

PARA-Drive: Parallelized Architecture for Real-time Autonomous Driving

Xinshuo Weng¹, Boris Ivanovic¹, Yan Wang¹, Yue Wang^{1,2}, Marco Pavone^{1,3}
¹NVIDIA Research, ²University of Southern California, ³Stanford University
 {xweng, bivanovic, yanwan, yuewang, mpavone}@nvidia.com

Abstract

Recent works have proposed end-to-end autonomous vehicle (AV) architectures comprised of differentiable modules, achieving state-of-the-art driving performance. While they provide advantages over the traditional perception-prediction-planning pipeline (e.g., removing information bottlenecks between components and alleviating integration challenges), they do so using a diverse combination of tasks, modules, and their interconnectivity. As of yet, however, there has been no systematic analysis of the necessity of these modules or the impact of their connectivity, placement, and internal representations on overall driving performance. Addressing this gap, our work conducts a comprehensive exploration of the design space of end-to-end modular AV stacks. Our findings culminate in the development of PARA-Drive¹: a fully parallel end-to-end AV architecture. PARA-Drive not only achieves state-of-the-art performance in perception, prediction, and planning, but also significantly enhances runtime speed by nearly 3 \times , without compromising on interpretability or safety.

1. Introduction

Developing a full-stack autonomous vehicle (AV) system poses significant challenges due to the need to integrate many complex modules such as detection, tracking, prediction, localization, mapping, and planning. To address these challenges, there has been a growing trend towards the development of end-to-end yet modular AV systems, such as UniAD [12], VAD [13] and OccNet [25]. These systems have gained popularity because they have successfully integrated various modules and tasks and achieved state-of-the-art motion planning performance. Also, they enhance safety and interpretability by optimizing upstream perception and prediction modules in an end-to-end manner.

Although these end-to-end modular AV stacks have been successful, there are notable differences in their architectural designs (e.g., incorporating occupancy vs. not, semantic BEV vs. vectorized maps, using queries or outputs, see Fig. 1) and it remains unclear which design, if any, is opti-

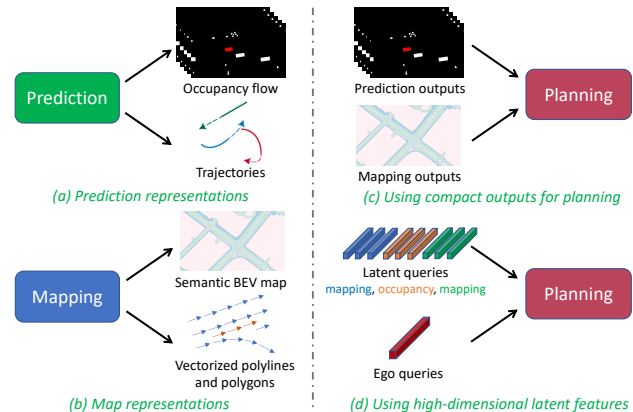


Figure 1. Examples of **design variations** for end-to-end AV stack. (a)(b) Modules for the same task can be designed with different representations; (c)(d) variations of information flow to planning.

mal. Aside from these recent few architectures, the vast design space of end-to-end modular AV stacks remains largely under-explored, primarily due to the multiplicity of possible tasks, associated information representations, and inter-module connectivities. In this work, we tackle this challenge and conduct a systematic exploration of this design space organized along the following three key axes: Module necessity, placement, and information flow (Sec. 3.1).

To effectively navigate this complex landscape, we build a versatile framework that enables the flexible manipulation of an end-to-end stack’s computational graph, including the ability to activate or deactivate modules, modify inter-module connectivity, and access outputs from different layers of upstream modules. With this framework, we uncover several intriguing insights: state-of-the-art performance can be achieved with sequential, hybrid, or even pure parallel designs; state-of-the-art planning performance can be achieved with *only* bird’s-eye-view (BEV) features as inputs to planning (with proper auxiliary tasks and high-level commands); and modules which are redundant when placed sequentially may in fact be complementary when placed in parallel (Sec. 3.3).

We further leverage these insights to develop PARA-Drive, a full-stack parallelized AV architecture that encompasses a diverse set of modules for the co-training of BEV features (Sec. 4). Through comprehensive experiments on

¹Project: <https://xinshuoweng.github.io/paradrive/>

the real-world nuScenes [3] dataset, we show that PARA-Drive significantly outperforms prior work in terms of planning performance (*e.g.*, up to 28.8% reduction in L2 errors and 43.3% reduction in collision rates), with consistent improvements observed in map compliance rates and in challenging scenarios where the ego-vehicle makes turns or lane changes. Our core contributions are fourfold, we:

- (1) Build a flexible framework for end-to-end modular AV architecture, greatly simplifying exploring the design space of end-to-end AV;
- (2) Conduct a systematic study of fundamental design choices for end-to-end AV, instrumental in guiding the future development of modules and stack integration;
- (3) Propose PARA-Drive, an elegant and efficient architecture for end-to-end driving that achieves state-of-the-art perception, prediction, and planning performance, reduces collision rates, and offers high flexibility in interpretability and runtime speed;
- (4) Standardize and enhance evaluation methodologies for end-to-end motion planning (Sec. 3.2) in the open-loop setting, and re-establish a consistent comparison for prior art in end-to-end AV stacks.

2. Related Work

End-to-End Motion Planning. Traditional AV development has predominantly focused on training standalone modules and integrating them to form a complete AV system. However, this approach faces significant integration challenges during deployment. Information bottlenecks are common, with potential loss of information due to thresholding and filtering during inter-module communication. Additionally, the separate training of modules leads to misaligned objectives, resulting in upstream tasks not being optimally tailored for downstream-aware learning.

To overcome these challenges, there have been many recent advancements towards end-to-end planning approaches, pioneered by prior works such as [2, 21]. These approaches are appealing as they offer efficient runtime, and eliminate integration challenges and information bottlenecks. Recent work in this direction has further enhanced the success of end-to-end approaches in closed-loop driving. For instance, [5] improves end-to-end driving by distilling information from a privileged expert. [8] introduces a planning network conditioned on high-level driving commands. Extending beyond camera inputs, Transfuser [22] incorporates LiDAR data and TCP [26] improves output representations for planning by simultaneously considering trajectories and actions.

