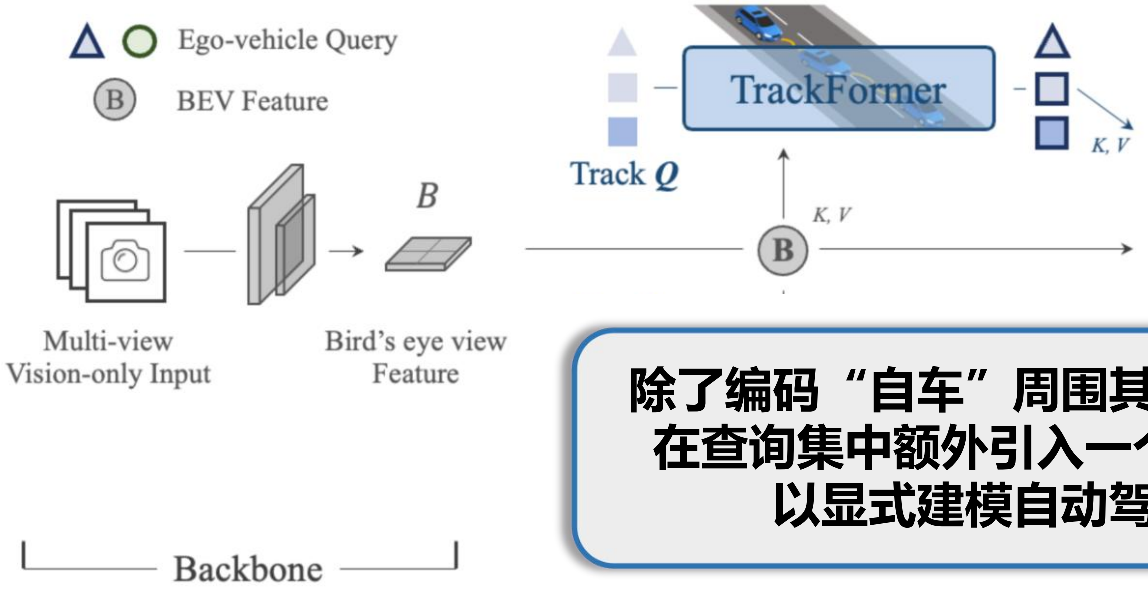
除了编码“自车”周围其他智能体的查询外在查询集中额外引入一个“自车查询”，以显式建模自动驾驶车辆本身

训练阶段：

1\. 在每个训练序列开始时，所有查询都被视为 detection queries，用于预测所有新出现的目标。detection queries 通过 Hungarian 算法与真实标注匹配。它们将被存储并通过查询交互模块更新，用于下一时间步以 track queries 的形式进行跟踪。

2\. 在下一时间步中， track queries 将依据对应的 trackID 与部分真实标注目标体直接匹配，而 detection queries 将与剩余的新生智能体进行匹配。为了稳定训练，预测框与真实框的 3D IoU 大于阈值 0.5 的查询匹配才会被存储和更新。

2 Method



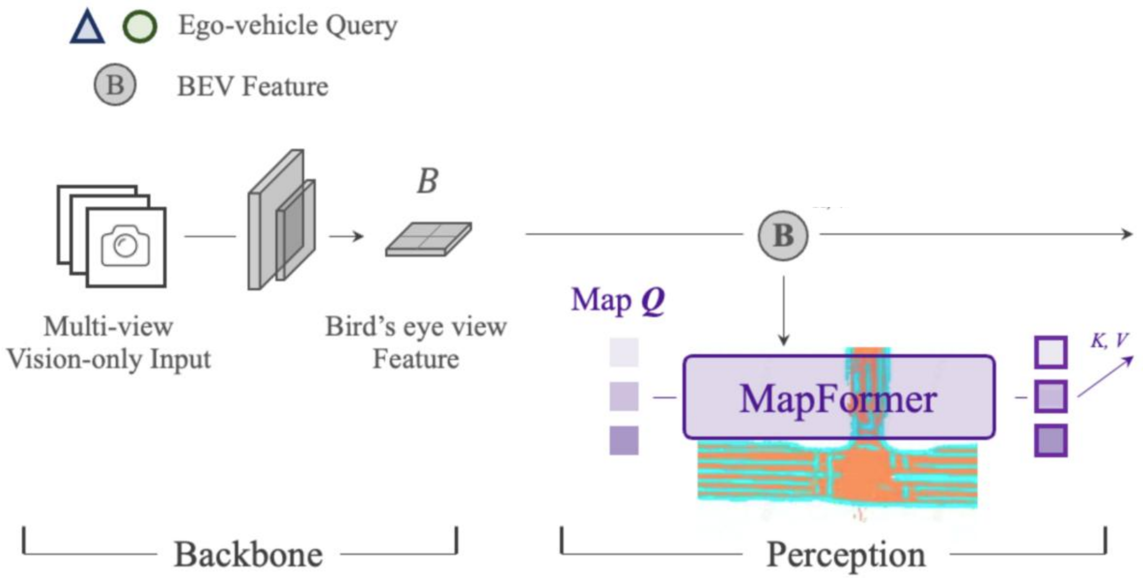
除了编码“自车”周围其他智能体的查询外在查询集中额外引入一个“自车查询”，以显式建模自动驾驶车辆本身

推理阶段：

1. 序列中的每一帧会顺序地送入网络
2. 由于无法获得真实值，使用分类得分（detection queries 阈值设为 0.4， track queries 阈值设为 0.35）来过滤查询，而不再采用 3D IoU 指标
3. 为了避免跟踪轨迹因短时遮挡而中断，在推理阶段为轨迹引入了生命周期机制。一个 track queries 只有在其对应的分类得分在连续一段时间内低于 0.35 时，才会被认为完全消失并被移除。

2

Method

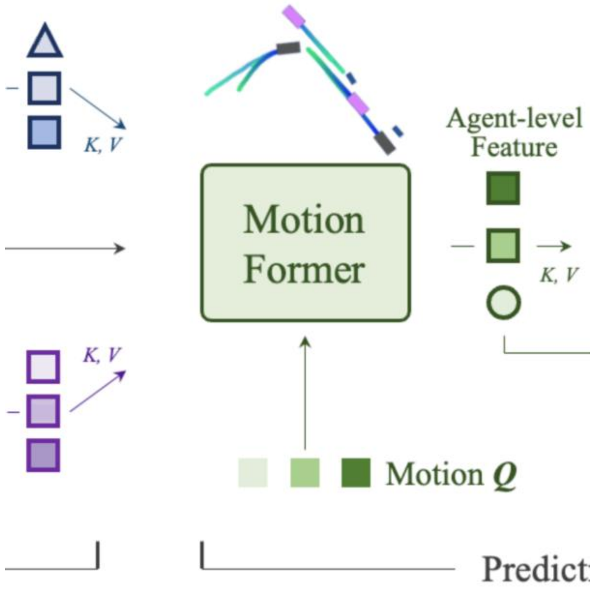


MapFormer：基于二维全景分割方法 Panoptic SegFormer。利用稀疏表示的道路元素作为 map queries，以编码位置与结构信息，辅助下游的运动预测

