端到端自动驾驶算法详解

目 录

CONTENTS

UniAD- 出处

Planning-oriented Autonomous Driving

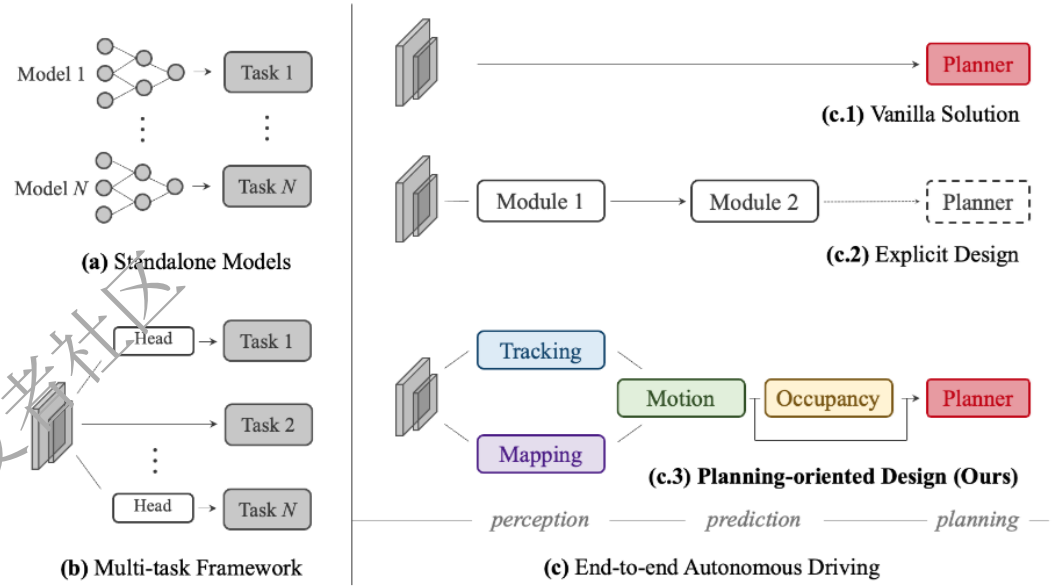
Yihan Hu[1,2*], Jiazhi Yang[1*], Li Chen[1*†], Keyu Li[1*], Chonghao Sima[1], Xizhou Zhu[3,1]
Siqi Chai[2], Senyao Du[2], Tianwei Lin[2], Wenhai Wang[1], Lewei Lu[3], Xiaosong Jia[1]
Qiang Liu[2], Jifeng Dai[1], Yu Qiao[1], Hongyang Li[1†]

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

UniAD- 背景

--Planning-oriented Autonomous Driving

(a) 针对不同的任务部署单独的模型。
(b) 多任务学习方案共享主干网络。每个任务有自己的专用分支网络，这些分支从主干网络中提取共享特征，并进一步处理这些特征以完成特定任务。可能导致负迁移问题。
(c) 端到端范式将感知和预测模块结合在一起。
  (c.1) 对规划进行直接优化。
  (c.2) 设计中带有部分组件。
  (c.3) 以规划为导向，并正确的组织先前的任务以促进规划。

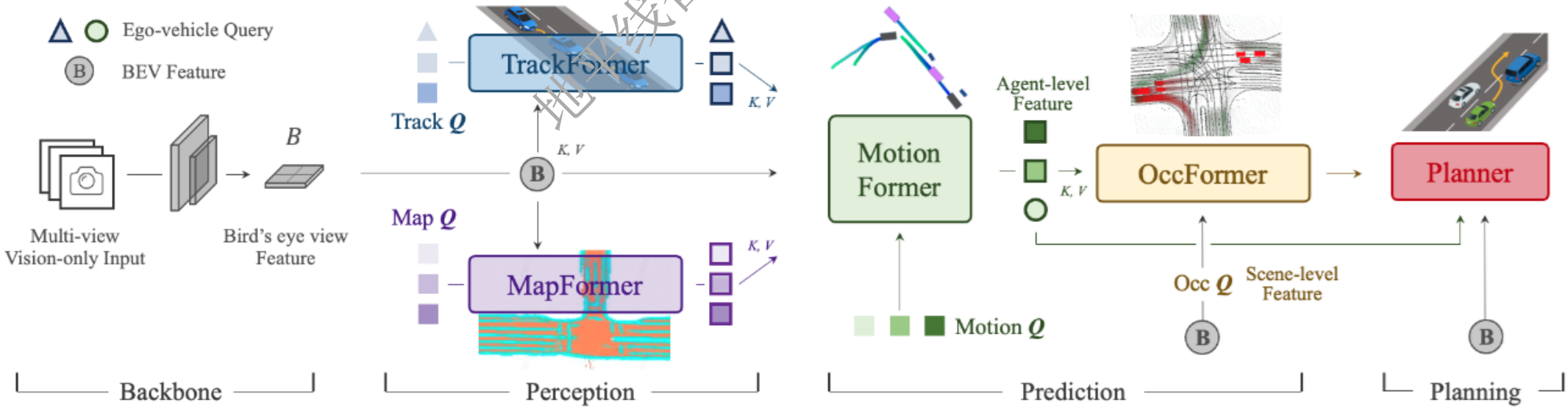


- UniAD是以规划为导向设计的，连接所有节点是基于query设计的。
- 与经典的bounding box表示相比，query受益于更大的感受域，可减少上游预测的符合误差。
- Query可以灵活地对各种交互进行建模和编码，例如多个物体之间的关系。
- 第一个全面研究自动驾驶领域感知、预测和规划等多种任务联合合作的作品。

Design	Approach	Perception			Prediction		Plan
		Det.	Track	Map	Motion	Occ.	
(b)	NMP [101]	✓			✓		✓
	NEAT [19]			✓			✓
	BEVerse [105]	✓		✓		✓	
(c.1)	[14, 16, 78, 97]						✓
(c.2)	PnPNet† [57]	✓	✓		✓		
	ViP3D† [30]	✓	✓		✓		
	P3 [82]					✓	✓
	MP3 [11]			✓		✓	✓
	ST-P3 [38]			✓		✓	✓
	LAV [15]	✓		✓	✓		✓
(c.3)	**UniAD** (ours)	✓	✓	✓	✓	✓	✓

UniAD- 流程介绍

- 基于transformer解码器，query起到连接pipeline的作用。
- 多视角图像通过特征提取器转化为图像特征，再通过BEV编码器转换为BEV的特征B。
- TrackFormer中，从B中的query物体的信息来进行检测和跟踪。
- MapFormer中，从B中query道路元素信息，并执行地图的全景分割，例如分割出人行道，红绿灯。
- MotionFormer使用交互信息和特征值，预测每个物体未来轨迹。
- 自车query，以明确地对自车进行建模，用于后续规划。
- OccFormer使用B作为query，以及motion信息，预测物体未来occupancy的情况。
- Planner使用来自MotionFormer的自车query来预测规划结果，并避开OccFormer预测的占用区域来避免碰撞。



UniAD- 模块详解

- TrackFormer进行检测与多目标追踪，通过Track query模型追踪物体的整个生命周期。包括：
  - 检测query：检测新出现的物体。
  - 跟踪query：通过在当前帧中的跟踪query与先前记录的query交互，实现对已检测物体的持续跟踪。
  - 自车query：建模自车，预测自车未来轨迹。

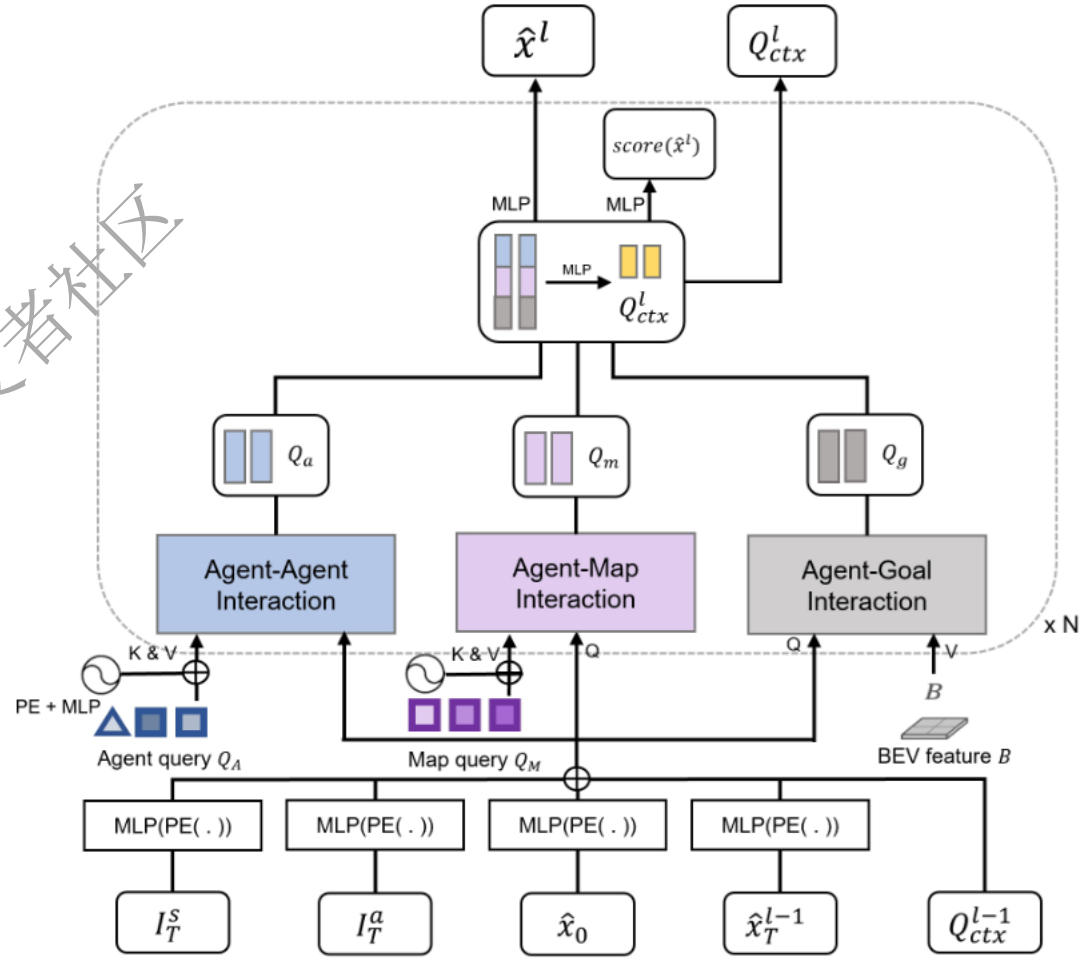
- MapFormer使用一组Map query表示地图中的不同元素，如人行道、车道线、可行驶区域等，以帮助下游任务学习环境信息。Map query被传送至MotionFormer进行与地图元素的交互。

- MotionFormer以物体特征和地图特征为输入，输出场景中所有agent在多种模态下的未来轨迹。通过一次向前传播即可输出所有智能体的未来轨迹，节省了以agent为中心的方法中每步对齐坐标空间的计算消耗。为持续建模自车运动信息，利用TrackFormer中的自车查询query向量学习自车未来轨迹。MotionFormer由多层cross-attention模块组成，每层包含三种不同的注意力，即Agent-Agent；Agent-Map；Agent-Goal。

UniAD- MotionFormer

- 由N个堆叠的交互模块组成，每个模块内进行Agent-Agent；Agent-Map；Agent-Goal的关系建模。 Agent-Agent和Agent-Map使用标准的Transformer解码器层， Agent-Goal为可变形的交叉注意力模块

