
End-to-End Autonomous Driving without Costly Modularization and 3D Manual Annotation

Anonymous Author(s)

Affiliation

Address

email

A Supplementary Material

The supplementary material presents additional designing and explaining details of our Unsupervised pretext task for end-to-end Autonomous Driving (UAD) in the manuscript.

- **Different Partition Angles**

We explore the influence of different partition angles in angular pretext to learn better spatio-temporal knowledge.

- **Different Direction Thresholds**

We explore the influence of different thresholds in direction prediction to enhance planning robustness in complex driving scenarios.

- **Different Backbones and Pre-trained Weights**

We compare the performance of different backbones and pre-trained weights on our method.

- **Objectness Label Generation with GT Boxes**

We compare the generated objectness label between using the pseudo ROIs from GroundingDINO [10] and ground-truth boxes on different backbones.

- **Settings for ROI Generation**

We ablate different settings for the open-set 2D detector GroundingDINO, which provides ROIs for the label generation of angular perception pretext.

- **Different Image Sizes and BEV Resolution**

We compare the performance with different input sizes of multi-view images and BEV resolutions.

- **Runtime Analysis**

We evaluate the runtime of each module of UAD and compare with modularized UniAD [6], which demonstrates the efficiency of our method.

- **Classification of Angular Perception**

We evaluate the objectness prediction in the angular perception pretext, which demonstrates the enhanced perception capability in complex driving scenarios.

- **Influence of Pre-training**

We evaluate the influence of pre-training by detailing the training losses and planning performances with different pre-trained weights.

- **More Visualizations**

We provide more visualizations for the predicted angular-wise objectness and planning results in the open-loop evaluation of nuScenes [1] and closed-loop simulation of CARLA [3].

A.1 Different Partition Angles

The proposed angular perception pretext divides the BEV space into multiple sectors. We explore the influence of partition angle θ in Tab 1. Experimental results show that the L2 error and inference

Table 1: Ablation on different partition angles in the proposed angular pretext.

#	Partition Angle	L2 (m) ↓				Collision (%) ↓				FPS
		1s	2s	3s	Avg.	1s	2s	3s	Avg.	
①	1°	0.35	0.78	1.42	0.85	0.01	0.28	0.68	0.32	5.0
②	2°	0.34	0.77	1.46	0.86	0.01	0.22	0.48	0.24	6.3
③	4°	0.39	0.81	1.50	0.90	0.01	0.12	0.43	0.19	7.2
④	8°	0.38	0.85	1.55	0.93	0.01	0.18	0.55	0.25	7.7
⑤	15°	0.47	0.94	1.69	1.03	0.03	0.20	0.60	0.28	8.1
⑥	30°	0.48	1.00	1.75	1.08	0.05	0.28	0.63	0.32	8.4

Table 2: Ablation on different thresholds of direction prediction in the directional augmentation.

#	Threshold (m)	L2 (m) ↓				Collision (%) ↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
①	0.5	0.35	0.79	1.43	0.86	0.03	0.18	0.71	0.31
②	0.8	0.35	0.77	1.46	0.86	0.01	0.12	0.68	0.27
③	1.2	0.39	0.81	1.50	0.90	0.01	0.12	0.43	0.19
④	1.5	0.40	0.82	1.52	0.91	0.02	0.15	0.42	0.20
⑤	2.0	0.38	0.85	1.55	0.93	0.01	0.08	0.48	0.19

Table 3: Ablation on different backbones and pre-trained weights.

#	Backbone	Pretrained Weight	L2 (m) ↓				Collision (%) ↓				FPS
			1s	2s	3s	Avg.	1s	2s	3s	Avg.	
①	Res50	None	0.43	0.94	1.65	1.01	0.03	0.37	0.86	0.42	9.6
②		ImageNet	0.41	0.90	1.66	0.99	0.03	0.32	0.80	0.38	
③	Res101	None	0.40	0.87	1.59	0.95	0.02	0.23	0.59	0.28	7.2
④		ImageNet	0.37	0.84	1.53	0.91	0.01	0.18	0.50	0.23	
⑤		COCO	0.36	0.83	1.51	0.90	0.01	0.16	0.45	0.21	
⑥		NuImages	0.39	0.81	1.50	0.90	0.01	0.12	0.43	0.19	

speed gradually increase with the partition angle. The model with partition angle of 1° (①) achieves the best average L2 error of 0.85m. And the partition angle of 4° contributes to the best average collision rate of 0.19% (③). This reveals that a smaller partition angle helps learn more fine-grained environmental representations, eventually benefiting planning. In contrast, the model with a large partition angle sparsely perceives the scene. Despite reducing the computation cost, it will also degrade the safety of the end-to-end autonomous driving system.