Despite these advancements, end-to-end planning approaches still face significant challenges in terms of interpretability, verifiability, and safety for real-world deployment. This has led to a growing interest in designing end-to-end yet modular AV architectures, aiming to combine the

benefits of both traditional and end-to-end approaches.

End-to-End Yet Modular Architecture distinguishes itself from traditional and end-to-end planning approaches by integrating modular design with end-to-end training. As a result, it maintains safety and interpretability, while simultaneously optimizing all modules for downstream planning.

Designing an end-to-end modular stack is challenging because the composition and interaction of modules in these architectures lead to distinct design variations. For example, towards a sequential design, P3 [23] and ST-P3 [11] employ the outputs from semantic occupancy prediction for downstream planning. Instead of learning occupancy with supervised learning, [10, 15] develop self-supervised occupancy forecasting to predict free space, which is then used for optimization-based planning. Hybrid designs, such as those explored in [29], leverage BEV features directly in subsequent modules, combining sequential and parallel inter-module connections. This trend is further exemplified in prior works such as [4, 12, 13, 25], which often incorporate a parallel head for online mapping.

To demonstrate these design variations, we compare recent state-of-the-art architectures with our proposed PARA-Drive in Fig. 2. These architectures differ in module inclusion, output representations, and information flow to the planning module. For example, UniAD and OccNet incorporate the predicted occupancy map for planning, while VAD does not. UniAD and OccNet treat online mapping as a dense prediction task, predicting the per-pixel or per-voxel semantics of map elements, whereas VAD opts for vectorized map representations. Also, UniAD and VAD use a multi-head architecture where the planning head accesses intermediate outputs from perception heads, such as latent query features, in contrast to OccNet, which relies on the final outputs of perception heads for planning.

Our work contributes to this evolving landscape by focusing on stack-level design and systematically exploring key dimensions of the design space in end-to-end AV. We hope that our findings can help future development in end-to-end modular design. Also, unlike existing architectures, which often suffer from relatively slow runtime speeds due to dependencies between modules, our PARA-Drive is designed for parallel operation, significantly enhancing efficiency by activating parallel heads only as needed.

Multi-Task Learning (MTL) is not a brand-new topic in AVs and there has been substantial development. Beyond focusing on MTL for perception and prediction [1, 18, 31], MTL has also been applied to end-to-end AV for planning, with recent approaches typically utilizing BEV features for various auxiliary tasks. For example, [28] detects and predicts objects' bounding boxes in parallel to learn the cost volume for planning. Also, the extension of Transfuser [7] incorporates mapping, depth, and object detection heads in

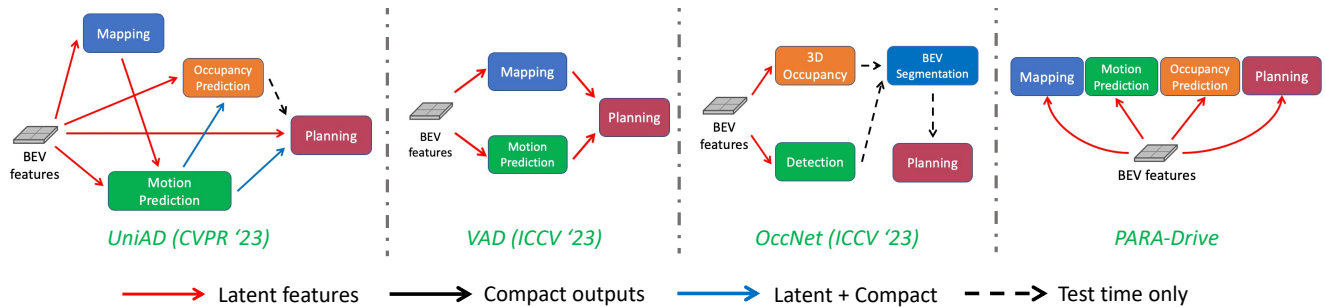


Figure 2. Visual comparison between our PARA-Drive with recent state-of-the-art architectures for end-to-end modular AV, *i.e.*, UniAD [12], VAD [13], OccNet [25]). We highlight three major design variations: (1) the inclusion of a different set of modules, (2) distinct module placements due to different inter-module connections, and (3) information flow with different representations such as high-dimensional latent features and compact outputs from upstream modules. Differing from prior work, our proposed PARA-Drive is designed completely parallelized for runtime efficiency, while achieving state-of-the-art driving performance and maintaining safety and interpretability.

parallel to motion planning. However, these works do not employ the occupancy prediction task, which we find crucial in parallelized AV architectures. In contrast, NEAT [6] and ReasonNet [24] adopt BEV occupancy prediction in MTL-style AV architectures, which however do not predict instance-wise object motions. Moreover, ReasonNet does not predict semantic maps and only relies on learning the map information implicitly from sensor inputs.

In contrast to these prior works in MTL, hybrid designs such as UniAD [12] have demonstrated the efficacy of combining instance-wise motion prediction with scene-level occupancy prediction to enhance end-to-end driving performance. Corroborating this argument, our findings (Sec 3.3) suggest that, beyond incremental improvements observed in hybrid designs, it is crucial to employ instance-wise motion prediction and scene-level occupancy prediction, as well as online mapping, in order to achieve state-of-the-art planning performance in a pure parallelized design. With these insights, we propose PARA-Drive, the first parallelized AV architecture with superior performance and real-time efficiency than AV architectures with hybrid designs.

3. The Design Space of End-to-End AV Stacks

3.1. Key Dimensions

To systematically explore the design space, we first identify and analyze critical dimensions that define the design space.

- (1) *Necessity of modules*: Which modules are essential in an end-to-end modular stack? For a given task, different modules may be designed with distinct representations (*e.g.*, occupancy prediction vs. trajectory forecasting). One needs to consider whether including both types of modules, despite potential redundancy, is more beneficial than choosing just one.
- (2) *Module placement*: Once modules are selected, how should they be arranged within the stack? This includes considering various design configurations, such as sequential and parallel designs, or hybrid designs that

contain both sequential and parallel connections.

- (3) *Information flow*: When a module depends on its predecessor’s output, should we pass only final compact outputs in a relatively low dimension (*e.g.*, bounding boxes or BEV segmentation map) or high-dimensional intermediate features (*e.g.*, token queries)? Would it be beneficial to pass both?