1. 将 map queries 分解为“thing queries”和“stuff queries”
2. “thing queries”用于建模实例级地图元素（车道线、边界、斑马线），并通过二分图匹配与真实值对应
3. “stuff queries”仅负责语义类元素（可行驶区域）并采用固定类别分配方式

MapFormer 堆叠 6 层位置解码器和 4 层 mask 解码器，选择最后一层中位置解码器输出之后的“thing queries”作为下游任务的地图查询 Q_M，送入 MotionFormer 以进行“智能体—地图交互”

2 Method



MotionFormer：

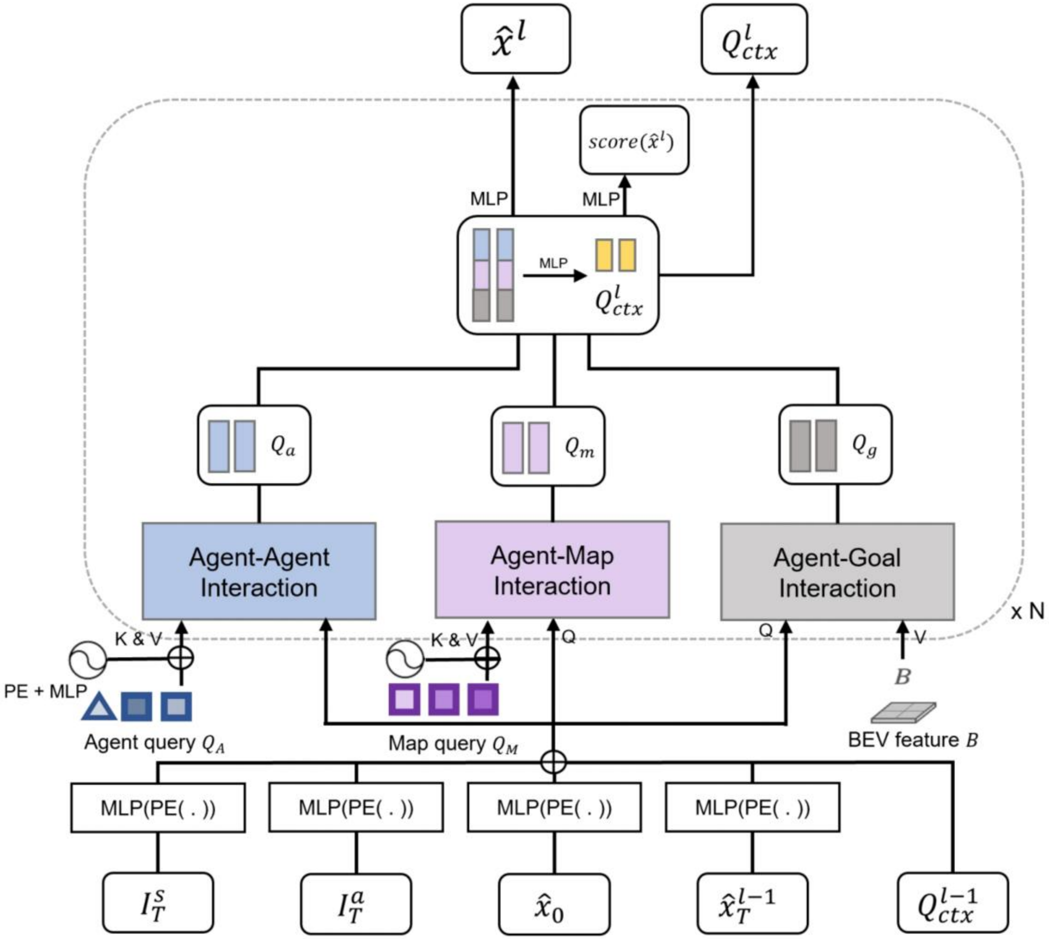
输入：基于来自 TrackFormer 的动态智能体查询 Q_A 与来自 MapFormer 的静态地图查询 Q_M

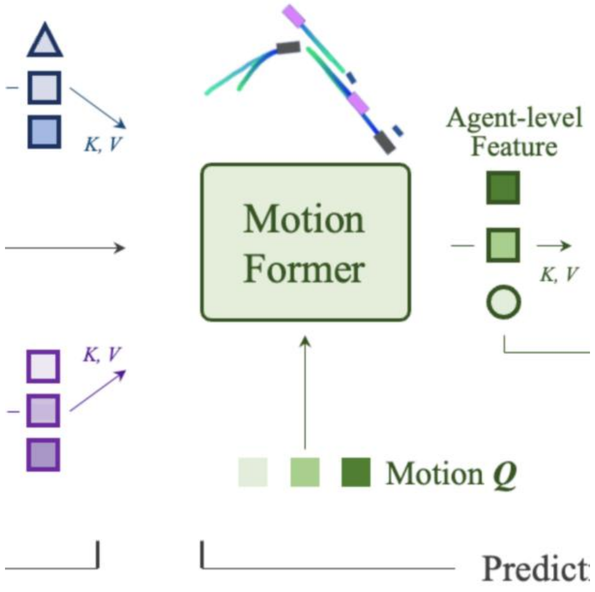
输出：预测所有智能体的多模态未来运动，即 top-k 个可能轨迹，表示为 $\{\hat{x}_{i,k} \in R^{T\times 2} \mid i=1,\ldots,N_a; k=1,\ldots,K\}$，其中 i 表示智能体索引，k 表示轨迹模态索引，T 为预测时间范围长度。

优势：

1. 通过一次前向传播生成整个场景的多智能体轨迹，大幅减少了将整个场景对齐到每个智能体坐标系的计算成本
2. 将 TrackFormer 中的自车查询传递给 MotionFormer，使自车能够在考虑未来动态的情况下与其他智能体交互。