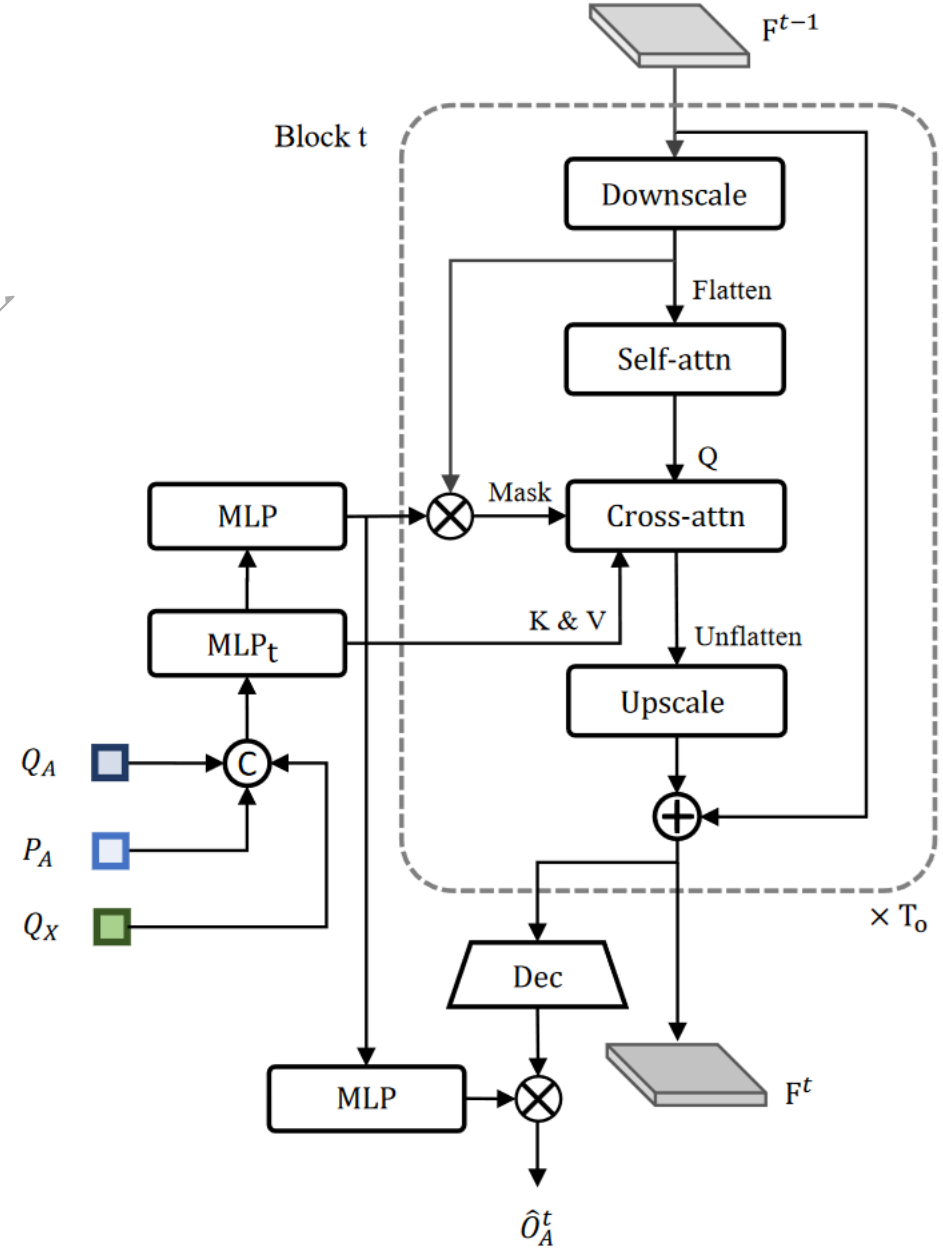
- I_T^s ： 场景级锚点位置
- I_T^a ： 物体级锚点位置
- $\hat{x}_0$ ： 物体的当前为主
- $\hat{x}_T^{l-1}$ 上一层预测的未来目标位置



UniAD- OccFormer

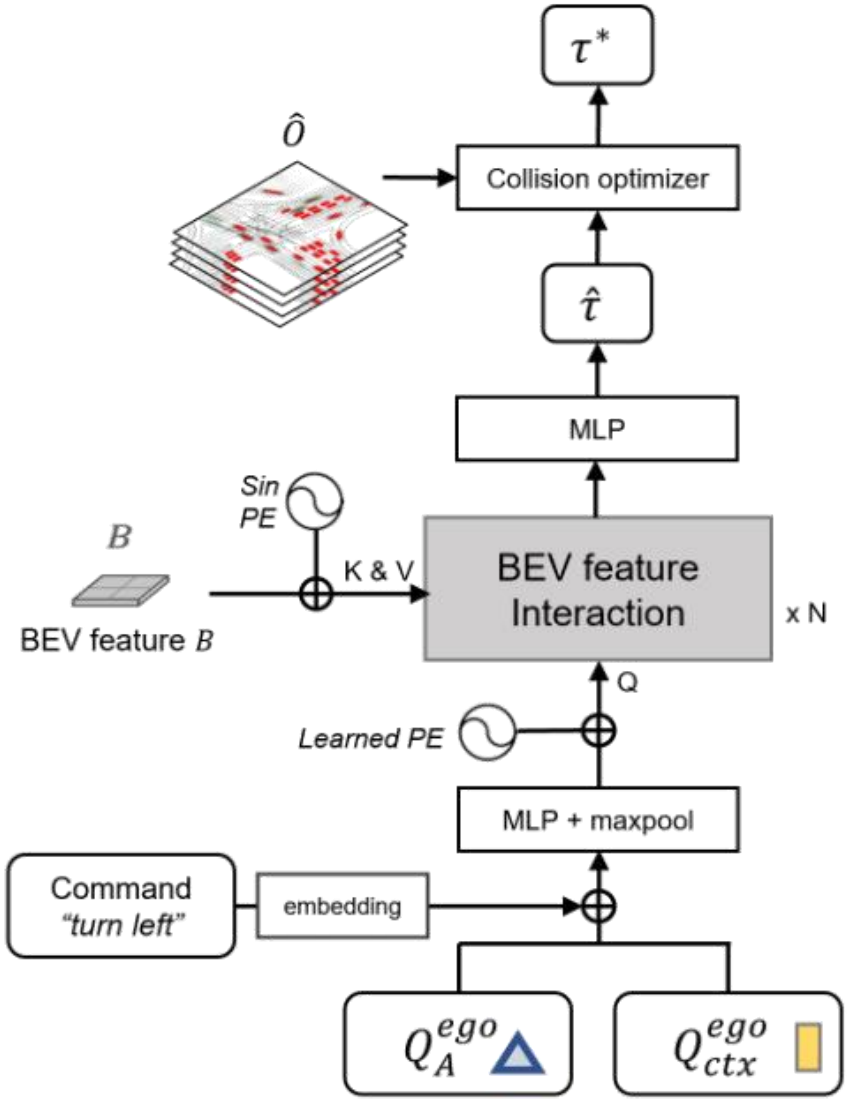
--Planning-oriented Autonomous Driving

- 占用栅格图是一种离散化的BEV 表示形式，其中每个格子代表的值代表当前位置是否被物体占用。占用栅格预测任务是指预测未来多步的占用栅格图，即未来 BEV 的占用情况。
- OccFormer 中利用注意力机制，将场景中密集的各栅格表示为查询向量，将物体特征表示为键与值。通过多层 Transformer 的解码器，查询向量将多次更新，用于表示未来时序的 BEV 特征图。为了更好地对齐物体与各栅格的位置关系，本文引入了一个基于占用栅格的注意力掩码，该掩码使得注意力计算只在位置对应的栅格-物体特征之间进行。



UniAD- Planner

- 为了规划自车未来的运动轨迹，将MotionFormer更新后的自车查询向量与BEV特征进行注意力机制交互，让自车query感知整个BEV环境，隐式地学习周围环境和其他智能体。为了避免与周围车的碰撞，利用占用栅格预测模块的输出对自车路径进行优化，避免未来可能有物体占用的区域。
- 下层分别来自跟踪模块和轨迹预测模块的自车query特征，与命令嵌入。通过MLP层进行编码，最大池化层选择最显著的模态特征。用自车作为query，BEV特征B作为k和v进入transformer decoder。最终通过OCC的碰撞避免优化器对预测轨迹再此优化，最终达到更安全的路径规划。



UniAD- 实验结果

- 本文在具有挑战性的nuScenes数据集基准上实例化UniAD。

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 previ-

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 perfor-

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

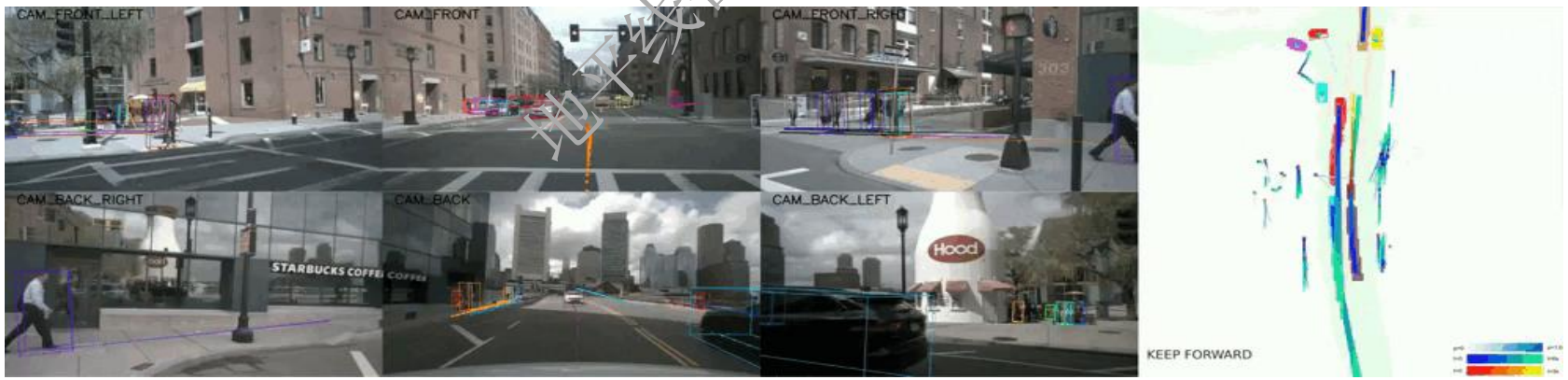
Table 6. **Occupancy prediction.** UniAD gets significant improve-

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 colli-

UniAD- 可视化

UniAD在目标检测阶段未检测出一个隐蔽的物体，但在规划时仍有效地施加了注意力到相关区域。证明了UniAD具备从上游模块错误中恢复的能力。



晴天直行，UniAD 可以感知左前方等待的黑色车辆，预测其未来轨迹（即将左转驶入自车的车道），并立即减速以进行避让。

UniAD- 具有挑战

- 部署困难。
- 采用开环测试，无闭环测试结果。
- 可解释性不够直观。
- 长尾场景测试效果未知。

DriveVLM- 出处

DRIVEVLM: The Convergence of Autonomous Driving and Large Vision-Language Models

Xiaoyu Tian[1*] Junru Gu[1*] Bailin Li[2*] Yicheng Liu[1] Chenxu Hu[1]
Yang Wang[2] Kun Zhan[2] Peng Jia[2] Xianpeng Lang[2] Hang Zhao[1†]