The exploration of the design space is marked by a significant challenge: the potential complexity arising from the combination of multiple dimensions, each of which may appear manageable in isolation. This composite nature has led to the design space encompassing a wide range of configurations, spanning from the non-modular AV stacks that conduct motion planning directly from input sensor data [2, 8, 22, 26] to various highly modular end-to-end approaches such as those in prior work [12, 13, 25, 29].

3.2. Standardized Evaluation Methodology

Datasets and Metrics. Following prior work in end-to-end AV [10–13, 15, 30], we utilize the real-world nuScenes [3] driving dataset to experiment with the variants of end-to-end modular AV architectures in exploring the design space. As we focus on the impact of the design on overall driving performance, we use metrics to evaluate the planning performance for the majority of our comparison. In particular, we follow prior work and compute the L2 error between the ground truth (GT) and the predicted trajectories of the ego vehicle over a 3-second horizon at 2Hz and calculate the collision rate with surrounding dynamic agents in bird’s eye view. Also, we report both the averaged error at the time horizon of 1s, 2s, and 3s as $Ave_{1,2,3s}$ (used in prior work) and also the averaged error of the entire horizon as Ave_{all} .

Evaluation Protocols. To rigorously explore the design space, it is important to ensure the robustness and consistency of evaluation. However, due to the absence of a standardized benchmark for planning on nuScenes, prior work has developed evaluation protocols individually, introducing inconsistencies. To mitigate this issue, rather than di-

Table 1. Planning performance on the nuScenes *val* set. To assess the impact of inconsistencies in evaluation methodologies, we conducted a step-by-step analysis, transforming the VAD evaluation protocol to align with that of UniAD. Our analysis shows that, under a unified evaluation protocol, the gap in L2 errors between VAD and UniAD narrows significantly (previous $\delta=1.03\text{-}0.72$, current $\delta=0.9474\text{-}0.9086$) while UniAD shows better performance in collision rates. This contrasts with conclusions drawn from directly comparing the reported results (UniAD* and VAD*) in prior work and highlights the importance of the consistency of evaluation methodologies.

Evaluation Methodology	Methods	Collision Rates (%) ↓						L2 (m) ↓									
		0.5s	1.0s	1.5s	2.0s	2.5s	3.0s	Ave _{1,2,3s}	Ave _{all}	0.5s	1.0s	1.5s	2.0s	2.5s	3.0s	Ave _{1,2,3s}	Ave _{all}
VAD evaluation protocol	VAD* [14]	-	0.07	-	0.17	-	0.40	0.21	-	-	0.41	-	0.70	-	1.05	0.72	-
	VAD	-	0.07	-	0.17	-	0.40	0.21	-	-	0.4084	-	0.6980	-	1.0511	0.7192	-
+ Remove averaging over time	VAD	0.10	0.04	0.16	0.39	0.61	1.11	0.51	0.40	0.2788	0.5380	0.8239	1.1513	1.5322	1.9821	1.2238	1.0511
+ Frame masking strategy	VAD	0.08	0.03	0.13	0.35	0.52	0.95	0.44	0.34	0.2633	0.4959	0.7439	1.0189	1.3296	1.6849	1.0665	0.9227
+ Excluding pedestrians	VAD	0.05	0.02	0.15	0.28	0.48	0.85	0.38	0.30	0.2633	0.4959	0.7439	1.0189	1.3296	1.6849	1.0665	0.9227
= UniAD evaluation protocol	UniAD* [12]	-	0.05	-	0.17	-	0.71	0.31	-	-	0.48	-	0.96	-	1.65	1.03	-
	UniAD ²	0.02	0.05	0.08	0.13	0.30	0.61	0.27	0.20	0.3509	0.5465	0.7691	1.0294	1.3269	1.6605	1.0788	0.9472
Remove noise in the first frame	VAD	0.05	0.02	0.10	0.20	0.37	0.58	0.27	0.22	0.2181	0.4084	0.6179	0.8577	1.1359	1.4956	0.9086	0.7830
	UniAD	0.02	0.02	0.03	0.03	0.15	0.46	0.17	0.12	0.3170	0.4774	0.6664	0.8947	1.1646	1.4703	0.9474	0.8317

¹ VAD* and UniAD* (rows shown in grey) denote the numbers reported in the original papers, which closely match our re-production efforts and demonstrate the validity of our analysis.

² Our UniAD reproduction is re-trained after fixing the bug reported here: <https://github.com/OpenDriveLab/UniAD/issues/92>.

rectly comparing results and numbers in prior work, we undertake a thorough analysis of existing evaluation methodologies, primarily those used in UniAD [12], VAD [13], and AD-MLP [30]. Surprisingly, our analysis reveals notable inconsistencies in these implementations, including:

- (1) *Averaging over the time dimension*: Taking the L2 error at the 3-second horizon for example, UniAD computes the L_2^{3s} by averaging over N samples in the *val* set, whereas VAD and AD-MLP compute this error by averaging over both samples and time intervals³:

$$L_2^{3s} = \frac{1}{N} \sum \frac{1}{6} \sum_{t \in \{0.5s, 1s, 1.5s, 2s, 2.5s, 3s\}} L_2^t, \quad (1)$$

resulting in significantly smaller L2 errors and collision rates for VAD and AD-MLP compared to UniAD;

- (2) *Filtering agents*: On one hand, UniAD excludes pedestrians from the GT occupancy map, leading to lower collision rates compared to VAD and AD-MLP. On the other hand, UniAD includes invisible objects in every frame, making the evaluation more challenging;
- (3) *Frame masking strategy*: VAD and AD-MLP exclude data clips if any one of the frames is invalid in the data sequence, while UniAD includes such clips but assigns a zero error to those invalid frames. The inclusion of these clips with frames assigned with zero error can reduce the overall error rate in the evaluation;
- (4) *Random noise in the first frame*: State-of-the-art end-to-end AV stacks use temporal information in inputs, and as a result have inferior performance in the first frame due to zero initialization of input features and ego vehicle’s state. AD-MLP addresses this by excluding the first two frames in their evaluation protocol whereas UniAD and VAD do not, leading to artificially higher errors in the evaluation of UniAD and VAD.

While some of these evaluation inconsistencies might appear minor at the first glance, our empirical analysis in Table

³This issue is concurrently being discussed on the GitHub repository: <https://github.com/hustvl/VAD/issues/33>.