2 Method

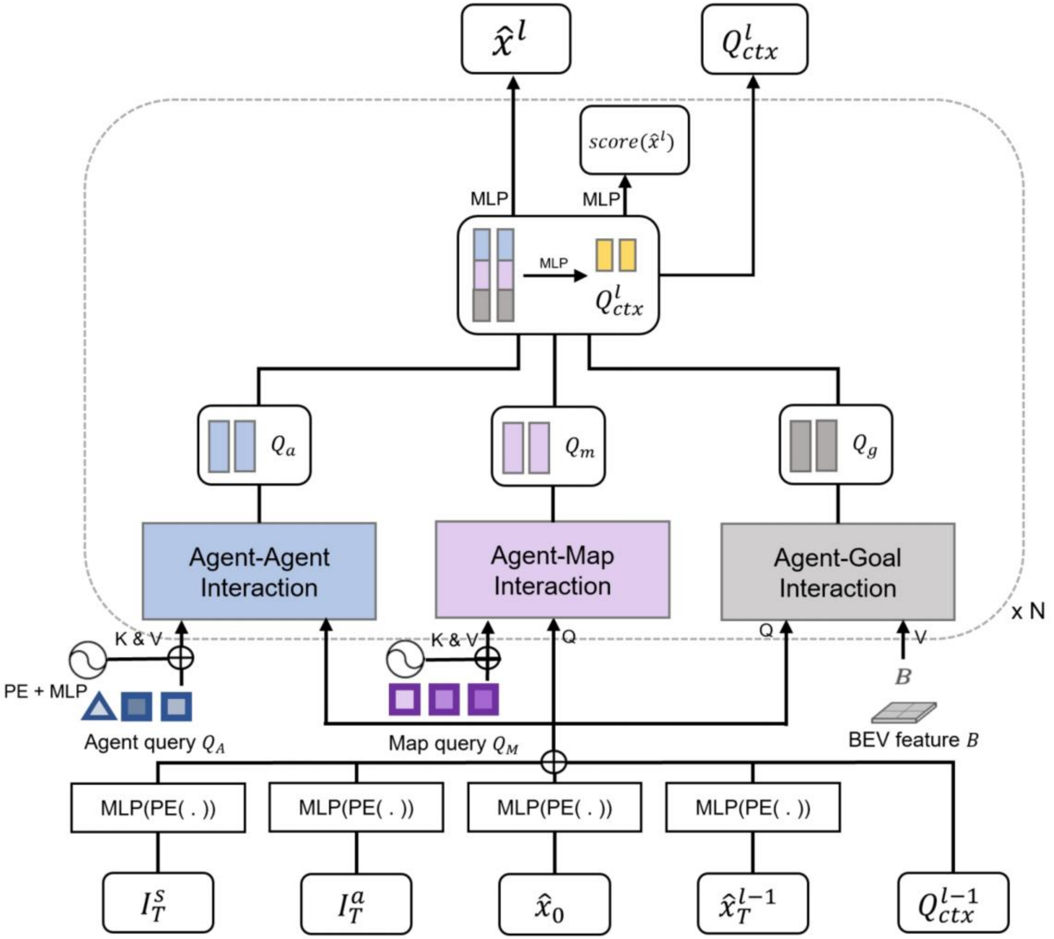




MotionFormer 每层的输入查询称为 motion queries，由两部分组成：

1. 前一层产生的查询上下文 Q_{ctx}
2. 查询位置 Q_{pos}。

2 Method



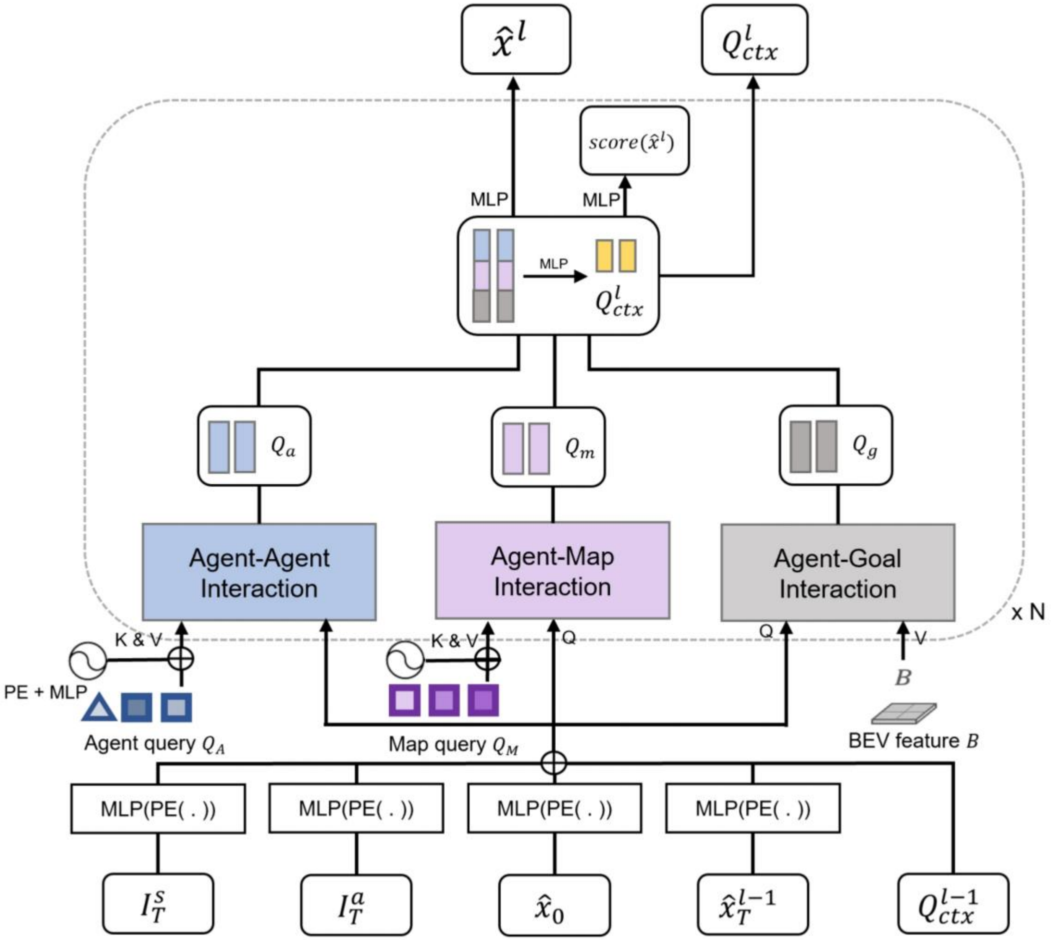
MotionFormer 使用 I_T^a、I_T^s、$\hat{x}_0$、$\hat{x}_T^{l-1} \in R^{K\times 2}$ 来嵌入查询位置 Q_{pos}

锚点在所有智能体的训练数据上通过 k - means聚类得到。

1. 将智能体级锚点 I_T^a 根据每个智能体当前的位置和朝向旋转、平移到全局坐标系中，得到场景级锚点记为 $I_{i,T}^s = R_i I_T^a + T_i$
2. 采用上一层预测的目标点 $\hat{x}_T^{l-1}$ 和智能体的当前位置 $\hat{x}_0$。
3. 应用 MLP 和正弦位置编码，将其汇总为查询位置 $Q_{pos} \in R^{K\times D}$

$$Q_{\text{pos}} = MLP(PE(I^s)) + MLP(PE(I^a)) + MLP(PE(\hat{x}_0)) + MLP(PE(\hat{x}_T^{l-1}))$$

2 Method



MotionFormer 由 N 层组成，每层捕获三种类型的交互：

“智能体—智能体”、“智能体—地图”以及“智能体—目标点”。

1. 对于每个运动查询 $Q_{i,k}$，其与其他智能体 Q_A 或地图元素 Q_M 的交互可表示为：

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

其中 MHCA、MHSA 分别表示多头交叉注意力和多头自注意力。

2. 由于目标点（即期望位置）对于细化预测轨迹也至关重要，

通过可变形注意力设计了“智能体—目标点”注意力机制：

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

其中 $\hat{x}_T^{l-1}$ 为上一层预测轨迹的终点。

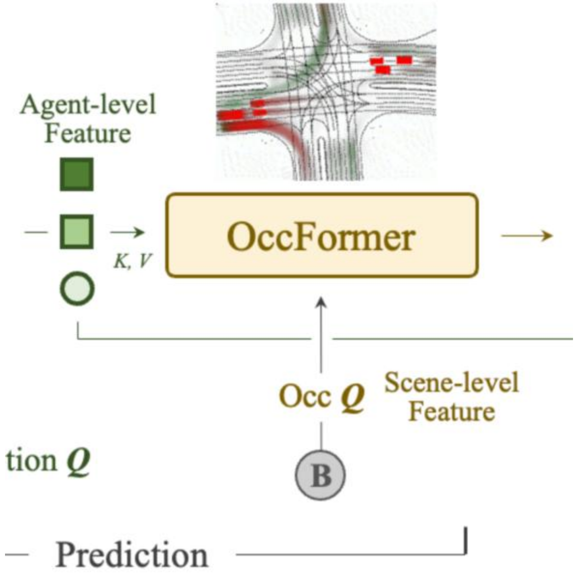
$\mathrm{DeformAttn}(q, r, x)$ 为可变形注意力模块，其输入为查询 q、参考点 r 和空间特征 x，