[1]IIIS, Tsinghua University [2]Li Auto

DriveVLM- 背景

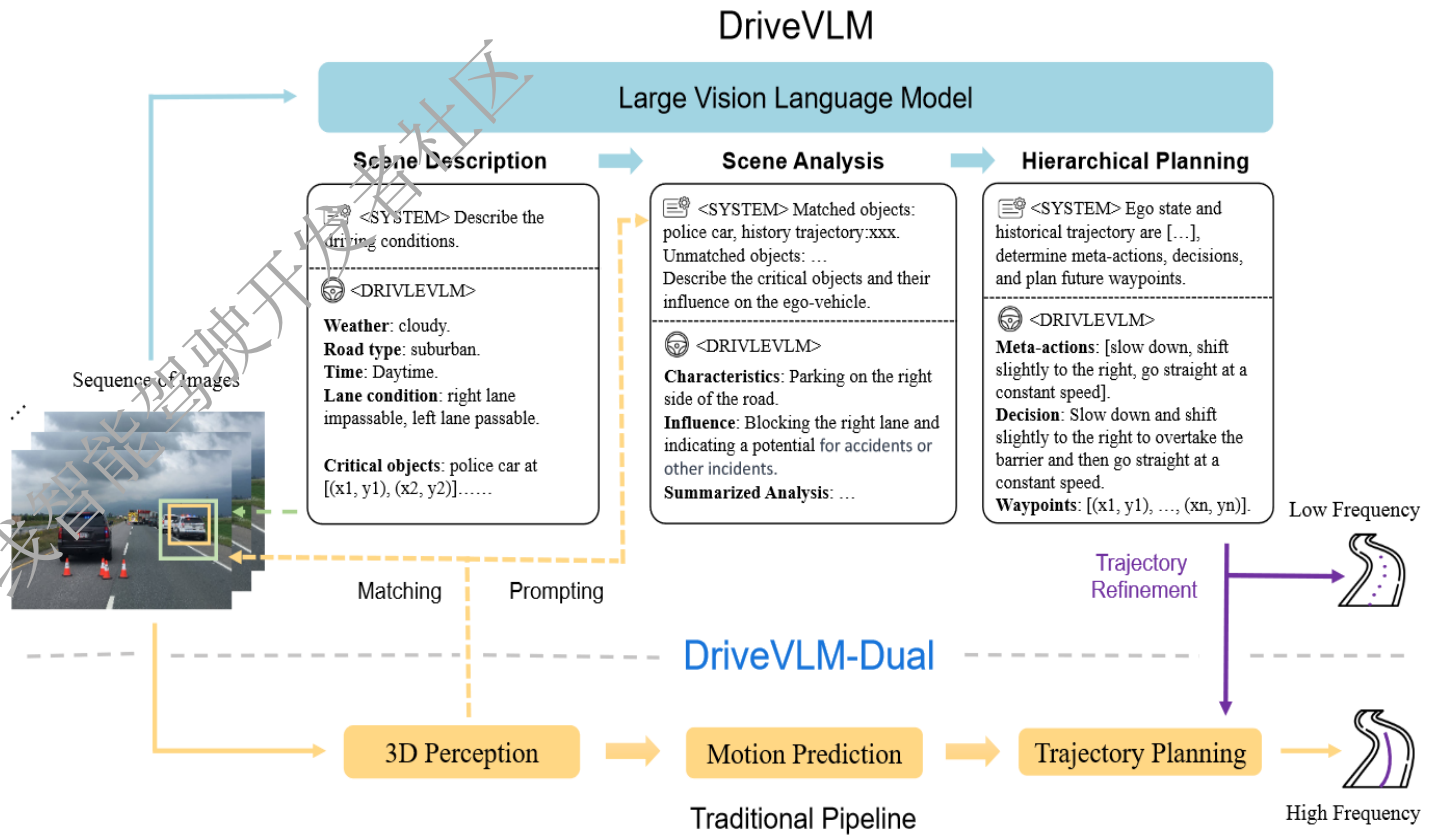
- 常规端到端自动驾驶在处理常规场景时表现尚可，但在面对复杂场景或者长尾场景时会遇到较大的挑战。
- 常规模块的设计缺乏“场景理解”能力所导致的，比如感知模块常常只是检测识别常见物体，忽略了长尾物体及其特性的识别。

- 提出了一种将大视觉语言模型用于自动驾驶场景的方法DriveVLM。
- 提出了一种大模型与传统自动驾驶模块相结合的方法DriveVLM-Dual。
- 提出了一套挖掘复杂和长尾驾驶场景的数据挖掘流程，并以此构建了多样化地SUP-AD数据集。

DriveVLM- 详解DriveVLM

--DRIVEVLM: The Convergence of Autonomous Driving and Large Vision-Language Models

- 场景描述：环境描述和关键物体识别。
- 场景分析：对当前驾驶场景进行更加全面的场景分析，比传统自动驾驶更加的全面。将物体特征分为3个方面——静态属性（Cs）、运动状态（Cm）和特殊行为（Cp）。并非强制所有关键物体都输出这三方面的信息，而是使模型学会应该自适应地输出某个物体在这三方面中可能包含的方面。
- 层级规划：依次推理对应自车未来驾驶决策的元动作（粒度的动作）、决策描述（对于驾驶决策更加详细多维地描述）、轨迹点三种规划目标。

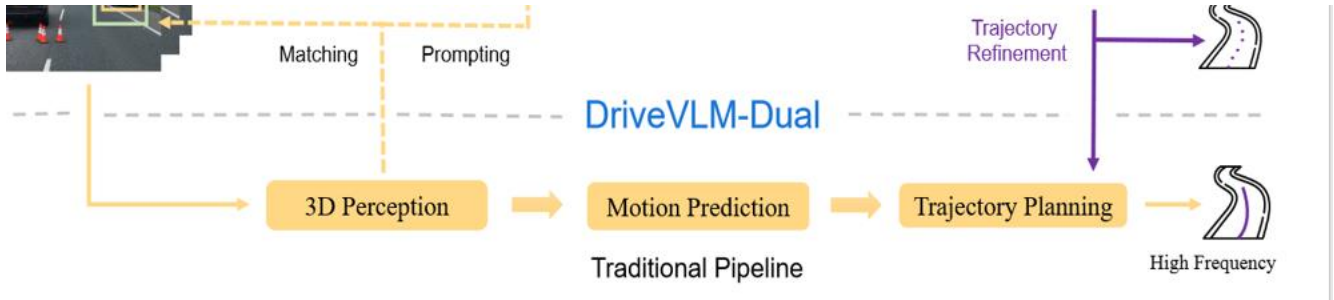


DriveVLM- DriveVLM-Dual

--DRIVEVLM: The Convergence of Autonomous Driving and Large Vision-Language Models

- 尽管现有的大视觉语言模型在识别长尾物体和理解复杂场景方面表现优越，但VLM有时在涉及到推理物体的细微运动状态改变时表现不佳。另外，由于大语言模型巨大的参数量，导致模型的推理时间相比传统自动驾驶系统往往具有较高的延迟，阻碍了其对环境的快速实时反应。
- DriveVLM-Dual ： VLM与传统自动驾驶系统互相协作。

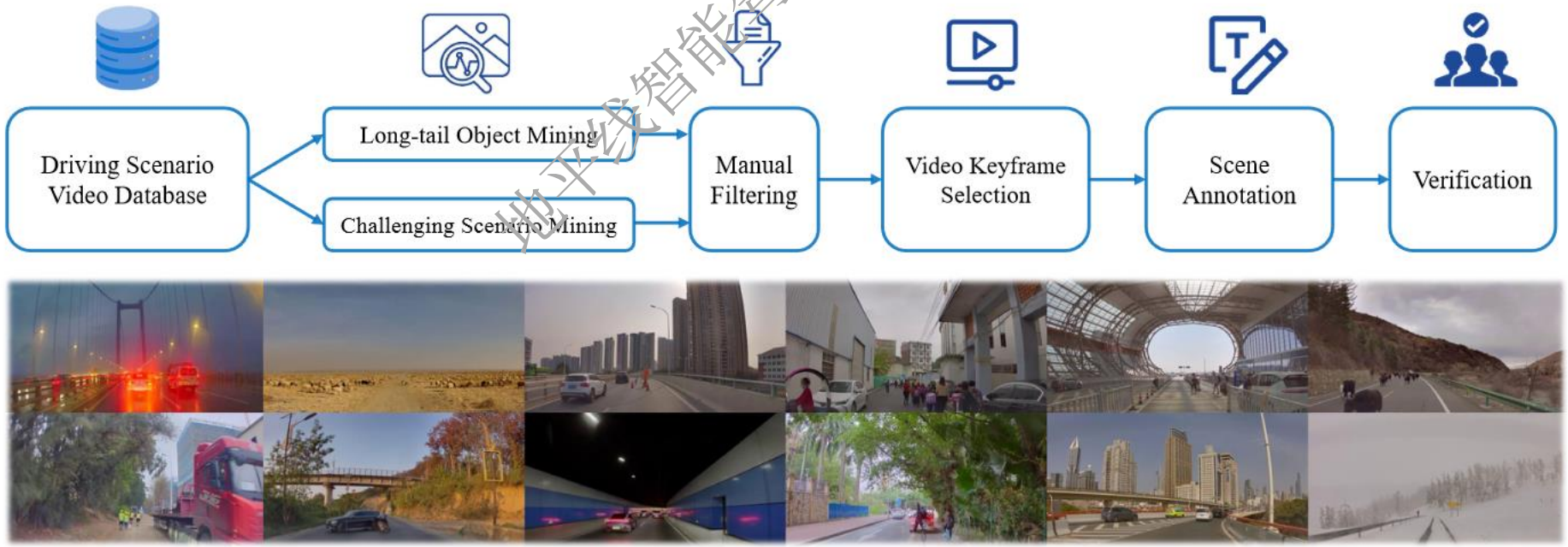
- 3D感知信息融合：将3D检测器检测到的目标物体与VLM检测到的物体进行匹配。
- 对于匹配的关键物体：将感知模块中预测信息用来辅助VLM进行物体特征的推理。
- 对于没有匹配的关键物体：不使用的3D感知信息作为辅助，但依旧用VLM分析。
- 高频轨迹优化：在DriveVLM输出一个规划轨迹之后，将其作为一个参考轨迹送入经典的规划模块进行一个二阶段的轨迹优化。



DriveVLM- SUP-AD数据集

--DRIVEVLM: The Convergence of Autonomous Driving and Large Vision-Language Models

- 数据集构建数据挖掘和标注的方法：首先从海量自动驾驶数据中进行长尾目标挖掘和挑战性场景挖掘来收集样本数据，之后对于每个场景选择一个关键帧，并进行相应的场景信息标注。
- 长尾目标挖掘：首先预定义了一系列长尾目标类别，比如异形车辆、道路杂物和横穿马路的动物等。接下来，使用基于CLIP的搜索引擎从海量自动驾驶数据中挖掘这些长尾场景。最后人工检查以筛选出与指定类别不一致的场景。
- 挑战性场景挖掘：这些场景一般是根据记录的驾驶操作变化得到的，例如急刹车等。在得到相应数据后，同样会进行人工筛选来过滤出不满足要求的数据。
- 关键帧选择：在大多数具有挑战性的场景中，关键帧是在需要显著改变速度或方向之前的时刻。选择在实际操作前0.5秒到1秒作为关键帧，以确保改变驾驶决策的最佳反应时间。
- 场景标注：对于选取好关键帧后的数据，由一组标注员进行场景标注，包括任务提到的场景描述、场景分析和规划等内容信息。



DriveVLM- 数据集演示

--DRIVEVLM: The Convergence of Autonomous Driving and Large Vision-Language Models