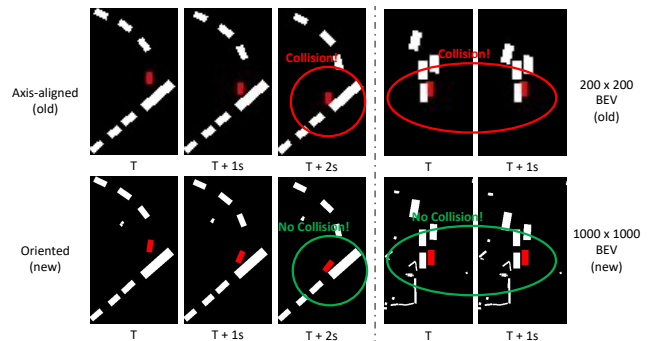


Figure 3. False positive errors triggered by axis-aligned ego vehicle bounding boxes and coarse BEV discretization. We visualize the ego vehicle in red with the GT trajectories. (Left) Without using the oriented bounding box, it is easy to trigger false positive collisions with nearby objects in the turning scenarios. (Right) When vehicles stop for the red light, a false positive collision can be triggered using a coarse BEV discretization. Our standardized evaluation resolves both of these issues.

1 reveals that each of them can significantly skew comparative analysis of prior art. This is particularly pertinent given the structure of the nuScenes dataset, which consists of relatively short data clips—typically only 40 frames each. Excluding even a few frames in each data clip can lead to a performance change ranging from 5% to 10%. Consequently, establishing a standardized evaluation methodology is crucial to ensure trustworthy comparisons.

The Missing Pieces. Besides the inconsistencies, we uncover several issues in Fig. 3 in existing evaluation methodologies that can also significantly distort the analysis:

- (1) *Axis-aligned ego vehicle representation*: Existing evaluation of calculating collision rates can generate artificial false positives and negatives due to the neglect of the ego vehicle’s orientation in the bounding box;
- (2) *Coarse BEV discretization*: The prevalent use of a 200×200 resolution grid (0.5 meters per grid cell) strikes a good balance between model performance and efficiency for training. However, this low resolution can in-

Table 2. Step-by-step analysis of how our standardized evaluation methodology improves over the existing evaluation protocol. First, without using the oriented ego-vehicle bounding box and finer-resolution BEV discretization, we find that there is a significant amount of collision rates even with GT trajectories, undermining the validity of the evaluation. Also, despite that AD-MLP achieved strong performance in L2 in the *val* set, its performance significantly drops (*very intuitively*) in collision rates in the targeted scenario evaluation, and is also weak in the map compliance rates, justifying the need to incorporate these metrics.

Scenarios	Evaluation Methodology	Methods	Col. ↓		L2 ↓		Map Comp. (%) ↓	
			Ave _{all}	Ave _{all}	Offroad	OffLane		
<i>val</i>	UniAD - 1st frame	GT	0.38 ⁴	-	-	-	-	-
		UniAD	0.20	0.8317	-	-	-	-
	+ Ego's orientation	GT	0.32	-	-	-	-	-
		UniAD	0.12	0.8317	-	-	-	-
	+ Finer discretization	GT	0.00	-	-	-	-	-
		UniAD	0.08	0.8317	-	-	-	-
		+ Pedestrians = Our standardized	GT	0.00	-	-	-	-
		UniAD	0.40	0.8317	0.91	1.74	-	-
		VAD	0.30	0.7830	1.03	1.93	-	-
		AD-MLP ⁵	0.20	0.5568	1.21	2.45	-	-
<i>targeted</i>	Our standardized	UniAD	0.15	0.9935	-	-	-	-
		VAD	0.34	1.0840	-	-	-	-
		AD-MLP	0.94	0.9360	-	-	-	-

⁴ Non-negative collision rates calculated with GT trajectories reflect false positive collisions. The collision rates of AV stacks are not directly comparable to those of GT (rows in grey) because frames with GT collisions are excluded in AV stack evaluation.

⁵ We re-implemented AD-MLP based on our framework as the released model checkpoint along with the data file is trained with GT data leakage by the time of our submission. Reference: <https://github.com/E2E-AD/AD-MLP/issues/4>.

roduce artificial false positive collisions in evaluations, even with GT trajectories, also pointed out in [30].

- (3) *Map compliance metrics* are crucial complements to L2 errors and collision rates. Our findings indicate that omitting the online mapping network can degrade qualitative results (*e.g.*, driving off the lane or road) without necessarily impacting L2 and collision rates;
- (4) *Targeted Scenario Evaluation*: Given that the majority of driving logs involve “going straight”, even simple models [30] without perception can perform extremely well in the L2 metric. Therefore, it’s crucial to conduct the evaluation in a subset of the dataset involving complex scenarios such as turning and lane changing.

Concretely, our map compliance metrics calculate the off-road rates and off-lane rates, with the latter measuring whether each of the predicted waypoints matches to the same lane as the corresponding GT trajectories. For targeted scenario evaluation, we exclude frames with a command of “keep forward”, which results in a total of 686 challenging key-frames on the nuScenes *val* set.

To measure how each evaluation change could impact the planning performance, we summarize the results in Table 2. First, looking at the performance of the collision rates in the 1st block, we find that even the GT trajectories have a high collision rate, *e.g.*, 0.38% in UniAD evaluation protocol, undermining the robustness of evaluation. After we switched to using an oriented box and finer-resolution BEV grid, we reduced the false positive collision rates to 0% for GT trajectories. After adding pedestrians in the consider-

Table 3. We conduct ablation of the inter-module connectivity to analyze its impact on module placement and information flow. The number in the front of each row refers to the edge in Fig. 4. TTO refers to test-time optimization in planning based on occupancy. Map, Mot, Occ, and Plan refer to the mapping, motion prediction, occupancy prediction, and planning modules respectively.