并在参考点周围的空间特征上执行稀疏注意力。

3. Q_a、Q_m、Q_g 会被拼接并传入多层感知机，形成查询上下文 Q_{ctx}。

随后，Q_{ctx} 会被送入下一层进一步细化，或在最后一层解码为预测结果。

2 Method



占用栅格地图是一种离散化的鸟瞰图表示，每个栅格表示其是否被占据的置信度，而占用预测任务旨在预测未来栅格图随时间的变化。此前的方法通常利用RNN结构，根据观测到的BEV特征在时间上展开预测未来。

→然而，这些方法往往需要高度手工设计的聚类后处理步骤来生成每个智能体的占用图，因为其大多不区分智能体，而是将整张BEV特征压缩进RNN隐状态中。由于缺乏智能体级的知识，这类方法难以全局地预测所有智能体的行为。为此，提出了OccFormer，以两种方式同时整合场景级与智能体级语义信息：

（1）在展开未来时间步时，一个密集场景特征通过注意力模块获取智能体级特征。

（2）通过智能体特征与密集场景特征的矩阵乘法，生成实例级占用图。

2 Method

OccFormer由 T_o 个顺序堆叠的模块组成，其中 T_o 表示预测时间范围。

每个模块的输入：以丰富的智能体特征 G^t 以及来自上一层的状态（密集特征） F^{t-1}

每个模块的输出：在结合实例级与场景级信息后生成时间步 t 的 F^t。

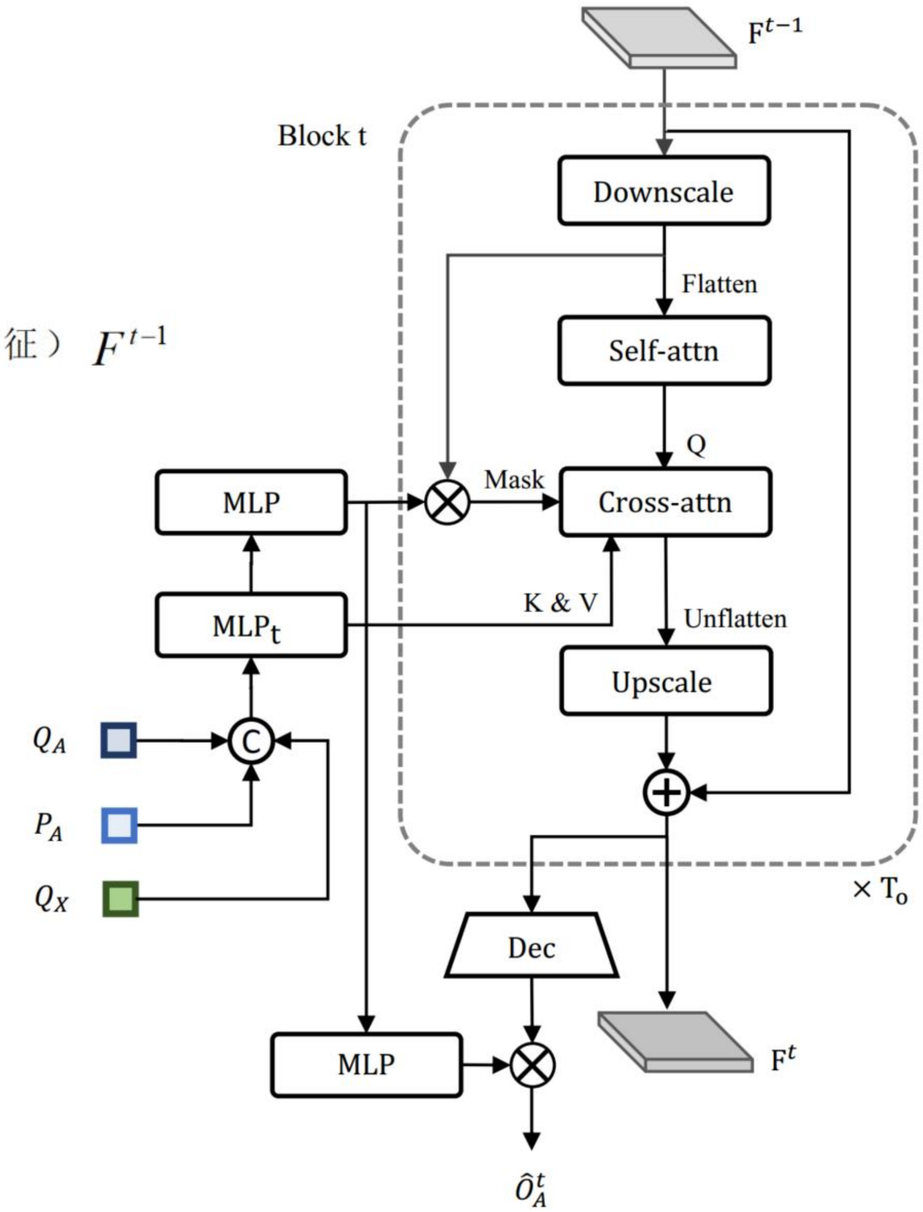
为了获得具有动态与空间先验的智能体特征 G^t，

将 MotionFormer 中的运动查询进行最大池化，得到 $Q_X \in R^{N_a \times D}$

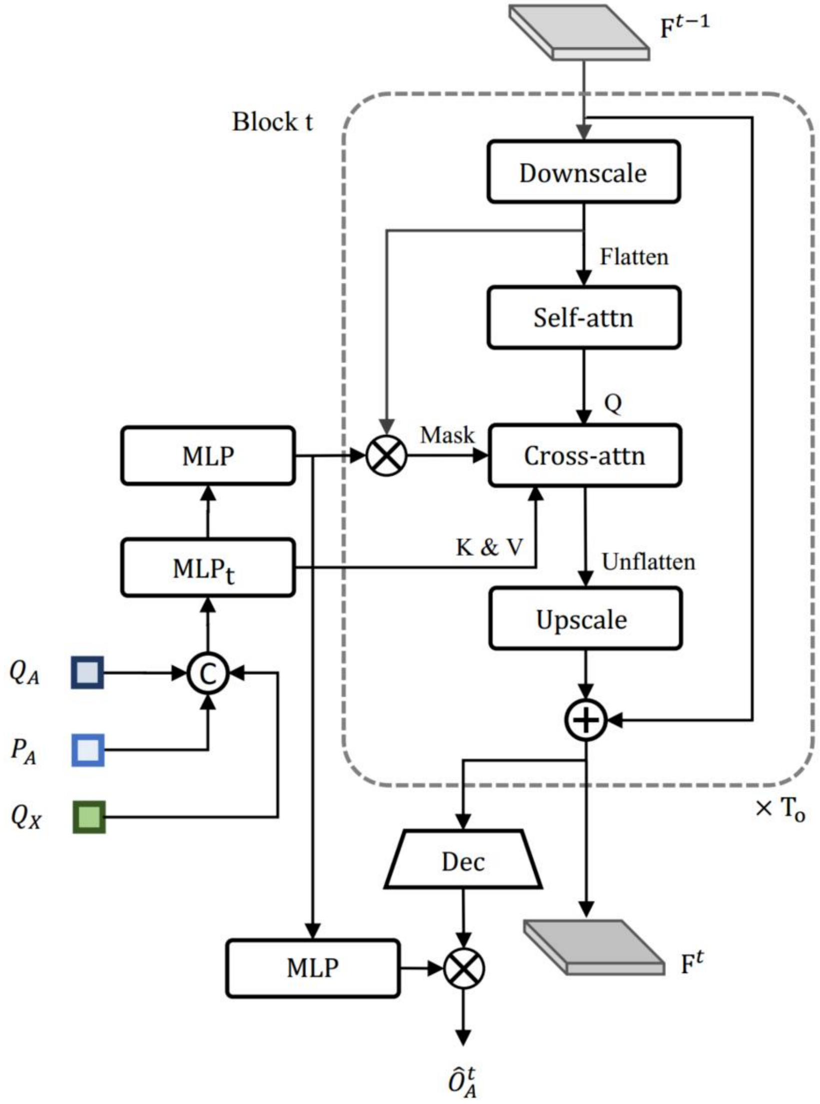