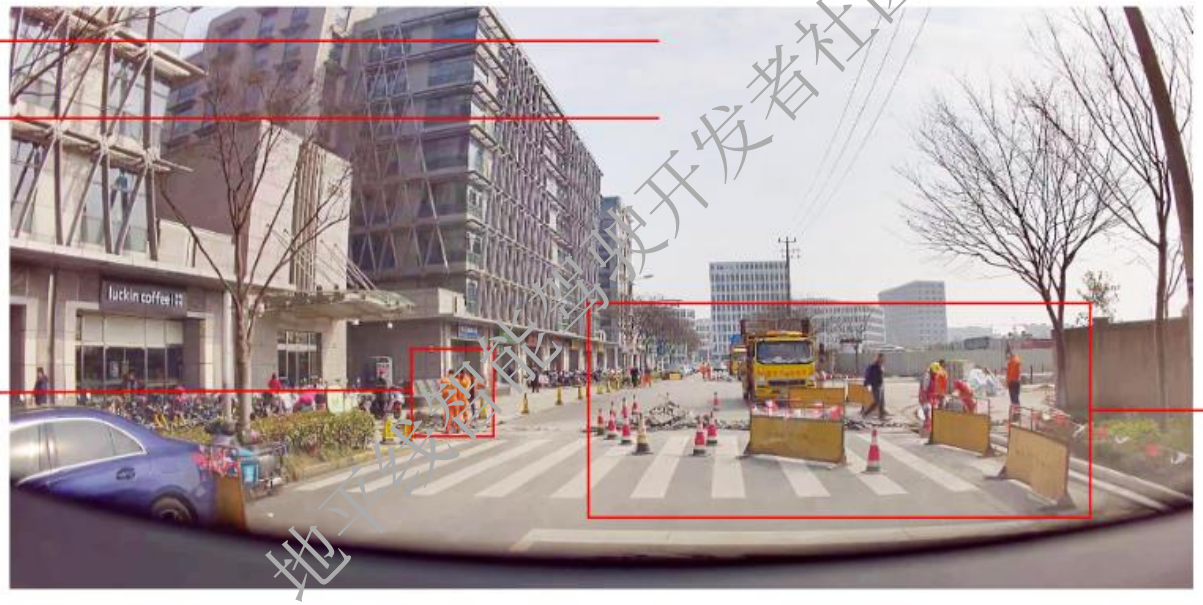
Scene Summary: The ego vehicle is moving at a constant speed along the current lane, with ongoing road construction work ahead; there are three construction workers working on the left side of the lane at the roadside.

Weather: Sunny

Time: Daytime

Critical Object 1:

Class:	Three Construction Workers
Characteristic:	Construction work on the side of the lane to the left of the host vehicle
Influence:	Affects the normal speed of the host vehicle



Road Condition: Construction

Lane Condition: Own Lane

Critical Object 2:

Class:	Construction Zone
Characteristic:	Road repair in front of the host vehicle lane
Influence:	Affects the host vehicle to drive straight normally

Meta Action: ["Slow down", "Change lane to the left", "Go straight slowly"]

Decision Description: Decelerate and change lanes to the left, keeping a safe distance from the construction workers on the left front side.

DriveVLM- 实验结果

--DRIVEVLM: The Convergence of Autonomous Driving and Large Vision-Language Models

- 在的SUP-AD和nuScenes数据集上进行了相应的实验来验证DriveVLM的有效性。

Table 1. **Results on the test set of our proposed SUP-AD dataset.** †: Using the official API of GPT-4V. For Lynx and CogVLM, we utilize the training split for fine-tuning purposes. In contrast, for GPT-4V, we employ in-context learning.

Method	Scene Description	Meta-actions
Fine-tuning w/ Lynx [54]	0.46	0.15
Fine-tuning w/ CogVLM [44]	0.49	0.22
GPT-4V† [36]	0.38	0.19
DriveVLM w/ Qwen	**0.71**	**0.37**

Table 4. **Inference speed on the NVIDIA Orin platform.** "Trad. AD" represents end-to-end traditional autonomous driving method similar to VAD [24].

Method	Trad. AD	DriveVLM	DriveVLM-Dual
Inference time per scene	0.3s	1.5s	0.3s

Table 2. **Planning results on the nuScenes validation dataset.** DriveVLM-Dual achieves the best performance. † means using perception and occupancy prediction results from Uni-AD. ‡ denotes cooperating with VAD [24]. All the models take ego states as input.

Method	L2 (m) ↓				Collision (%) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
NMP† [53]	-	-	2.31	-	-	-	1.92	-
SA-NMP† [53]	-	-	2.05	-	-	-	1.59	-
FF† [20]	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43
EO† [25]	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33
ST-P3 [21]	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD [22]	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31
VAD-Base [24]	0.17	0.34	0.60	0.37	0.07	0.10	0.24	0.14
DriveVLM	0.18	0.34	0.68	0.40	0.10	0.22	0.45	0.27
DriveVLM-Dual†	0.17	0.37	0.63	0.39	0.08	0.18	0.35	0.20
DriveVLM-Dual‡	**0.15**	**0.29**	**0.48**	**0.31**	**0.05**	**0.08**	**0.17**	**0.10**

DriveVLM- 可优化之处

--DRIVEVLM: The Convergence of Autonomous Driving and Large Vision-Language Models

- 对于场景的描述需要大量的算力，但是大部分去情况如此详细的描述不必须。

SparseDrive- 出处

SparseDrive: End-to-End Autonomous Driving via Sparse Scene Representation

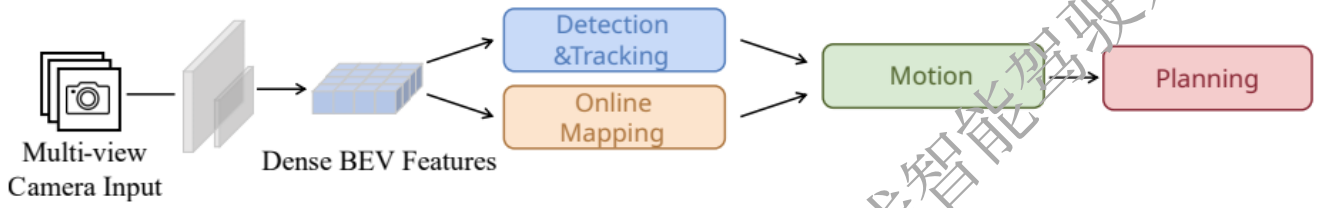
Wenchao Sun[1,2] **Xuewu Lin**[2] **Yining Shi**[1] **Chuang Zhang**[1] **Haoran Wu**[1] **Sifa Zheng**[1]

[1] Tsinghua University [2] Horizon

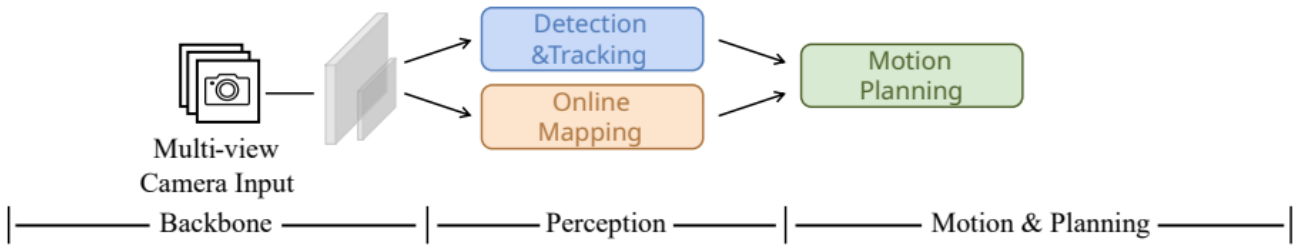
SparseDrive- 背景

--SparseDrive: End-to-End Autonomous Driving via Sparse Scene Representation

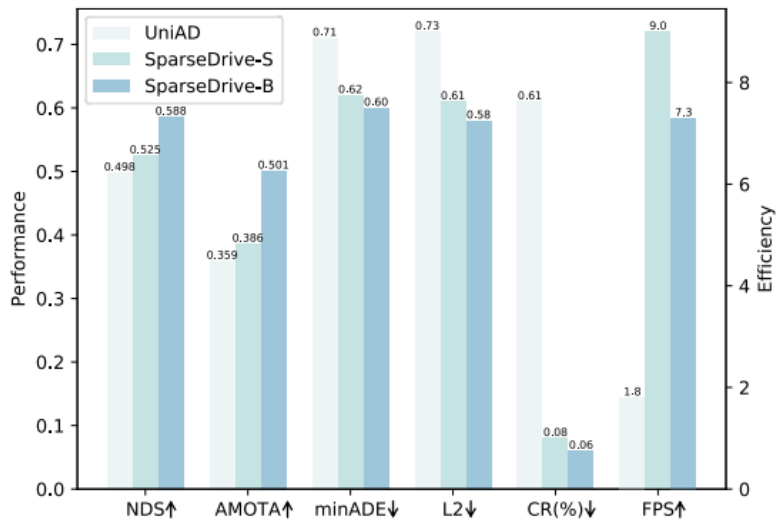
(a) 以BEV为中心的范式：使用稠密的BEV特征，计算成本高昂。在性能和效率方面并不令人满意，特别是在规划安全性方面。
(b) 稀疏为中心的范式：其统一了多任务与稀疏实例表示。
(c) 性能和效率比较。



(a) BEV-Centric paradigm.



(b) Sparse-Centric paradigm.

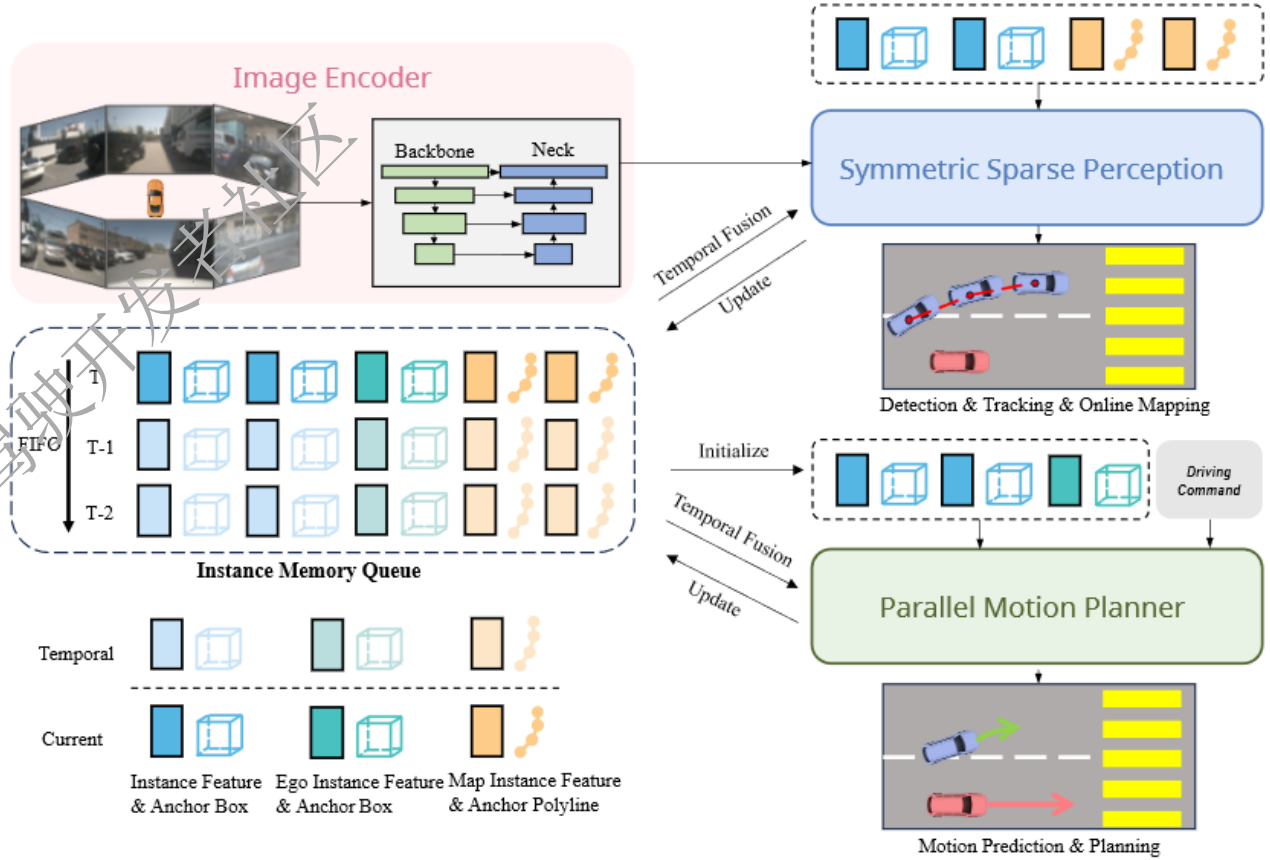


(c) Comparison between our method and privous SOTA[15].