Methods	Col. Rates (%) ↓				L2 (m) ↓			
	1.0s	2.0s	3.0s	Ave _{all}	1.0s	2.0s	3.0s	Ave _{all}
UniAD	0.32	0.29	0.73	0.40	0.48	0.89	1.47	0.83
(1) UniAD - “Map-Mot” query	0.07	0.27	0.56	0.30	0.42	0.80	1.36	0.75
(2) UniAD - “Occ-Plan” TTO	0.00	0.10	0.61	0.16	0.36	0.81	1.41	0.74
(1) + (2) Baseline	0.00	0.07	0.51	0.13	0.24	0.55	1.07	0.53
(1) Baseline + “Map-Pred” query, filtered	0.00	0.07	0.75	0.19	0.37	0.70	1.29	0.68
(3) Baseline - “Mot-Occ”	0.00	0.05	0.56	0.14	0.26	0.57	1.08	0.54
(4) Baseline - “Mot-Plan”	0.00	0.12	0.72	0.20	0.28	0.61	1.13	0.57
(5) Baseline + “Map-Plan”, BEV	0.00	0.09	0.48	0.12	0.27	0.60	1.12	0.56
(5) Baseline + “Map-Plan”, query	0.00	0.15	0.68	0.20	0.29	0.62	1.13	0.58
(6) Baseline + “Occ-Plan”, query	0.00	0.03	0.37	0.09	0.30	0.68	1.23	0.63
(6) Baseline + “Occ-Plan”, BEV	0.00	0.07	0.56	0.14	0.24	0.55	1.07	0.52
(7) Baseline + “Map-Mot”, BEV	0.07	0.68	1.70	0.63	0.47	0.98	1.77	0.92

ation of collision rate evaluation, we arrived at our standardized evaluation methodology. One interesting finding is that the performance of AD-MLP, despite strong in L2 in the *val* set, is significantly worse in collision rates in the *targeted* scenarios, and is also relatively weak on the map compliance error rates. This is intuitive because AD-MLP only relies on ego vehicle’s states and past trajectories, and does not have perception of surrounding objects and map elements, which can cause significant safety issues. This justifies the need to include targeted scenario evaluation and map compliance rates in our standardized evaluation.

Combining these findings, our standardized evaluation effectively resolves the inconsistency between existing evaluation methodologies, addresses the false positives in collision rates, and also incorporates map compliance rates and targeted scenario evaluation, providing a more robust evaluation framework on nuScenes. Unless otherwise mentioned, we will base subsequent experiments on our standardized evaluation. Our implementation of the standardized evaluation methodology will be released along with our approach to help future research in this direction.

3.3. Exploration of the Design Space

We build our framework based on UniAD [12] for two primary reasons: (1) UniAD includes the most extensive range of tasks and modules; (2) UniAD has strong planning performance to start with. To determine module placements and information flow, we need to identify which of the inter-module dependencies is useful. To that end, we first conduct a systematic ablation on all the inter-module connections in UniAD and summarize the results in Table 3.

Module Placements. Given the existing four inter-module connectivities (*i.e.*, edge (1)(2)(3)(4)), we first observe that removing edges (1)(2) in Fig. 4 could in fact lead to more robust performance as shown in row 1-4 in Table 3. For (1), we find it is because UniAD uses the lane query features from the 1st-stage mapping head, which are noisy, therefore

Table 4. Ablation on information flow when the BEV feature maps are not used in planning. High-dimensional latent queries tend to carry more information for planning performance in this case.

Methods	Col. Rates (%) ↓				L2 (m) ↓			
	1.0s	2.0s	3.0s	Ave _{all}	1.0s	2.0s	3.0s	Ave _{all}
Baseline	0.00	0.07	0.51	0.13	0.24	0.55	1.07	0.53
Baseline - BEV ⁶	3.41	8.09	7.91	5.88	2.83	5.37	7.61	4.66
(4) Baseline - BEV + "Mot-Plan" bbox	0.00	0.29	3.94	0.97	0.34	1.15	2.53	1.10
(4) Baseline - BEV + "Mot-Plan" query	0.00	0.10	0.46	0.14	0.24	0.58	1.09	0.54
(5) Baseline - BEV + "Map-Plan" BEV	0.12	1.07	3.44	1.22	0.95	1.82	2.63	1.59
(5) Baseline - BEV + "Map-Plan" query	0.02	0.20	0.65	0.20	0.28	0.62	1.15	0.58
(6) Baseline - BEV + "Occ-Plan" BEV	0.48	1.75	4.84	1.85	1.96	3.75	5.41	3.26
(6) Baseline - BEV + "Occ-Plan" query	0.14	0.27	1.00	0.38	0.42	0.82	1.47	0.78

⁶ This model also removes the edge (4) and we omit for simplicity, *i.e.*, the only input to planning is the high-level command. We aim to use this model to serve as the basis to validate which information added can help the most in planning, in the case of not using the BEV features.

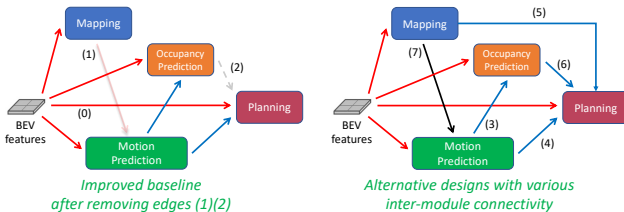


Figure 4. We highlight all inter-module connectivities that build a conditional dependency between modules. For example, edge (1) refers to using outputs from the mapping module for motion prediction, making this part of the network a sequential design. (Left) We find that removing edges (1)(2) can lead to more robust planning performance, which we call “improved baseline”. (Right) Ablation on various inter-module connectivities on top of the improved baseline does not further improve the performance.

removing interaction between lane and motion queries improves the performance. For edge (2), it is because the test-time optimization (TTO) is not in the training process, and it tends to generate trajectories that deviate from human driving logs. In our supplementary material, we present visualizations showing that TTO often results in zigzag-like trajectories near multiple objects, thereby increasing the L2 error and cannot guarantee to avoid collision. After eliminating (1)(2), we established an improved baseline, as shown in Fig. 4 (left), achieving better planning performance.

Interestingly, based on the improved baseline, we show that removing and adding other inter-module connections does not improve the planning performance in the rest of Table 3. In fact, adding edge (6) with query feature and edge (7) with compact semantic BEV map decreases the performance. Also, for edge (1), we experiment with less noisy queries from the final-stage transformer after filtering. Although better performance is achieved compared to the case using noisy query features, but adding this edge still leads to slightly worse performance than the improved baseline. This leads to our observation that we can eliminate all these edges while maintaining the same performance as the improved baseline and reducing the module dependency.

Information Flow. In Table 3, we also compare the planning performance by passing different representations to the planning module for the same edge. For example, in rows

Table 5. Ablation experiments on the necessity of modules.