随后，将其与上游的跟踪查询 Q_A 以及当前位置嵌入 P_A

通过时间特定的 MLP 进行融合：

$$G^t = MLP([Q_A, P_A, Q_X]), t = 1, \ldots, T_o$$



2 Method



像素—智能体交互的注意力：

将密集特征 F_{ds}^t 作为查询，将智能体特征作为键和值，用于随时间更新密集特征

1. F_{ds}^t 首先通过自注意力层以建模远距离栅格之间的响应

2. 再通过交叉注意力建模智能体特征 G^t 与每个栅格特征之间的交互

3. 为了对齐像素与智能体的对应关系，在交叉注意力中加入了注意力掩码，使得每个像素仅能关注在时间步 t 占据该位置的智能体

注意力掩码 O_m^t 由额外的智能体级特征 M^t 与密集特征 F_{ds}^t 相乘生成

该智能体级特征称为掩码特征 $M^t = MLP(G^t)$。

4. 密集特征的更新过程为：

$$D_{ds}^t = MHCA(MHSA(F_{ds}^t), G^t, attn\_mask = O_m^t)$$

D_{ds}^t 被上采样至 B 的 $1/4$ 尺度。然后将 D_{ds}^t 与模块输入 F^{t-1} 相加作为残差连接，得到的特征 F^t 传递给下一模块。

实例级占用保持每个智能体的身份信息，通过矩阵乘法直接生成

为获得原始大小为 $H \times W$ 的 BEV 特征 B 的占用预测，

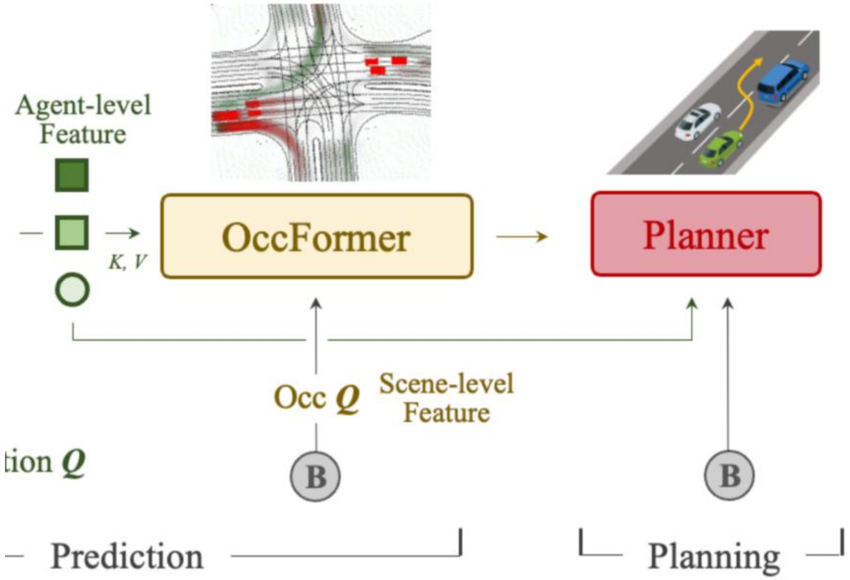
通过卷积解码器将场景级特征 F^t 上采样至 $F_{dec}^t \in R^{C\times H\times W}$