SparseDrive- 流程介绍

--SparseDrive: End-to-End Autonomous Driving via Sparse Scene Representation

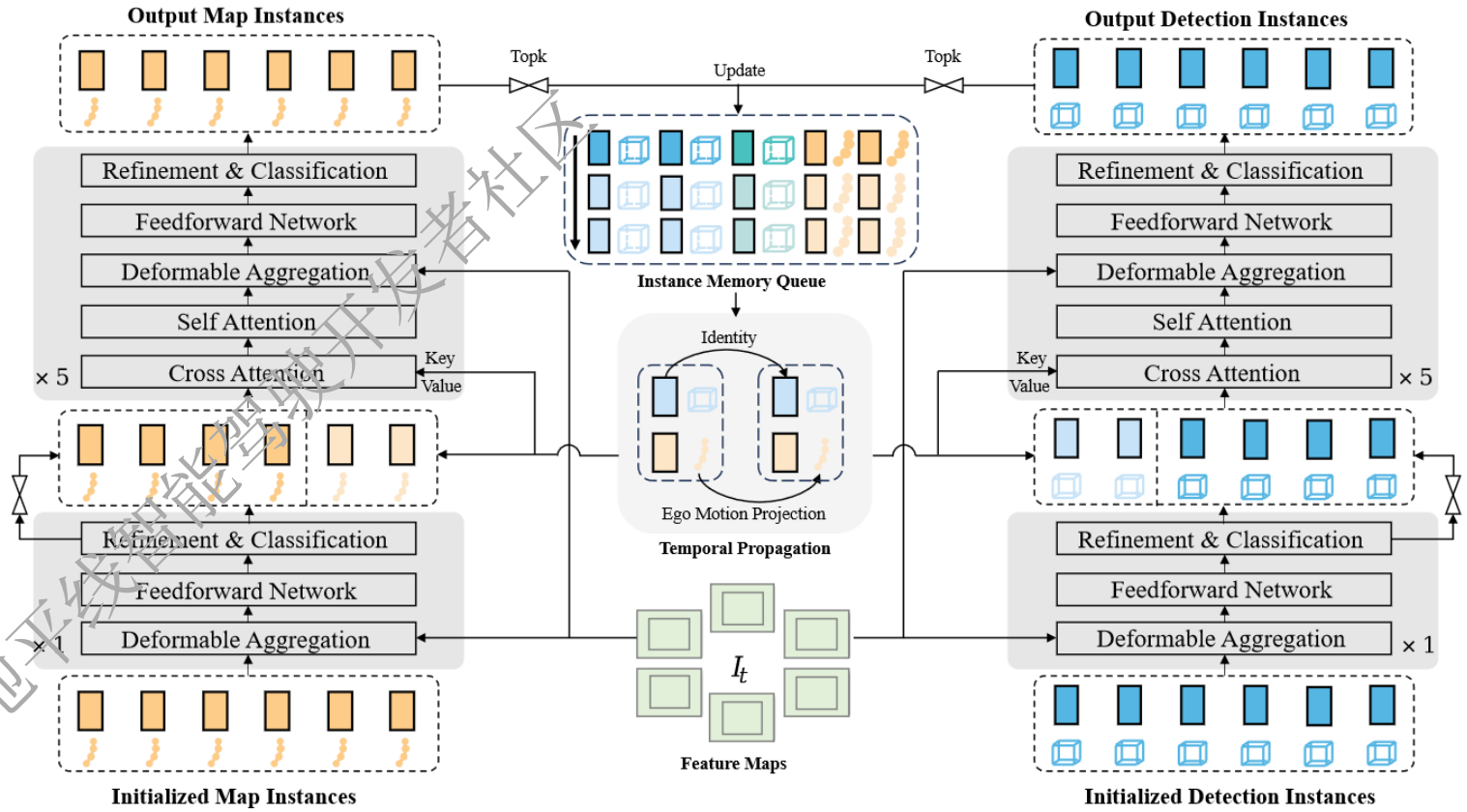
- 图像编码器：多视角摄像头输入，覆盖车辆周围的环境。获取图像特征图。
- 对称稀疏感知模块：特征图聚合成两组实例。分别代表周围的代理和地图元素。
- 并行运动规划器：预测周围物体和自车的多模态轨迹。通过分层规划选择策略结合碰撞感知重评分模块，选择安全的轨迹作为最终的规划结果。



SparseDrive- 对称稀疏感知模块

--SparseDrive: End-to-End Autonomous Driving via Sparse Scene Representation

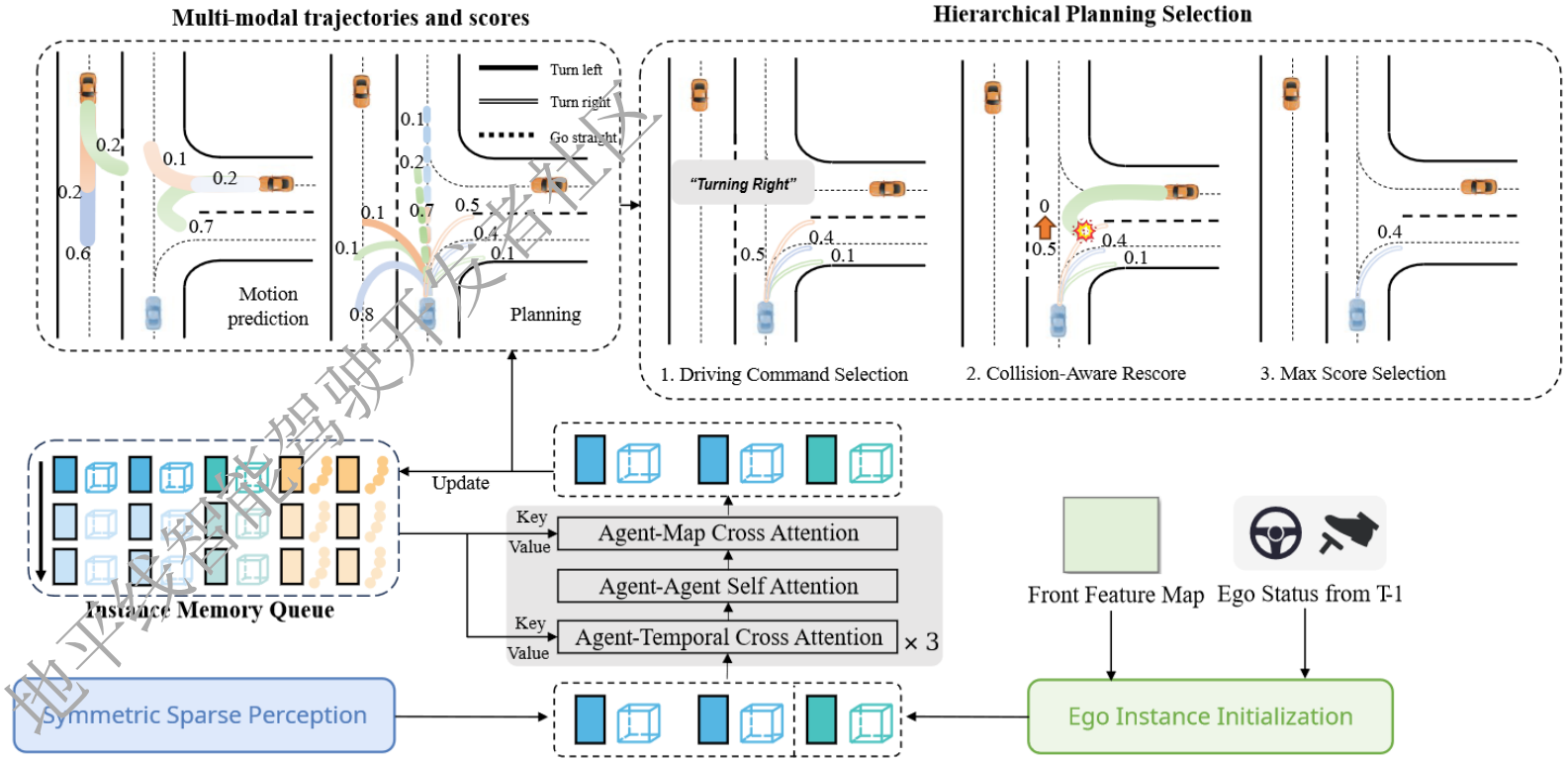
- 该模块在对称结构中统一了检测、跟踪和在线映射任务，采用对称模型架构，学习驾驶场景的完全稀疏表示。
- Sparse Detection：包括6个解码层（1个非时间解码器和5个时间解码器）。
- Sparse Online Mapping：与检测分支共享相同的模型结构，但实例定义不同。
- Sparse Tracking：一旦检测物体置信度超过阈值，实例被锁定并分配一个ID，贯穿时间传播过程中保持不变。



SparseDrive- 并行运动规划器

--SparseDrive: End-to-End Autonomous Driving via Sparse Scene Representation

- 重新审视了运动预测和规划之间的相似性，提出了并行设计的运动规划器，并引入了层次规划选择策略和碰撞感知重评分模块，以提高规划性能。
- Ego Instance Initialization：使用前置摄像头的最小特征图通过平均池化进行特征初始化。使用上一帧预测的速度作为当前帧的速度初始值，以防自车状态泄露。
- Spatial-Temporal Interactions：通过代理-时间交叉注意力、代理-代理自注意力和代理-地图交叉注意力进行时空交互。
- Hierarchical Planning Selection：结合碰撞感知重评分模块，从多模态轨迹提案中选择最终的规划轨迹。



SparseDrive- 实验结果

--SparseDrive: End-to-End Autonomous Driving via Sparse Scene Representation

- 实验在nuScenes数据集上进行，该数据集包含1000个复杂的驾驶场景，每个场景持续约20秒

Method	minADE(m)↓	minFDE(m)↓	MR↓	EPA↑
Cons Pos. [15]	5.80	10.27	0.347	-
Cons Vel. [15]	2.13	4.01	0.318	-
Traditional [10]	2.06	3.02	0.277	0.209
PnPNet [28]	1.15	1.95	0.226	0.222
ViP3D [10]	2.05	2.84	0.246	0.226
UniAD[15]	0.71	1.02	0.151	0.456
SparseDrive-S	0.62	0.99	0.136	0.482
SparseDrive-B	**0.60**	**0.96**	**0.132**	**0.555**

(a) Prediction results.