Scenarios	Methods	Col. ↓	L2 ↓	Map Comp. (%) ↓	
		Ave _{all}	Ave _{all}	Offroad	OffLane
val	Baseline - edge (4) ⁷	0.20	0.5734	0.32	1.20
	Baseline - Map.	0.16	0.5332	0.71	3.03
	Baseline - Occ.	0.64	0.8174	4.19	4.43
	Baseline - Mot.	0.60	0.8561	4.05	4.49
val	Baseline - Occ. + "Mot-Plan"	0.14	0.5483	0.38	0.82
	Baseline - Mot. + "Occ-Plan"	0.25	0.5953	0.80	1.33

⁷ Similar to Table 4 row 2, where we eliminate the edge (4) as the base model to fairly compare the impact of removing each upstream module on planning.

8-9 for edge (5), we pass either the compact semantic BEV map denoting the road and lane geometry information or the latent queries of the map elements to downstream planning. Similarly, in rows 10-11 for edge (6), we compare the use of compact BEV occupancy maps or object queries for planning. Interestingly, despite that passing the compact BEV outputs leads to slightly better performance than passing the high-dimensional query features, we find that it is not necessary to pass either the compact outputs or the per-module latent queries due to the potential redundancy with information already flowing from the BEV feature maps into the planning module via the edge (0).

To further validate the information flow on what representation should be used in the downstream planning, we now eliminate the use of BEV feature maps to planning – edge (0) – on top of the improved baseline and compare the performance of adding edge (4)(5)(6) in Table 4. Comparing rows 1-2, we confirm again that removing the BEV features leads to a significant performance drop in the end-to-end planning. In the subsequent experiments, we observed that despite imposing dependency between upstream modules with planning, passing the latent queries with high dimensional information tends to bring a stronger performance increase compared to using the compact output representations alone as inputs to planning.

Necessity of Modules. We summarize the results in Table 5. Since we removed edge (4) from the baseline, the planning module only uses the BEV feature maps and does not condition other upstream modules. In this case, we observe that the removal of any auxiliary task could lead to a significant performance drop on the planning task. In particular, despite removing the online mapping task does not lead to higher L2 errors and collision rates, the map compliance errors are increased by a large margin. These experiments justify the need for all these modules for proper co-training of the BEV features toward a parallel design. Interestingly, we also find that the occupancy prediction task and motion prediction task are indeed somewhat redundant toward a sequential or hybrid design. Specifically, if we add the edge (4) between motion prediction and planning in the case of removing the occupancy task, we can recover the performance by explicitly passing the query features from upstream modules. Similar observations are also observed in the case of removing the motion prediction task.

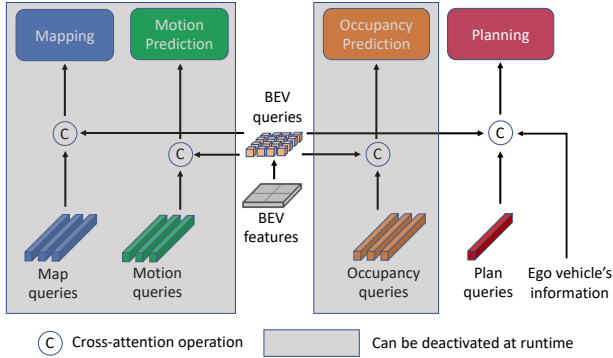


Figure 5. **PARA-Drive** architecture. Perception, prediction, and planning modules are co-trained in parallel. No dependency is introduced between modules and information passing between modules is conducted implicitly via the tokenized BEV query features. Runtime speed can be boosted by deactivating modules in grey.

Key Insights. We summarize our observations below:

- (1) Upstream modules designed for a similar task but with different representations tend to be redundant in sequential or hybrid designs. However, in more parallel-oriented designs, these different modules demonstrate complementary benefits to planning and are crucial to co-training of the BEV features;
- (2) Similar state-of-the-art performance achieved in hybrid designs can also be achieved with a parallel design. However, the architecture design becomes more intricate with sequential connection in hybrid designs due to the potential negative impacts of dependencies between particular modules on overall performance;
- (3) Passing high-dimensional queries yields better results than passing the compact outputs in hybrid designs, especially in the case of not using BEV features in planning. Surprisingly, if the BEV feature map is learned properly via co-training, using the BEV feature alone in planning can lead to state-of-the-art performance.

4. PARA-Drive

Integrating the above insights, we introduce PARA-Drive, a parallelized modular AV stack that encompasses a diverse set of modules for the co-training of BEV features. This BEV feature map, in conjunction with the data from the ego vehicle (*e.g.*, high-level commands, CAN bus, history trajectories), forms the exclusive input to the planning head.

PARA-Drive is illustrated in Fig. 5. In particular, it contains four modules: online mapping, tracking and motion prediction, occupancy prediction, motion planning, all of which are co-trained in parallel. Inspired by [12, 13], we equip each module with its own set of learnable query features tailored to its tasks. PARA-Drive processes a sequence of camera images as inputs to construct both current and historical BEV features. Through cross-attention, the query features of each module interact with the BEV features, en-

sureing that relevant information is captured for each task.

A distinct advantage of PARA-Drive, setting it apart from prior work with hybrid designs [12, 13, 25], lies in its *operational independence* of the planning module. Post co-training, the planning head operates independently of other perception and prediction modules. This allows for significant flexibility during inference: modules such as mapping, motion, and occupancy can be deactivated or run at a reduced frame rate only when needed for interpretability, conducting safety checks, or user display. Consequently, this design significantly boosts the runtime efficiency of the motion planning module, enabling more frequent re-planning and thereby enhancing overall safety for deployment.

Backbone and BEV features. Following [16], we keep one frame of historical BEV features in memory to perform cross-attention to obtain temporal information. We primarily use the R50⁸ backbone which we find sufficient to achieve state-of-the-art planning performance.

Online Mapping. We use Panoptic Segformer [17] and treat mapping as a pixel-wise segmentation task. In particular, the output is a 4-channel semantic BEV map. Each channel represents the likelihood of pixels being categorized into one of four classes: road boundary, lane divider, pedestrian crossing area, and drivable area. To optimize the mapping head, we employ a combination of L1 loss, Dice loss, and GIoU loss in order to learn both the bounding box and the pixel-wise mask of each map element.