A.2 Different Direction Thresholds

The direction prediction that the ego car intends to maneuver (*i.e.*, *left*, *straight* and *right*) is proposed to enhance the steering capability for autonomous driving. The label is generated with the threshold δ (see Eq. 7 in the manuscript), which determines the ground-truth direction of each waypoint in the expert trajectory. Here we explore the influence by ablating different thresholds, as shown in Tab. 2. Experimental results show that the L2 error gradually increases with the direction threshold. The model with δ of 0.5m (①) achieves the lowest L2 error of 0.86m. It reveals that a smaller threshold will force the planner to fit the expert navigation, leading to a closer distance between the predicted trajectory and the ground truth. In contrast, the collision rate benefits more from larger thresholds. The model with δ of 2.0m obtains the best collision rate at 2s of 0.08% (⑤), showing the effectiveness for robust planning. Notably, the threshold of 1.2m contributes to a great balance with the average L2 error of 0.90m and average collision rate of 0.19%.

A.3 Different Backbones and Pre-trained Weights

As a common sense, pre-training the backbone network with fundamental tasks like image classification on ImageNet [2] will benefit the sub-tasks. The previous method UniAD [6] uses the pre-trained weights of BEVFormer [8]. What surprised us is that when replacing the pre-trained weights with the one learned on ImageNet, the performance of UniAD dramatically degraded (see “Influence of Pre-training” for more details). This inspires us to explore the influence of backbone settings on our framework. As shown in Tab. 3, interestingly, even without any pre-training, our model still outperforms UniAD with pre-trained ResNet101 and VAD with pre-trained ResNet50. This verifies the effectiveness of our unsupervised pretext task on modeling the driving scenes. We also use publicly available pre-trained weights on detection datasets like COCO [9] and nuImages [1] to train our model, which shows better performance. These experimental results and observations demonstrate that a potentially promising topic is *how to pre-train a model for end-to-end autonomous driving*. We leave this to future research.

Table 4: Ablation on 2D object boxes in pretext label generation.

#	Backbone	2D Object Box	L2 (m) ↓				Collision (%) ↓				FPS
			1s	2s	3s	Avg.	1s	2s	3s	Avg.	
①	Res50	Pseudo	0.41	0.90	1.66	0.99	0.03	0.32	0.80	0.38	9.6
②		GT	0.41	0.87	1.61	0.96	0.03	0.30	0.71	0.35	
③	Res101	Pseudo	0.39	0.81	1.50	0.90	0.01	0.12	0.43	0.19	7.2
④		GT	0.37	0.79	1.45	0.84	0.01	0.13	0.39	0.18	

Table 5: Ablation on the settings of ROI generation. The Conf. Thresh denotes the confidence threshold in GroundingDINO [10] to filter unreliable predictions. *vehicle, pedestrian, barrier* represent the used prompt words to obtain ROIs of corresponding classes. Rule Filter indicates filtering the ROIs that are more than half of the length or width of the image.

#	Conf. Thresh	Prompt Words	Rule Filter	L2 (m) ↓				Collision (%) ↓			
				1s	2s	3s	Avg.	1s	2s	3s	Avg.
①	0.35	{ <i>vehicle</i> }	-	0.48	0.98	1.75	1.07	0.08	0.38	0.80	0.42
②	0.35	{ <i>vehicle, pedestrian</i> }	-	0.47	0.94	1.69	1.03	0.04	0.27	0.71	0.34
③	0.35	{ <i>vehicle, pedestrian, barrier</i> }	-	0.43	0.88	1.60	0.97	0.03	0.23	0.60	0.29
④	0.35	{ <i>vehicle, pedestrian, barrier</i> }	✓	0.39	0.81	1.50	0.90	0.01	0.12	0.43	0.19
⑤	0.30	{ <i>vehicle, pedestrian, barrier</i> }	✓	0.39	0.82	1.45	0.89	0.01	0.21	0.51	0.24
⑥	0.40	{ <i>vehicle, pedestrian, barrier</i> }	✓	0.46	0.90	1.57	0.98	0.01	0.13	0.37	0.17