Method	L2(m)↓				Col. Rate(%)↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
FF* [13]	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43
EO* [22]	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33
ST-P3 [14]	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD† [15]	0.45	0.70	1.04	0.73	0.62	0.58	0.63	0.61
VAD† [20]	0.41	0.70	1.05	0.72	0.03	0.19	0.43	0.21
SparseDrive-S	**0.29**	0.58	0.96	0.61	**0.01**	0.05	0.18	0.08
SparseDrive-B	**0.29**	**0.55**	**0.91**	**0.58**	**0.01**	**0.02**	**0.13**	**0.06**

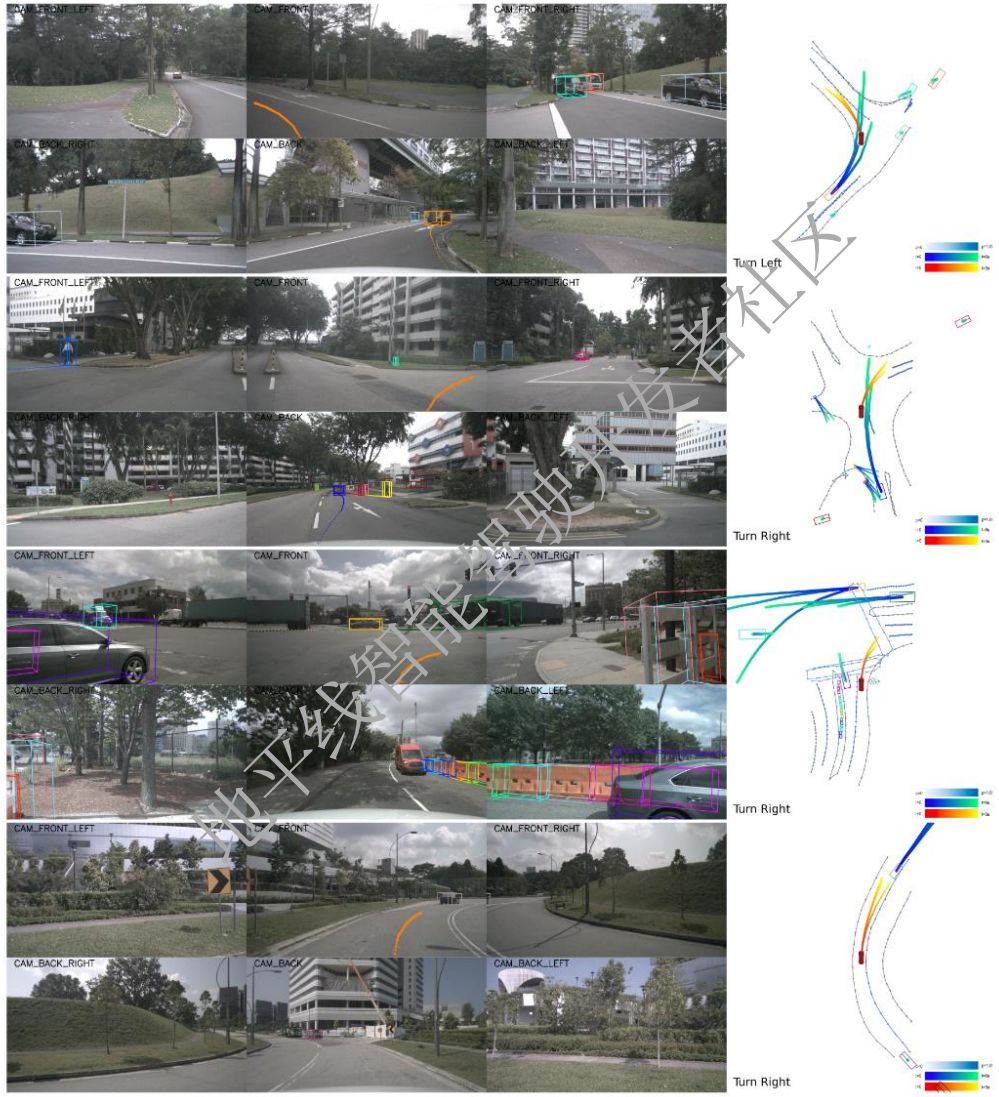
(b) Planning results.

- 在训练效率方面，SparseDrive-S 的表现最佳，其次是 SparseDrive-B，最后是 UniAD。
- 在推理效率方面，SparseDrive-S 依然表现最佳，SparseDrive-B 紧随其后，UniAD 的表现最差。

Method	Training Efficiency			Inference Efficiency			
	GPU Memory (G)	Batch Size	Time (h)	GPU Memory (M)	FLOPs (G)	Params (M)	FPS
UniAD [15]	50.0	1	48 + 96	2451	1709	125.0	1.8
SparseDrive-S	15.2	6	18 + 2	1294	192	85.9	9.0
SparseDrive-B	17.6	4	26 + 4	1437	787	104.7	7.3

SparseDrive- 可视化

--SparseDrive: End-to-End Autonomous Driving via Sparse Scene Representation



SparseDrive在交叉路口学习了不同的转弯模式

SparseDrive- 未来工作

- 对于长尾场景的测试不够充分。
- 无闭环测试结果。

PlanKD- 出处

On the Road to Portability: Compressing End-to-End Motion Planner for Autonomous Driving

Kaituo Feng[1], Changsheng Li[1]*, Dongchun Ren[2], Ye Yuan[1], Guoren Wang[1,3]
[1] Beijing Institute of Technology [2] ALLRIDE.AI
[3] Hebei Province Key Laboratory of Big Data Science and Intelligent Technology

PlanKD- 背景

--On the Road to Portability: Compressing End-to-End Motion Planner for Autonomous Driving

- 挑战
  - 在复杂的驾驶场景中，不是所有信息都对规划有益，一些信息可能是无关的甚至是噪音。因此，自动提取与规划相关的信息是至关重要的，而传统的知识蒸馏方法往往无法实现这一目标。
  - 在规划轨迹中，不同waypoints的重要性可能不同。例如，在交叉口附近与其他车辆接近的waypoints可能更加重要，因为此时自车需要主动与其他车辆互动，任何微小偏差都可能导致碰撞。确定关键waypoints并准确模拟它们是传统知识蒸馏方法面临的另一个重要挑战。
  - 在常规车辆中，车载设备上的计算资源也经常是有限的。因此，直接部署深层且庞大的规划器不可避免地需要更多的计算时间和资源，这使得快速响应潜在危险变得具有挑战性。

- PlanKD
  - 采用知识蒸馏来压缩端到端运动规划模型。基本思想是通过从更大的教师模型继承知识来训练一个简化的学生模型，并在部署期间使用学生模型来替代教师模型。

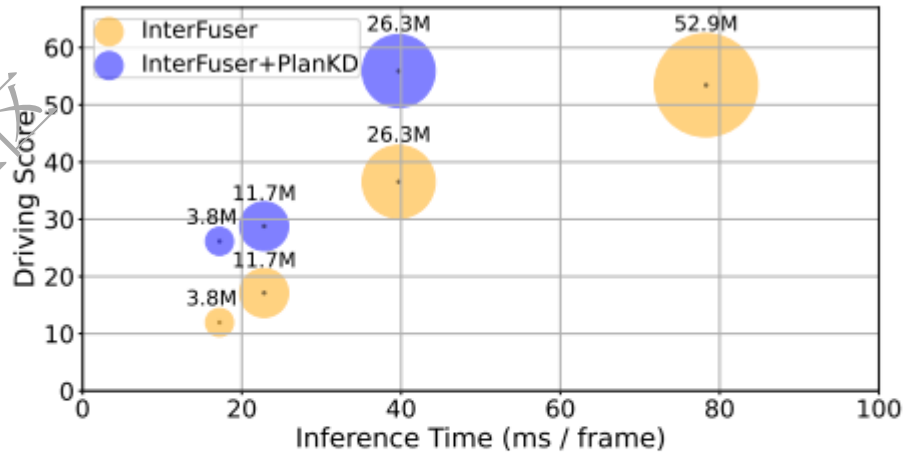
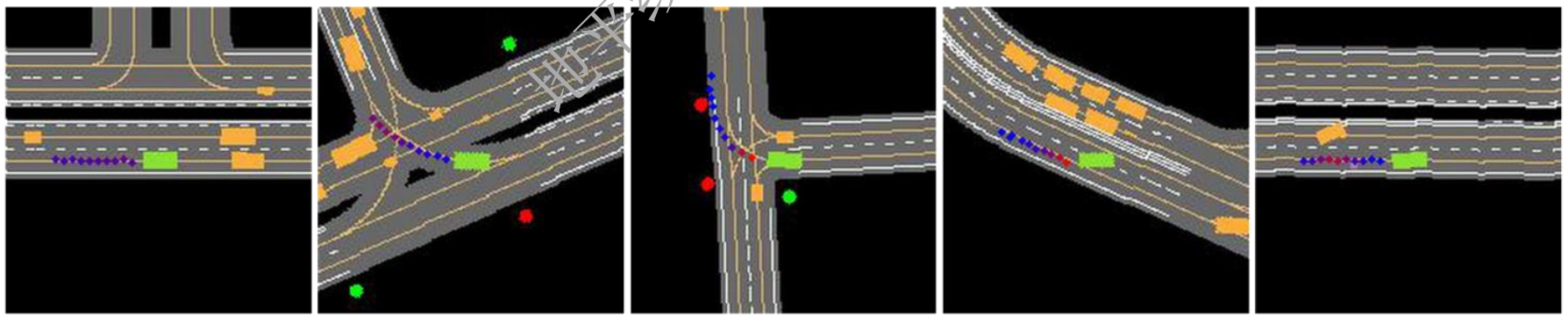


Figure 1. An illustration for the performance degradation of InterFuser [33] on Town05 Long Benchmark [31] as the number of parameters decreases. By leveraging our PlanKD, the performance of compact motion planners can be enhanced, and the inference time can be significantly lowered. The inference time is evaluated on GeForce RTX 3090 GPU in a server. Best viewed in color.

PlanKD- 优化思路

--On the Road to Portability: Compressing End-to-End Motion Planner for Autonomous Driving

- 思路
  - 在复杂的驾驶场景中，不是所有信息都对规划有益。文章提出了一个基于信息瓶颈原理的策略，其目标是提取包含最少且足够规划信息的与规划相关的特征。具体来说，本文最大化提取的与规划相关特征和规划状态的真值之间的互信息，同时最小化提取特征和中间特征映射之间的互信息。这一策略使PlanKD能够只在中间层提取关键的与规划相关的信息，从而增强学生模型的有效性。
  - 在规划轨迹中，不同waypoints的重要性可能不同。 PlanKD采用注意力机制计算每个 waypoints 及其在BEV中与关联上下文之间的注意力权重。为了在蒸馏过程中促进对安全关键 waypoints 的准确模仿，本文设计了一个 safety-aware ranking loss ， 鼓励对于靠近移动障碍物的 waypoints 给予更高的注意力权重。相应地，学生规划器的安全性可以显著增强。