对于智能体级特征，将掩码特征 M^t 通过另一个 MLP 更新为占用特征 $U^t \in R^{N_a \times C}$

时间步 t 的最终实例级占用为：

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

2 Method



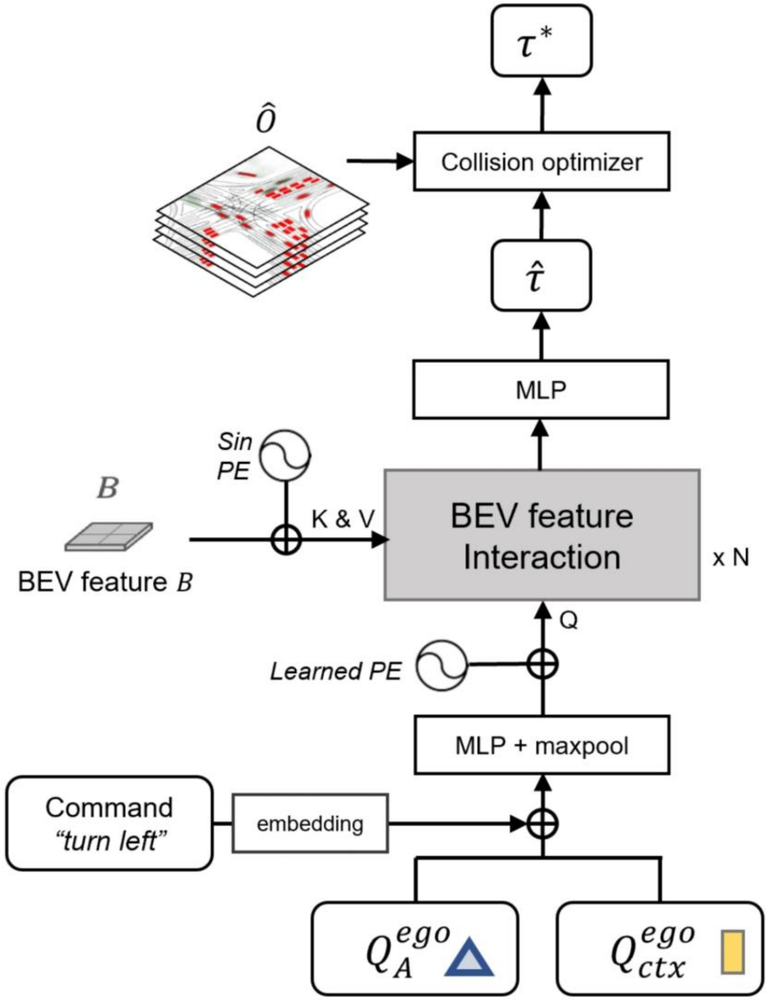
在无高精地图或预定义路径的情况下进行规划通常需要高层指令来指示行驶方向。

将原始导航信号（如左转、右转和直行）转换为三个可学习的嵌入，称为指令嵌入（command embeddings）。

1.将来自 MotionFormer 的自车查询与指令嵌入结合形成“规划查询（plan query）”。

2.将规划查询对 BEV 特征 B 进行注意力操作，使其感知周围环境，然后解码为未来路径点 $\hat{\tau}$。

2 Method



1.规划器获取来自跟踪和轨迹预测模块的自车查询，分别用蓝色三角形和黄色矩形表示

这两个查询与指令嵌入一起经过 MLP 编码，并通过最大池化聚合

2. BEV 特征交互模块采用标准 transformer 解码器层构建，并堆叠 $N=3$ 层。

为嵌入位置信息，将规划查询与可学习位置嵌入、BEV 特征与正弦位置嵌入，通过交叉注意力融合。

3.通过 MLP 回归规划轨迹 $\hat{\tau} \in R^{T_p \times 2}$。

4.为了进一步避免碰撞，仅在推理阶段基于牛顿法优化 $\hat{\tau}$，利用预测占用图 $\hat{O}$ 和规划轨迹 $\hat{\tau}$ 作为输入：

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

其中 $\hat{\tau}$ 为原始规划预测，τ^* 为优化后的规划结果，使代价函数 $f(\cdot)$ 最小。

$\hat{O}$ 为从 OccFormer 的实例级占用预测合并而成的经典二值占用图。

$$f\left(\tau, \hat{\tau}, \hat{O}\right) = \lambda_{coord}\left\|\tau, \hat{\tau}\right\|_2 + \lambda_{obs}\sum_t D\left(\tau_t, \hat{O}^t\right)$$

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

l_2 代价项将轨迹拉向原始预测

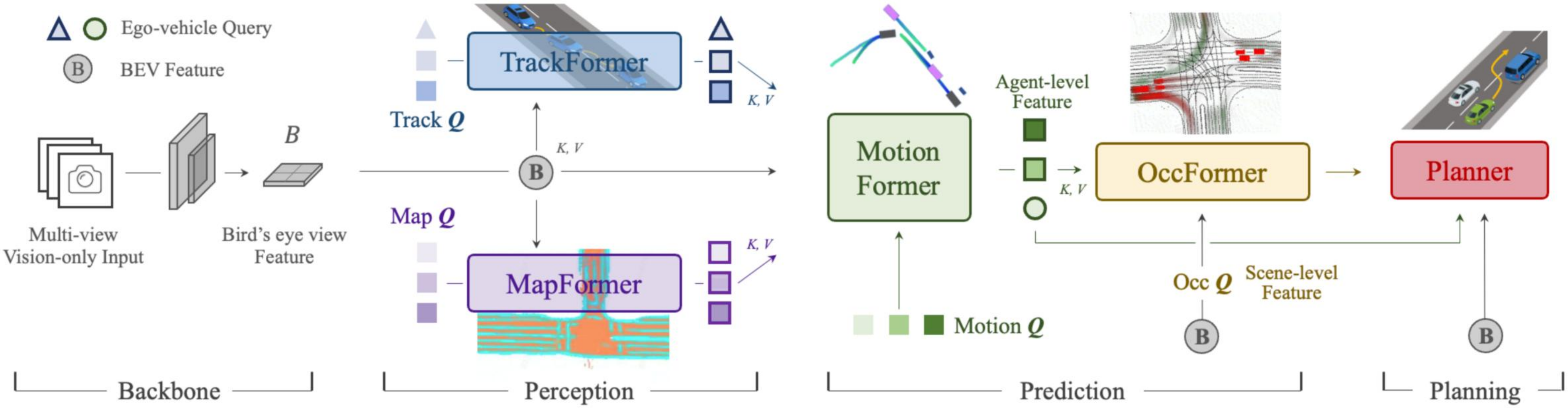
碰撞项 D 将其推离被占用的栅格

考虑约束区域 $S = \left\{(x,y) \mid \left\|(x,y) - \tau_t\right\|_2 < d, \hat{O}^t_{x,y} = 1\right\}$ 中的周围位置

2 Method

$$L_1 = L_{track} + L_{map}$$

$$L_2 = L_{track} + L_{map} + L_{motion} + L_{occ} + L_{plan}$$



3 Experiments

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*	✓	✓	✓	✓	✓	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
1	✓					0.348	1.333	791	-	-	-	-	-	-	-	-	-	-	-
2		✓				-	-	-	**0.305**	0.674	-	-	-	-	-	-	-	-	-
3	✓	✓				0.355	1.336	785	0.301	0.671	-	-	-	-	-	-	-	-	-
4			✓			-	-	-	-	-	0.815	1.224	0.182	-	-	-	-	-	-
5	✓		✓			0.360	1.350	919	-	-	0.751	1.109	0.162	-	-	-	-	-	-
6	✓	✓	✓			0.354	1.339	820	0.303	0.672	0.736(-9.7%)	1.066(-12.9%)	0.158	-	-	-	-	-	-
7				✓		-	-	-	-	-	-	-	-	60.5	37.0	52.4	29.8	-	-
8	✓			✓		0.360	**1.322**	809	-	-	-	-	-	62.1	38.4	52.2	32.1	-	-
9	✓	✓	✓	✓		0.359	1.359	1057	0.304	**0.675**	**0.710**(-3.5%)	**1.005**(-5.8%)	**0.146**	**62.3**	39.4	**53.1**	32.2	-	-
10					✓	-	-	-	-	-	-	-	-	-	-	-	-	1.131	0.773
11	✓	✓	✓		✓	**0.366**	1.337	889	0.303	0.672	0.741	1.077	0.157	-	-	-	-	1.014	0.717
12	✓	✓	✓	✓	✓	0.358	1.334	**641**	0.302	0.672	0.728	1.054	0.154	**62.3**	**39.5**	52.8	**32.3**	**1.004**	**0.430**