Motion and Occupancy Prediction are conceptually similar but differ in their output representation – the former focuses on sparse object-level outputs, while the latter generates a scene-level probabilistic BEV occupancy map. Following prior work [12, 20, 27], we employ self-attention between query features to facilitate interaction between agents, in addition to applying cross-attention with the BEV features. For both modules, we match the query features with GT using the Hungarian algorithm and apply the negative log-likelihood, Dice, and binary cross-entropy losses for training. Following [12], the motion prediction module also uses bounding boxes of tracked objects and their latent features as inputs, and therefore we consider tracking-prediction as a whole module and have omitted a separate tracking head in Fig. 5 for simplicity of illustration.

Motion Planning. In addition to the optional use of CAN bus data, our motion planning module employs high-level commands and a learnable query feature. The high-level command selects the feature embedding corresponding to the appropriate driving behavior mode and is then concatenated with the planning query. Following cross-attention with the BEV feature map, multi-layer perceptrons (MLPs) are employed to regress the planned future trajectory.

⁸Specifically, we use the released **BEVformerV2-t1** as our backbone.

Table 6. Main comparison with state-of-the-art approaches under our standardized evaluation methodology on the nuScenes *val* dataset. PARA-Drive achieves superior performance consistently on all metrics and targeted scenarios.

Scenarios	Evaluation Methodology	Methods	Using Ego's States	Collision Rates (%) ↓					L2 (m) ↓				Map Comp. Errors (%) ↓		
				1.0s	2.0s	3.0s	Ave _{1,2,3s}	Ave _{all}	1.0s	2.0s	3.0s	Ave _{1,2,3s}	Ave _{all}	Offroad	Offlane
<i>val</i>	Standardized	UniAD [12]	No	0.32	0.29	0.73	0.45	0.40	0.4774	0.8947	1.4703	0.9474	0.8317	0.91	1.74
		VAD [13]	No	0.02	0.26	0.83	0.37	0.30	0.4084	0.8577	1.4596	0.9086	0.7830	1.03	1.93
		PARA-Drive	No	0.00	0.12	0.65	0.26	0.17	0.2581	0.5927	1.1196	0.6568	0.5574	0.12	0.83
		AD-MLP [30]	Yes	0.00	0.14	0.70	0.28	0.20	0.2267	0.5847	1.1782	0.6632	0.5568	1.21	2.45
		PARA-Drive+	Yes	0.00	0.09	0.49	0.19	0.13	0.2035	0.5195	1.0425	0.5885	0.4939	0.11	0.78
<i>targeted</i>	Standardized	UniAD [12]	No	0.00	0.00	0.73	0.24	0.15	0.4698	1.0921	1.9162	1.1594	0.9935	-	-
		VAD [13]	No	0.00	0.29	0.87	0.39	0.34	0.5305	1.2015	2.0696	1.2672	1.0840	-	-
		PARA-Drive	No	0.00	0.00	0.72	0.24	0.14	0.3844	0.9729	1.8805	1.0793	0.9082	-	-
		AD-MLP [30]	Yes	0.00	0.58	3.62	1.40	0.94	0.3309	0.9938	2.0559	1.1269	0.9360	-	-
		PARA-Drive+	Yes	0.00	0.00	0.29	0.10	0.05	0.2856	0.7575	1.4608	0.8346	0.7018	-	-

Table 7. Comparison of performance on perception and prediction.

Methods	Detection and Motion Prediction				Mapping		Occupancy	
	mAP ↑	NDS ↑	AMOTA ↑	minADE ↓	IoU-lane ↑	IoU-road ↑	IoU-n ↑	VPQ-n ↑
UniAD, R101 [12]	0.38	0.50	0.36	0.73	0.30	0.67	62.3	52.8
Ours, R101	0.37	0.48	0.35	0.72	0.33	0.71	63.6	55.6
Ours, R50	0.32	0.44	0.30	0.89	0.32	0.70	58.9	51.4

4.1. Additional Results and Analysis

Table 6 summarizes the performance of all recent state-of-the-art approaches. On our standardized evaluation protocols, PARA-Drive sets the new state-of-the-art on all metrics. Also, our results re-established the benchmark for a fair comparison across prior art. Given the new benchmark, although we observed that VAD still outperforms UniAD on the *val* set for the collision and L2 errors, UniAD seems to perform better mapping and achieved a lower map compliance error rate, as well as lower errors in the targeted evaluation with more complex scenarios.

Using Ego Vehicle’s States. Prior work [13, 30] has shown that using CAN bus (velocity, acceleration, angular velocity, etc.) and history trajectories can improve planning. We also observed this for PARA-Drive, especially in targeted evaluation. However, in the case of using BEV features in planning, with proper co-training, we find that the improvements brought by the CAN bus and history trajectories become marginal in the *val* set, as shown in Table 6.

Also, prior work [30] has claimed that the use of only ego vehicle’s information alone (without any information from the input images and perception) can achieve state-of-the-art planning performance. Although we have similar observations in Table 6 in the *val* set, especially for the collision rates and L2 errors. We find that, in the *targeted* scenarios as well as the map compliance error rates, AD-MLP has significantly worse performance than PARA-Drive. This suggests that the open-loop evaluation scheme is still very informative and emphasizes again the importance of our standardized and enhanced evaluation methodology.

Performance on Perception and Prediction. We summarize the results in Table 7. Surprisingly, we find that the performance of other modules besides planning in PARA-Drive is on par or slightly better than UniAD when using the same R101 backbone, even though PARA-Drive is a parallelized architecture. This suggests that the co-training of

Table 8. Performance on old existing evaluation protocols

Eval.	Methods	Col. Rates (%) ↓				L2 (m) ↓			
		1.0s	2.0s	3.0s	Ave _{123s}	1.0s	2.0s	3.0s	Ave _{123s}
UniAD	GT	0.35	0.38	0.35	0.36	-	-	-	-
	VAD	0.02	0.28	0.85	0.38	0.50	1.02	1.68	1.07
	UniAD	0.05	0.17	0.71	0.31	0.48	0.96	1.65	1.03
	Ours	0.07	0.25	0.60	0.30	0.40	0.77	1.31	0.83
VAD	GT	1.02	0.96	0.91	0.96	-	-	-	-
	VAD	0.07	0.17	0.41	0.22	0.41	0.70	1.05	0.72
	UniAD	0.12	0.13	0.28	0.17	0.48	0.74	1.07	0.76
	Ours	0.14	0.23	0.39	0.25	0.25	0.46	0.74	0.48

BEV features for planning together with other tasks might not necessarily lead to the negative transfer issue as observed in prior work [9, 19] for co-training of perception-only tasks or with different CNN-based architectures.