67 A.4 Objectness Label Generation with GT Boxes

68 As mentioned in the manuscript, the essence of generating the angular objectness label lies in the
69 2D ROIs, which come from the open-set 2D detector GroundingDINO [10]. Here we explore the
70 influence of using the ground-truth 2D boxes as ROIs, which provide more high-quality samples for
71 the representation learning in the angular perception pretext. Tab. 4 shows that training with GT boxes
72 achieves consistent performance gains on both ResNet50 [4] and ResNet101 [4] (②,④ v.s. ①,③). This
73 reveals that accurate annotation does help to learn better spatio-temporal knowledge and improve ego
74 planning. Considering the cost in real-world deployment, training with accessible pseudo labels is a
75 more efficient way compared with the manual annotation, which also shows comparable performance
76 in autonomous driving (① v.s. ② and ③ v.s. ④).

77 A.5 Settings for ROI Generation.

78 The quality of learned spatio-temporal knowledge highly relies on the generated ROIs by the open-set
79 2D detector GroundingDINO [10], which are then projected as the BEV objectness label for training
80 the angular perception pretext. We explore the influence of generated ROIs with different settings,
81 as shown in Tab. 5. We take the setting with the confidence score of 0.35, prompt word of *vehicle*
82 and without the Rule Filter, as the baseline (①). By appending more prompt words (*e.g.*, *pedestrian*,
83 *barrier*), the planning performance gradually improves (③,② v.s.①), showing the enhanced perception
84 capability with more diversified objects. Filtering the ROIs with overlarge size (*i.e.*, Rule Filter)
85 brings considerable gains for the average L2 error of 0.07m and average collision rate of 0.10%
86 (④ v.s.③). One interesting observation is that decreasing the confidence threshold would slightly
87 improve the L2 error while causing higher collision rate (⑤ v.s.④). In contrast, increasing the threshold
88 obtains lower average collision rate of 0.17% and higher average L2 error of 0.98m. This reveals the
89 importance of providing diversified ROIs for angular perception learning as well as ensuring high
90 quality. The model with the confidence score of 0.35, all prompt words and Rule Filter achieves
91 balanced performance with the average L2 error of 0.90m and average collision rate of 0.19%.

92 A.6 Different Image Sizes and BEV Resolution

93 For safe autonomous driving, increasing the input size of the multi-view images and the resolution
94 of the built BEV representation is an effective way, which provide more detailed environmental
95 information. While benefiting perception and planning, it inevitably brings heavy computation cost.
96 We then ablate the image size and BEV resolution of our UAD to find a balanced version between
97 performance and efficiency, as shown in Tab. 6. The results show that our UAD with ResNet-101 [4],

Table 6: Comparison with different backbones, image sizes and BEV resolutions.

#	Method	Backbone	Image Size	BEV Resolution	1s	L2 (m) ↓			Avg.	1s	Collision (%) ↓			Avg.	FPS
①	UniAD [6]	R101	1600×900	200×200	0.48	0.96	1.65		1.03	0.05	0.17	0.71		0.31	2.1
②	VAD-Tiny [7]	R50	640×360	100×100	0.60	1.23	2.06		1.30	0.33	1.33	2.21		1.29	17.6
③	VAD-Base [7]	R50	1280×720	200×200	0.54	1.15	1.98		1.22	0.10	0.24	0.96		0.43	5.3
④	UAD (Ours)	R50	640×360	100×100	0.47	0.99	1.71		1.06	0.08	0.39	0.90		0.46	18.9
⑤	UAD (Ours)	R50	1600×900	200×200	0.41	0.90	1.66		0.99	0.03	0.32	0.80		0.38	9.6
⑥	UAD (Ours)	R101	1600×900	200×200	0.39	0.81	1.50		0.90	0.01	0.12	0.43		0.19	7.2

Table 7: Module runtime comparison between UniAD [6] and our UAD. The inference is measured on an NVIDIA Tesla A100 GPU.