每一行表示当模型包含第二列 Modules 所列任务模块时的性能表现

第一行（ID - 0）作为基础的多任务基线用于比较

3 Experiments

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*	✓	✓	✓	✓	✓	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
1	✓					0.348	1.333	791	-	-	-	-	-	-	-	-	-	-	-
2		✓				-	-	-	**0.305**	0.674	-	-	-	-	-	-	-	-	-
3	✓	✓				0.355	1.336	785	0.301	0.671	-	-	-	-	-	-	-	-	-
4			✓			-	-	-	-	-	0.815	1.224	0.182	-	-	-	-	-	-
5	✓		✓			0.360	1.350	919	-	-	0.751	1.109	0.162	-	-	-	-	-	-
6	✓	✓	✓			0.354	1.339	820	0.303	0.672	0.736(-9.7%)	1.066(-12.9%)	0.158	-	-	-	-	-	-
7				✓		-	-	-	-	-	-	-	-	60.5	37.0	52.4	29.8	-	-
8	✓			✓		0.360	**1.322**	809	-	-	-	-	-	62.1	38.4	52.2	32.1	-	-
9	✓	✓	✓	✓		0.359	1.359	1057	0.304	**0.675**	**0.710**(-3.5%)	**1.005**(-5.8%)	**0.146**	**62.3**	39.4	**53.1**	32.2	-	-
10					✓	-	-	-	-	-	-	-	-	-	-	-	-	1.131	0.773
11	✓	✓	✓		✓	**0.366**	1.337	889	0.303	0.672	0.741	1.077	0.157	-	-	-	-	1.014	0.717
12	✓	✓	✓	✓	✓	0.358	1.334	**641**	0.302	0.672	0.728	1.054	0.154	**62.3**	**39.5**	52.8	**32.3**	**1.004**	**0.430**

1.实验 10—12 中，当轨迹预测与占用预测同时加入实验 12 时，规划的 L2 误差与碰撞率两个指标均达到最佳表现

→“轨迹预测与占用预测”对“规划”重要性

2.在实验 7—9 中，当轨迹预测与占用预测加入实验实验 9，两类任务的表现均获提升

→“预测模型中同时包含智能体与场景表征”的必要性

3 Experiments

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*	✓	✓	✓	✓	✓	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
1	✓					0.348	1.333	791	-	-	-	-	-	-	-	-	-	-	-
2		✓				-	-	-	**0.305**	0.674	-	-	-	-	-	-	-	-	-
3	✓	✓				0.355	1.336	785	0.301	0.671	-	-	-	-	-	-	-	-	-
4			✓			-	-	-	-	-	0.815	1.224	0.182	-	-	-	-	-	-
5	✓		✓			0.360	1.350	919	-	-	0.751	1.109	0.162	-	-	-	-	-	-
6	✓	✓	✓			0.354	1.339	820	0.303	0.672	0.736(-9.7%)	1.066(-12.9%)	0.158	-	-	-	-	-	-
7				✓		-	-	-	-	-	-	-	-	60.5	37.0	52.4	29.8	-	-
8	✓			✓		0.360	**1.322**	809	-	-	-	-	-	62.1	38.4	52.2	32.1	-	-
9	✓	✓	✓	✓		0.359	1.359	1057	0.304	**0.675**	**0.710**(-3.5%)	**1.005**(-5.8%)	**0.146**	**62.3**	39.4	**53.1**	32.2	-	-
10					✓	-	-	-	-	-	-	-	-	-	-	-	-	1.131	0.773
11	✓	✓	✓		✓	**0.366**	1.337	889	0.303	0.672	0.741	1.077	0.157	-	-	-	-	1.014	0.717
12	✓	✓	✓	✓	✓	0.358	1.334	**641**	0.302	0.672	0.728	1.054	0.154	**62.3**	**39.5**	52.8	**32.3**	**1.004**	**0.430**

3.在实验 4—6 中，同时引入跟踪与建图节点可显著提升预测表现。

→“同时引入跟踪与建图”的必要性

4.在实验 1—3 中，结果表明将多个感知子任务联合训练可获得与单任务相当的性能。

3

Experiments

Method	minADE(m)↓	minFDE(m)↓	MR↓	EPA↑
PnPNet† [57]	1.15	1.95	0.226	0.222
ViP3D [30]	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**

Table 5. **Motion forecasting.** UniAD remarkably outperforms previous vision-based end-to-end methods. We also report two settings of modeling vehicles with constant positions or velocities as comparisons. †: Reimplemented with BEVFormer.

Method	AMOTA↑	AMOTP↓	Recall↑	IDS↓
Immortal Tracker† [93]	0.378	1.119	0.478	936
ViP3D [30]	0.217	1.625	0.363	-
QD3DT [36]	0.242	1.518	0.399	-
MUTR3D [104]	0.294	1.498	0.427	3822
UniAD	**0.359**	**1.320**	**0.467**	**906**

Table 3. **Multi-object tracking.** UniAD outperforms previous end-to-end MOT techniques (with image inputs only) on all metrics. †: Tracking-by-detection method with post-association, reimplemented with BEVFormer for a fair comparison.

Method	IoU-n.↑	IoU-f.↑	VPQ-n.↑	VPQ-f.↑
FIERY [35]	59.4	36.7	50.2	29.9
StretchBEV [1]	55.5	37.1	46.0	29.0
ST-P3 [38]	-	38.9	-	32.1
BEVerse† [105]	61.4	**40.9**	54.3	**36.1**
UniAD	**63.4**	40.2	**54.7**	33.5

Table 6. **Occupancy prediction.** UniAD gets significant improvement in nearby areas, which are more critical for planning. "n." and "f." indicates near ($30 \times 30m$) and far ($50 \times 50m$) evaluation ranges respectively. †: Trained with heavy augmentations.

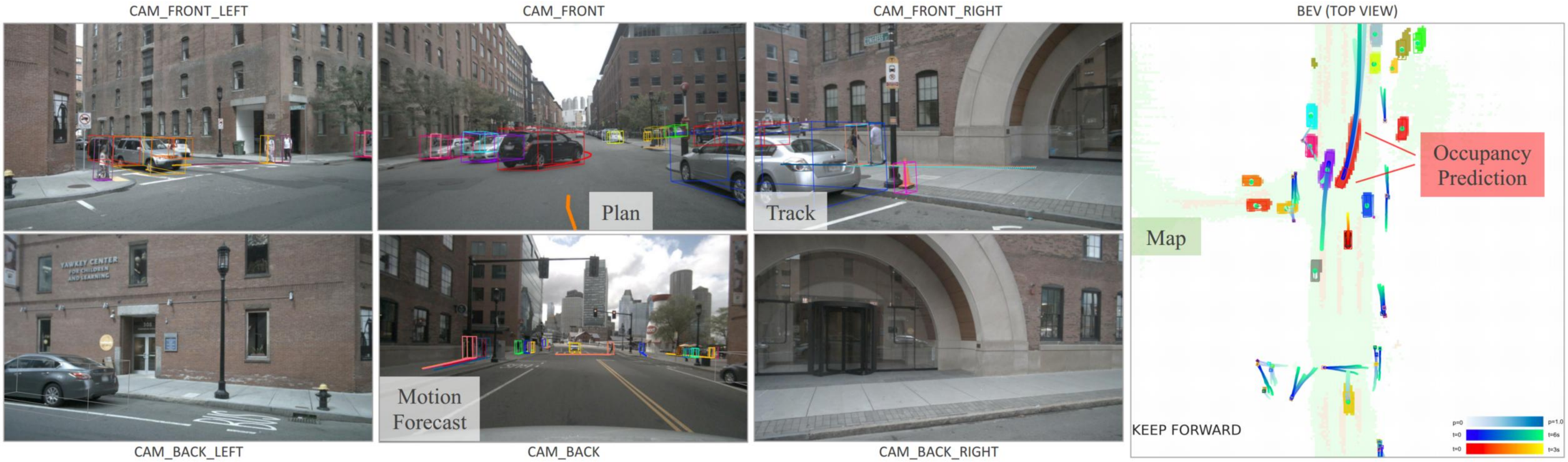
Method	Lanes↑	Drivable↑	Divider↑	Crossing↑
VPN [72]	18.0	76.0	-	-
LSS [76]	18.3	73.9	-	-
BEVFormer [55]	23.9	**77.5**	-	-
BEVerse† [105]	-	-	**30.6**	**17.2**
UniAD	**31.3**	69.1	25.7	13.8

Table 4. **Online mapping.** UniAD achieves competitive performance against state-of-the-art perception-oriented methods, with comprehensive road semantics. We report segmentation IoU (%). †: Reimplemented with BEVFormer.