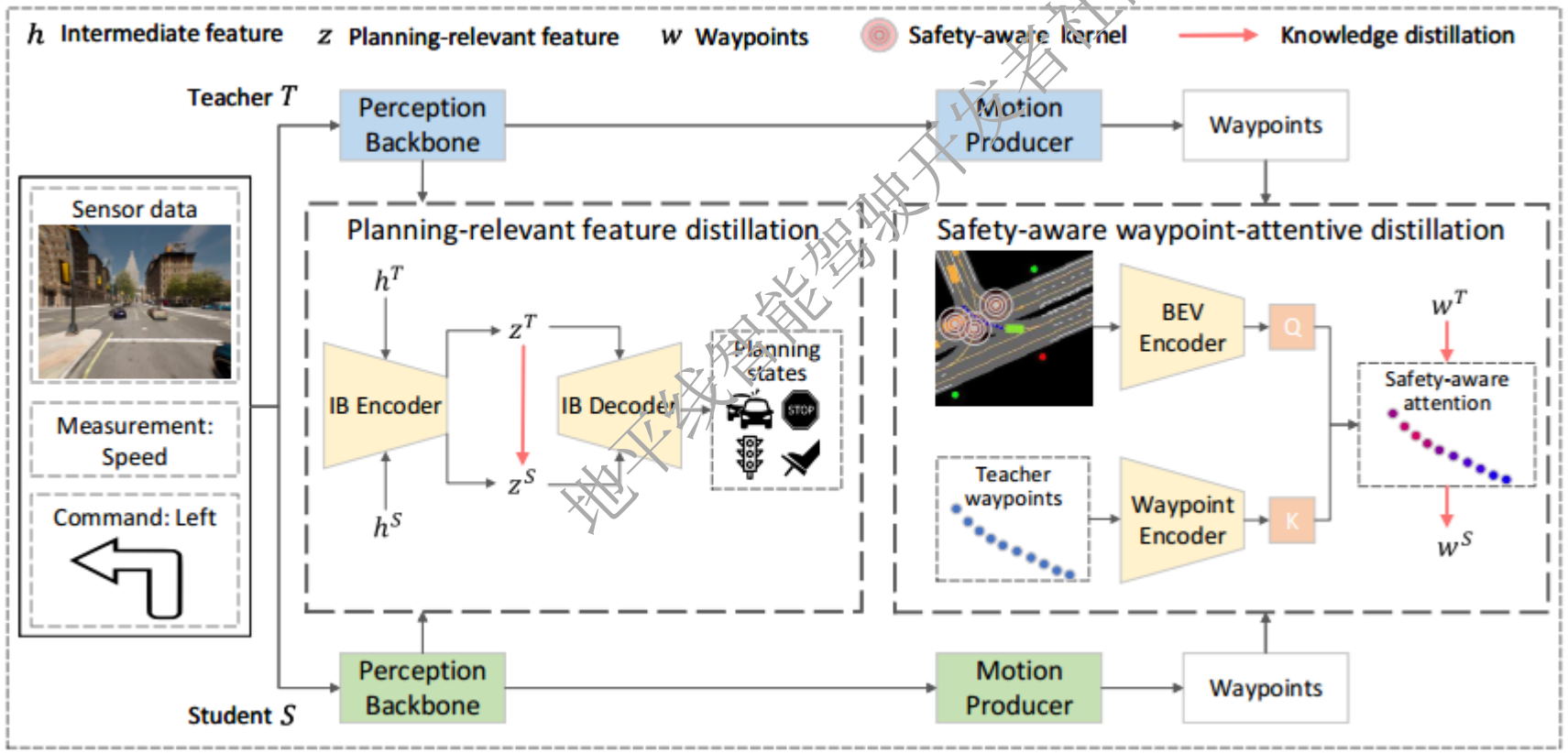
(a) (b) (c) (d) (e)

Figure 3. Visualizations of safety-aware attention weights under different driving scenarios. The green block denotes the ego-vehicle and the yellow blocks represent other road users (e.g. vehicles, bicycles). The redder a waypoint is, the higher attention weight it has.

PlanKD- 流程介绍

--On the Road to Portability: Compressing End-to-End Motion Planner for Autonomous Driving

- PlanKD 由两个模块组成：首先设计了与规划相关的特征提取模块。它利用信息瓶颈原理从中间特征映射中提取与规划相关的信息，并将这些信息传递给学生模型进行有效的提炼。此外，还设计了一个安全感知的路点关注蒸馏模块。该模块可以根据航路点的重要性为其分配自适应权重，并提取这些信息以提高整体安全性。



PlanKD- 贡献与实验结果

--On the Road to Portability: Compressing End-to-End Motion Planner for Autonomous Driving

(a) 本文构建了第一个旨在探索专用知识蒸馏方法以压缩自动驾驶中端到端运动规划器的尝试。
(b) 本文提出了一个通用且创新的框架 PlanKD，它使学生规划器能够继承中间层中与规划相关的知识，并促进关键 waypoints 的准确匹配以提高安全性。
(c) 实验表明，本文的 PlanKD 可以大幅提升小型规划器的性能，从而为资源有限的部署提供了一个更便携、更高效的解决方案。
(d) CARLA Town05闭环测试。

Table 1. Overall performance of motion planners of different size, with and without utilizing PlanKD, on the Town05 Long Benchmark. The inference time per frame is evaluated on GeForce RTX 3090 GPU.

Backbone	Param Count	Inference Time (ms)	With PlanKD	Driving Score(↑)	Route Completion(↑)	Infraction Score(↑)	Collision Rate(↓)	Infraction Rate(↓)
InterFuser	52.9M	78.3	-	53.44	97.52	0.549	0.090	0.078
	26.3M	39.7	✗	36.55	94.00	0.425	0.121	0.068
	26.3M	39.7	✓	55.90	97.44	0.562	0.094	0.093
	11.7M	22.8	✗	17.12	66.19	0.358	0.362	0.283
	11.7M	22.8	✓	28.79	80.50	0.430	0.315	0.202
	3.8M	17.2	✗	11.96	64.56	0.335	1.117	0.722
	3.8M	17.2	✓	26.15	70.95	0.410	0.361	0.265
TCP	25.8M	17.9	-	53.41	100.0	0.534	0.076	0.115
	13.9M	10.7	✗	39.96	91.06	0.443	0.183	0.157
	13.9M	10.7	✓	53.19	93.28	0.579	0.084	0.116
	7.6M	8.5	✗	25.88	52.69	0.690	0.110	0.101
	7.6M	8.5	✓	35.44	63.95	0.673	0.096	0.087
	3.1M	7.2	✗	16.16	31.33	0.781	0.098	0.161
	3.1M	7.2	✓	23.84	32.03	0.858	0.052	0.074

Table 2. Overall performance of motion planners of different size, with and without utilizing PlanKD, on the Town05 Short Benchmark. The inference time per frame is evaluated on GeForce RTX 3090 GPU.

Backbone	Param Count	Inference Time (ms)	With PlanKD	Driving Score(↑)	Route Completion(↑)	Infraction Score(↑)	Collision Rate(↓)	Infraction Rate(↓)
InterFuser	52.9M	78.3	-	94.88	99.88	0.950	0.141	0.000
	26.3M	39.7	✗	82.70	96.87	0.841	0.662	0.141
	26.3M	39.7	✓	93.69	96.43	0.960	0.207	0.000
	11.7M	22.8	✗	58.17	89.69	0.644	0.731	1.638
	11.7M	22.8	✓	74.57	96.65	0.762	0.479	0.622
	3.8M	17.2	✗	54.56	79.11	0.688	0.503	1.603
	3.8M	17.2	✓	65.18	81.52	0.807	0.337	0.863
TCP	25.8M	17.9	-	95.53	99.53	0.960	0.141	0.000
	13.9M	10.7	✗	84.53	88.53	0.960	0.141	0.000
	13.9M	10.7	✓	95.49	99.49	0.960	0.141	0.000
	7.6M	8.5	✗	79.34	86.84	0.919	0.303	0.000
	7.6M	8.5	✓	83.59	87.09	0.960	0.162	0.000
	3.1M	7.2	✗	58.70	65.17	0.930	0.162	0.143
	3.1M	7.2	✓	64.86	68.36	0.960	0.162	0.000

02

业界方案

毫末智行

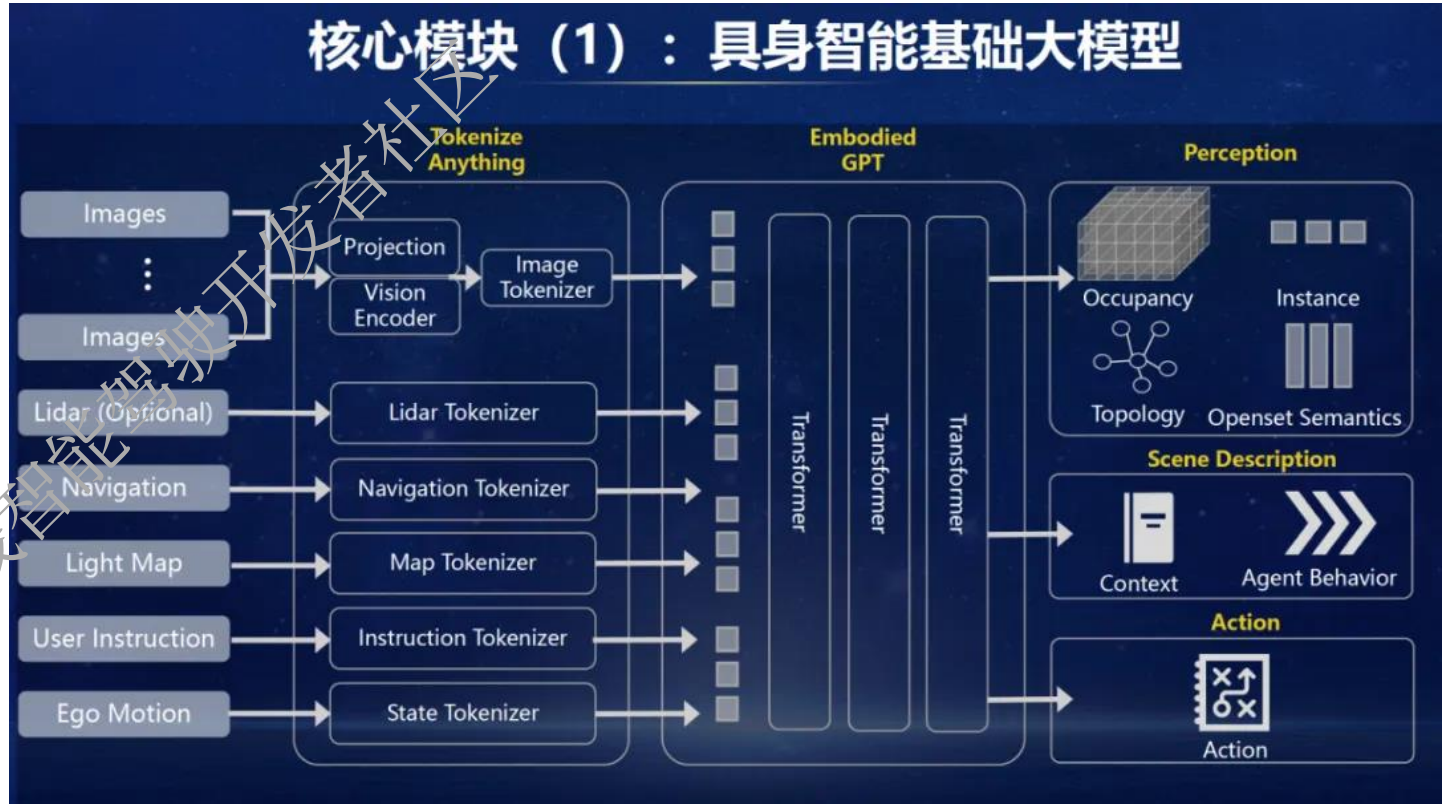
- 将端到端大模型进行拆分，分为2个阶段：解决感知问题，解决认知问题，这样做的优点有：
  - 可以先独立训练，再进行联合，降低训练难度
  - 不同阶段可以采用不同的数据，大幅降低数据成本。

- 在感知阶段，主要是将视觉信号转为感知结果，利用海量视频的采集数据和量产车回传各种corner case视频来训练。在认知阶段，根据感知结果来进行驾驶决策，不输入视频，只输入感知结果和驾驶行为即可。
- 建立自监督感知大模型，将车载摄像头的二维视频数据进行编码，然后通过NeRF渲染来预测视频的下一帧图像，构建4D特征空间。再通过多模态技术将视觉信号与文本信号对齐，实现识别万物。
- 引入大语言模型，感知万物后，将信息输入LLM，通过LLM提取世界知识，作为辅助特征来指导驾驶决策，这个系统复杂，算力消耗大，目前只能在云端运行，未来几年大概率向车端落地。