Performance on Existing Evaluation Methodologies. We summarize the results based on UniAD and VAD evaluation methodologies in Table 8. Despite the fact that the results are noisy and less robust due to false positive collisions (even with GT) in using these evaluation methodologies, PARA-Drive still outperforms prior art consistently.

Runtime Speed. As we can deactivate all modules besides the backbone and planning head, PARA-Drive achieved a 2.77× speed up compared to UniAD-base, with the compute primarily spent on the backbone. If switched to a more lightweight one, e.g., R50-tiny, we can achieve a near real-time speed. We believe that our model can be potentially optimized for embedded devices for real-time deployment.

Qualitative Results. Please refer to the supp. materials.

5. Conclusions and Limitations

Our work contributes to the rapidly evolving field of end-to-end AV by conducting a systematic analysis of the design space in the high-level architecture, offering insights into the necessity of modules, their placements, and the information flow between modules. These insights have led to the development of PARA-Drive, a novel, fully parallel AV architecture, which not only achieves state-of-the-art performance in perception, prediction, and planning but also significantly accelerates inference speed. Despite promising, the results are currently limited to the open-loop setting, for which we are working towards expanding the experiments to the closed-loop setting in simulation.

References

- [1] Ben Agro, Quinlan Sykora, Sergio Casas, and Raquel Urtasun. Implicit Occupancy Flow Fields for Perception and Prediction in Self-Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2
- [2] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars. *arXiv preprint arXiv:1604.07316*, 2016. 2, 3
- [3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multi-modal Dataset for Autonomous Driving. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2020. 2, 3
- [4] Sergio Casas, Abbas Sadat, and Raquel Urtasun. MP3: A Unified Model to Map, Perceive, Predict and Plan. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2021. 2
- [5] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by Cheating. In *Conf. on Robot Learning*, 2019. 2
- [6] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. NEAT: Neural Attention Fields for End-to-End Autonomous Driving. In *IEEE Int. Conf. on Computer Vision*, 2021. 3
- [7] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2022. 2
- [8] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-End Driving via Conditional Imitation Learning. In *Proc. IEEE Conf. on Robotics and Automation*, 2018. 2, 3
- [9] Michael Crawshaw. Multi-Task Learning with Deep Neural Networks: A Survey. *arXiv preprint arXiv:2009.09796*, 2020. 8
- [10] Peiyun Hu, Aaron Huang, John Dolan, David Held, and Deva Ramanan. Safe Local Motion Planning with Self-Supervised Freespace Forecasting. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2021. 2, 3
- [11] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. ST-P3: End-to-End Vision-Based Autonomous Driving via Spatial-Temporal Feature Learning. In *European Conf. on Computer Vision*, 2022. 2
- [12] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-Oriented Autonomous Driving. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2023. 1, 2, 3, 4, 5, 7, 8
- [13] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized Scene Representation for Efficient Autonomous Driving. *IEEE Int. Conf. on Computer Vision*, 2023. 1, 2, 3, 4, 7, 8
- [14] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD GitHub issue regarding the error averaged over the time dimension, 2023. 4
- [15] Tarasha Khurana, Peiyun Hu, Achal Dave, Jason Ziglar, David Held, and Deva Ramanan. Differentiable Raycasting for Self-supervised Occupancy Forecasting. In *European Conf. on Computer Vision*, 2022. 2, 3
- [16] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. BEVFormer: Learning Bird’s-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers. *European Conf. on Computer Vision*, 2022. 7
- [17] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, Ping Luo, and Tong Lu. Panoptic SegFormer: Delving Deeper into Panoptic Segmentation with Transformers. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2022. 7
- [18] Xiwen Liang, Yangxin Wu, Jianhua Han, Hang Xu, Chungjing Xu, and Xiaodan Liang. Effective Adaptation in Multi-Task Co-Training for Unified Autonomous Driving. In *Conf. on Neural Information Processing Systems*, 2022. 2
- [19] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird’s-Eye View Representation. In *Proc. IEEE Conf. on Robotics and Automation*, 2023. 8
- [20] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion Forecasting via Simple & Efficient Attention Networks. *arXiv preprint arXiv:2207.05844*, 2022. 7
- [21] Dean A Pomerleau. Alvin: An Autonomous Land Vehicle in a Neural Network. In *Conf. on Neural Information Processing Systems*, 1988. 2
- [22] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-Modal Fusion Transformer for End-to-End Autonomous Driving. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2021. 2, 3
- [23] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, and Raquel Urtasun. Perceive, Predict, and Plan: Safe Motion Planning Through Interpretable Semantic Representations. In *European Conf. on Computer Vision*, 2020. 2
- [24] Hao Shao, Letian Wang, Ruobing Chen, Steven L. Waslander, Hongsheng Li, and Yu Liu. ReasonNet: End-to-End Driving with Temporal and Global Reasoning. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2023. 3
- [25] Chonghao Sima, Wenwen Tong, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, and Hongyang Li. Scene as Occupancy. *IEEE Int. Conf. on Computer Vision*, 2023. 1, 2, 3, 7
- [26] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-Guided Control Prediction for End-to-End Autonomous Driving: A Simple yet Strong Baseline. *Conf. on Neural Information Processing Systems*, 2022. 2, 3

- [27] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting. In *IEEE Int. Conf. on Computer Vision*, 2021. 7
- [28] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-End Interpretable Neural Motion Planner. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019. 2
- [29] Wenyuan Zeng, Shenlong Wang, Renjie Liao, Yun Chen, Bin Yang, and Raquel Urtasun. DSDNet: Deep Structured Self-Driving Network. In *European Conf. on Computer Vision*, 2020. 2, 3
- [30] Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the Open-Loop Evaluation of End-to-End Autonomous Driving in nuScenes. *arXiv preprint arXiv:2305.10430*, 2023. 3, 4, 5, 8
- [31] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. BEVerse: Unified Perception and Prediction in Birds-Eye-View for Vision-Centric Autonomous Driving. *arXiv preprint arXiv:2205.09743*, 2022. 2