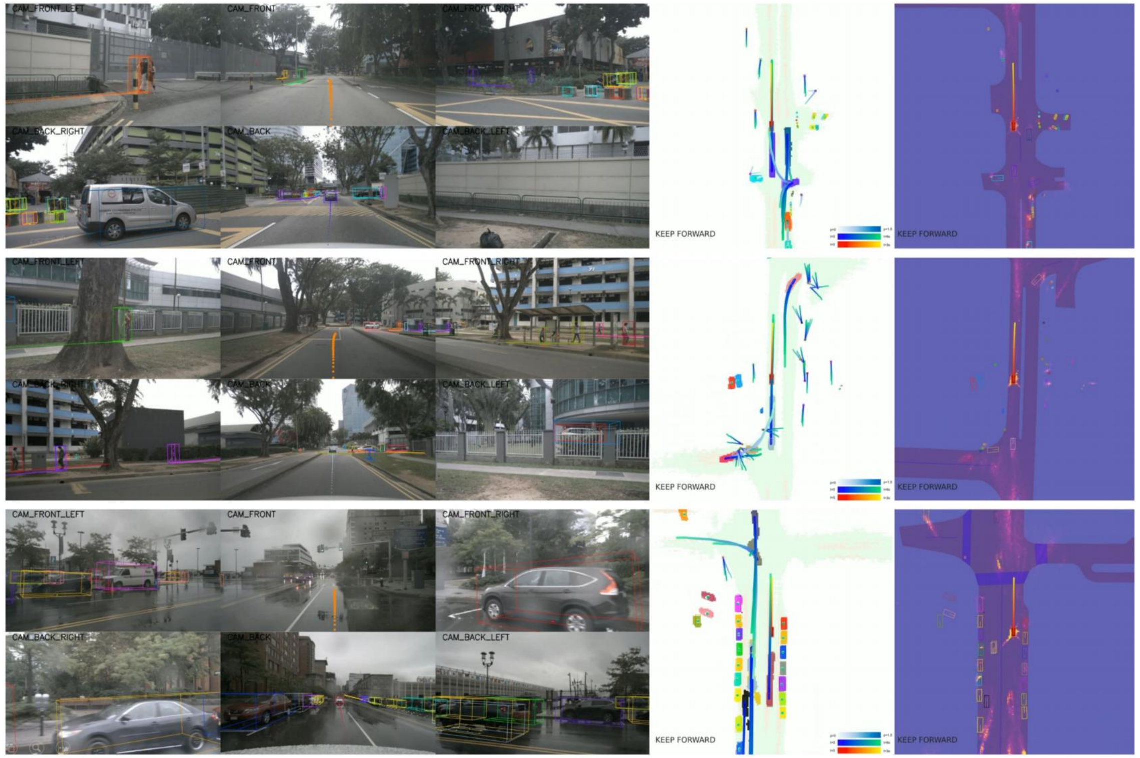
Method	L2(m)↓				Col. Rate(%)↓			
	$1s$	$2s$	$3s$	Avg.	$1s$	$2s$	$3s$	Avg.
NMP† [101]	-	-	2.31	-	-	-	1.92	-
SA-NMP† [101]	-	-	2.05	-	-	-	1.59	-
FF† [37]	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43
EO† [47]	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33
ST-P3 [38]	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD	**0.48**	**0.96**	**1.65**	**1.03**	**0.05**	**0.17**	**0.71**	**0.31**

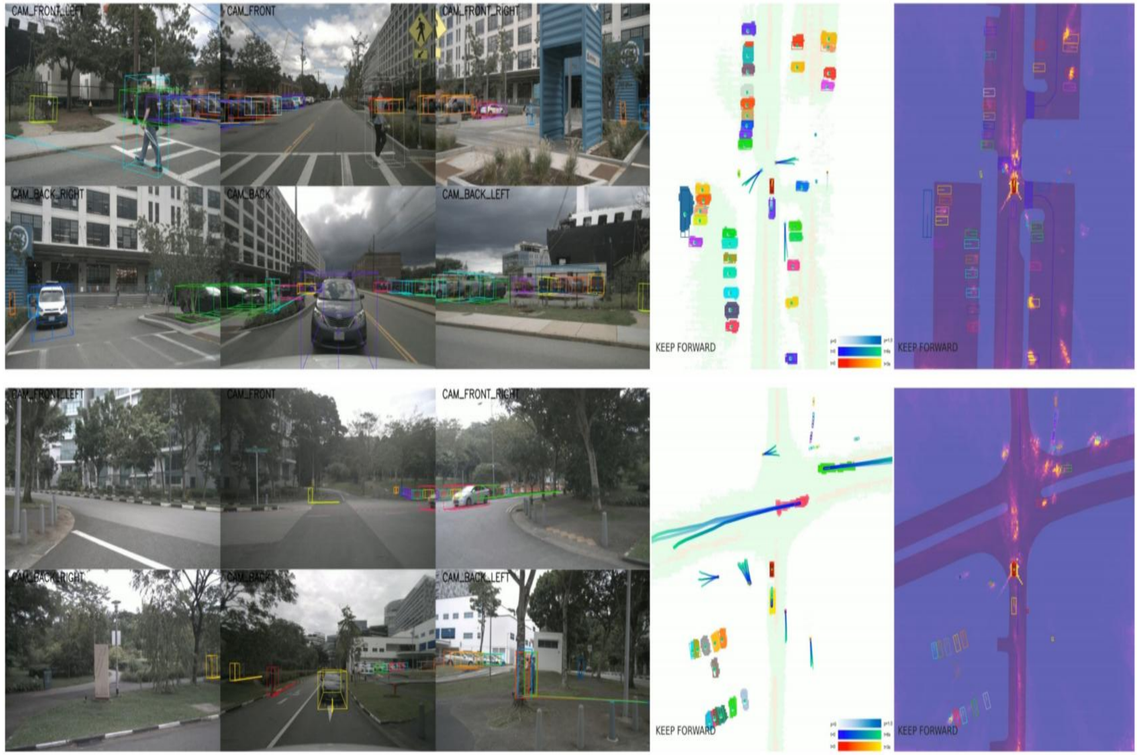
Table 7. **Planning.** UniAD achieves the lowest L2 error and collision rate in all time intervals and even outperforms LiDAR-based methods (†) in most cases, verifying the safety of our system.

3 Experiments



3 Experiments





Thanks!