ApolloFM

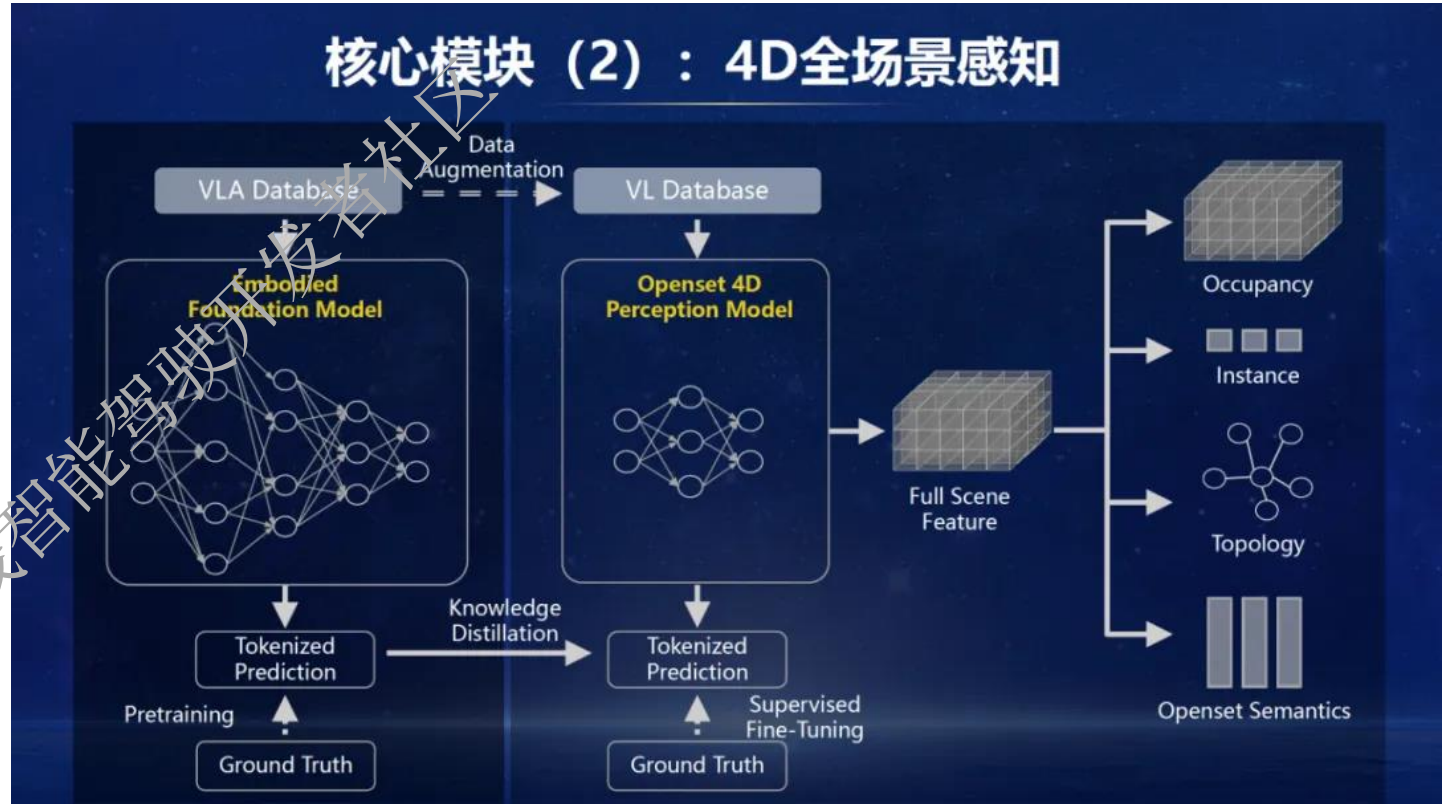
- 如何应用大模型、多模态、生成式、预训练等技术解决自动驾驶行业的核心挑战，是我们希望能够在新架构设计中重点关注的问题。

- 这个模型的输入和网络结构比较类似于多模态的GPT大模型。
- 输入：摄像头、激光雷达、导航、地图（仅需要导航级别地图，而不依赖高精度地图）、用户指令、自车位置等信息。
- 输出：开集的感知信息、场景描述信息、驾驶action。
- VLA（Vision-Language-Action）模型。



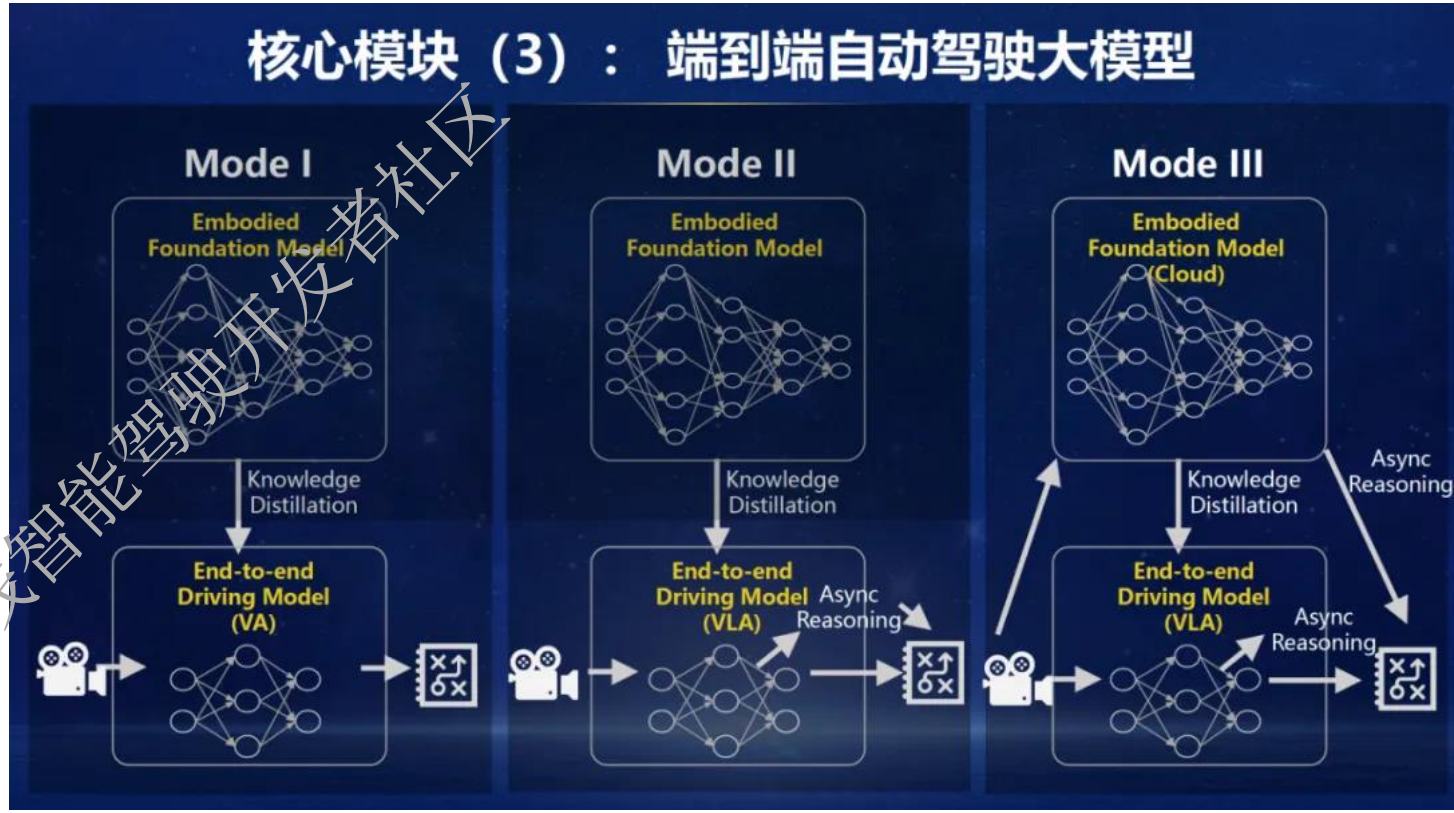
ApolloFM

- 这个模型在VLA 模型基础上蒸馏成为的一个 VL 模型。
- 会控制算力和网络结构保证实时运行。
- 它输出的要素是开集下的世界模型（感知万物），包括稠密表达、实例化表达以及之间元素的拓扑关系。



ApolloFM

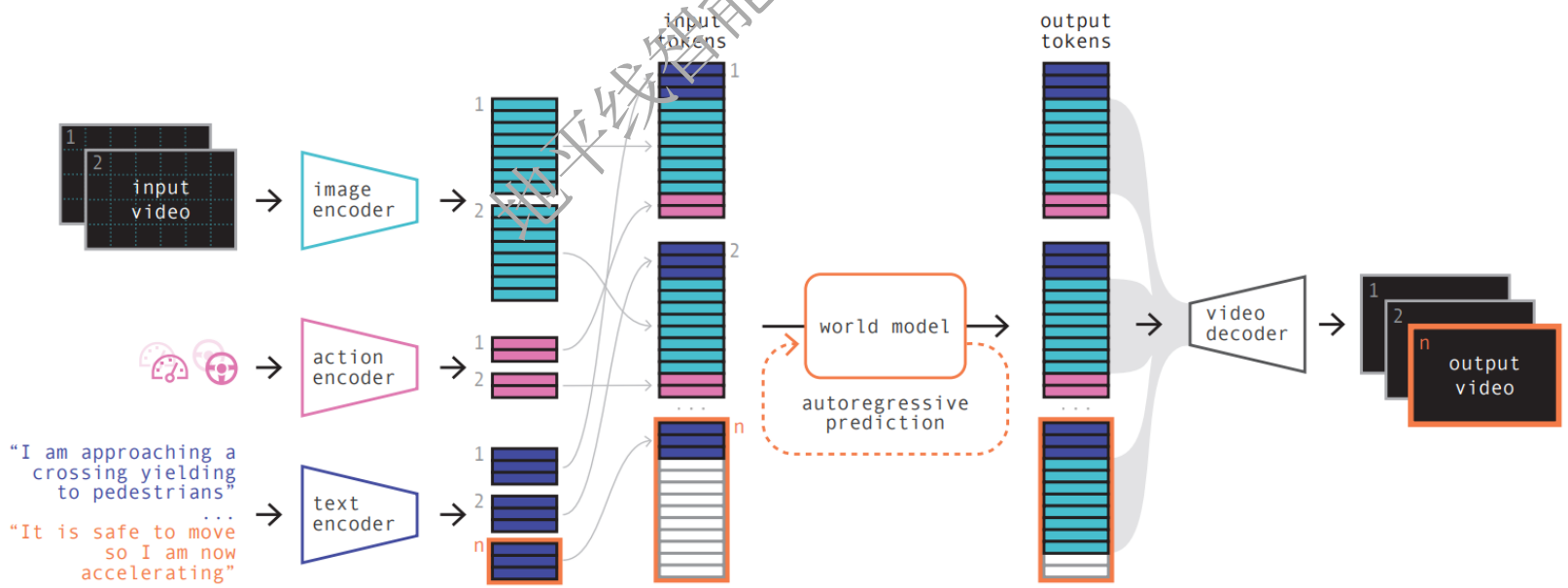
- 第一种模式实时端侧的VA，不再具有L的结构。这样的VA模型的好处是算力要求低，可以做到实时运行，类似于人类的小脑。这个模型具备端到端自动驾驶的能力，而且从VLA大模型中蒸馏出的VA模型有更高的上限，是端到端自动驾驶一条非常关键的路径。
- 第二种是实时端侧VA+异步端侧 VLA。
- 第三种模式是实时端侧VA+异步端侧VLA+异步云侧VLA。



Wayve

--GAIA-1: A Generative World Model for Autonomous Driving

- GAIA-1作为一种生成世界模型，凭借其强大的学习能力，能够结合视频、文本和动作输入，生成逼真的驾驶场景，并对车辆行为和场景特征进行精细控制。
- 通过无监督序列建模，GAIA-1能够预测序列中的下一个标记，从而捕捉未来事件的预测，并与真实样本的生成能力相结合，显著增强和加速了自动驾驶技术的训练过程。
- 世界模型和video diffusion decoder：世界模型负责理解场景中的high-level的部分及场景的动态演化信息，而video diffusion decoder则负责将潜在表征转化为具有真实细节的高质量视频。



Wayve

- 视频推断、场景推断 、文生视频。