Model Partition	UniAD			UAD (Ours)		
	Module	Latency (ms)	Proportion (%)	Module	Latency (ms)	Proportion (%)
Feature Extraction	Backbone	38.1 ± 0.5	8.2%	Backbone	36.0 ± 0.3	26.0%
	BEV Encoder	83.4 ± 0.5	17.9%	BEV Encoder	81.5 ± 0.4	58.9%
Sub-Task	Det&Track	145.3 ± 1.3	31.2%	Angular Partition	1.1 ± 0.1	0.8%
	Map	92.1 ± 0.7	19.8%	Dreaming Decoder	18.2 ± 0.2	13.2%
	Motion	50.6 ± 0.6	10.9%			
	Occupancy	45.9 ± 0.4	9.9%			
Prediction	Planning Head	9.7 ± 0.3	2.1%	Planning Head	1.5 ± 0.1	1.1%
Total	-	465.1 ± 4.3	100%	-	138.3 ± 1.1	100.0%

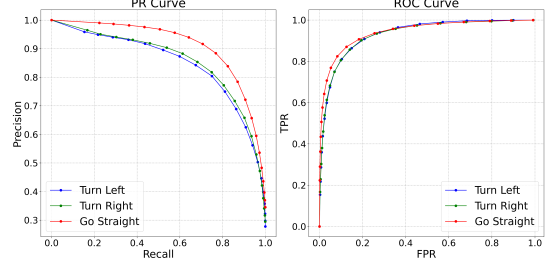


Figure 1: Visualization of the PR and ROC curves for the angular-wise objectness prediction in different driving scenes.

image size of 1600×900, BEV resolution of 200×200, achieves the best performance compared with previous methods UniAD [6] and VAD-Base [7] while running faster with 7.2FPS (⑥). By replacing the backbone with ResNet-50, our UAD is more efficient with little performance degradation (⑤ v.s. ⑥). We further align the settings of VAD-Tiny, which has an inference speed of outstanding 17.6FPS (②), to explore the influence of much smaller input sizes. Tab. 6 shows that our UAD still achieves excellent performance even compared with VAD-Base of high-resolution inputs (④ v.s. ③). Notably, our UAD of this version has the fastest inference speed of 18.9FPS. This again proves the effectiveness of our method in performing fine-grained perception, as well as the robustness to fit the inputs of different sizes.

A.7 Runtime Analysis

Tab. 7 compares the runtime of each module between the modularized method UniAD [6] and our UAD. As we adopt the Backbone and BEV Encoder from BEVFormer [8] that are the same in UniAD, the latency of feature extraction is similar with little difference due to different pre-processing. The modular sub-tasks in UniAD consume most of the runtime, *i.e.*, significant 71.8% for Det&Track (31.2%), Map (19.8%), Motion (10.9%) and Occupancy (9.9%), respectively. In contrast, our UAD performs simple Angular Partition and Dreaming Decoder, which take only 14.0% (19.3ms) to model the complex environment. This demonstrates our insight that it's a necessity to liberate end-to-end autonomous driving from costly modularization. The downstream Planning Head takes negligible 1.5ms to plan the ego trajectory, compared with 9.7ms in UniAD. Finally, our UAD finishes the inference with a total runtime of 138.3ms, 3.4× faster than the 465.1ms of UniAD, showing the efficiency of our design.

A.8 Classification of Angular Perception

The proposed angular perception pretext learns spatio-temporal knowledge of the driving scene by predicting the objectness of each sector region, which is supervised by the generated binary angular-wise label. We show the perception ability by evaluating the classification metrics based on the validation split of the nuScenes [1] dataset. Fig. 1 draws the Precision-Recall (PR) curve and Receiver-Operating-Characteristic (ROC) curve in different driving scenes (*i.e.*, *turn left*, *go straight* and *turn right*). In the PR curve, our UAD achieves balanced precision and recall scores in different driving scenes, showing the effectiveness of our pretext task to perceive the surrounding objects. Notably, the performance of *go straight* scenes is slightly better than the steering ones under all thresholds. This proves our insight to design tailored direction-aware learning strategy for improving

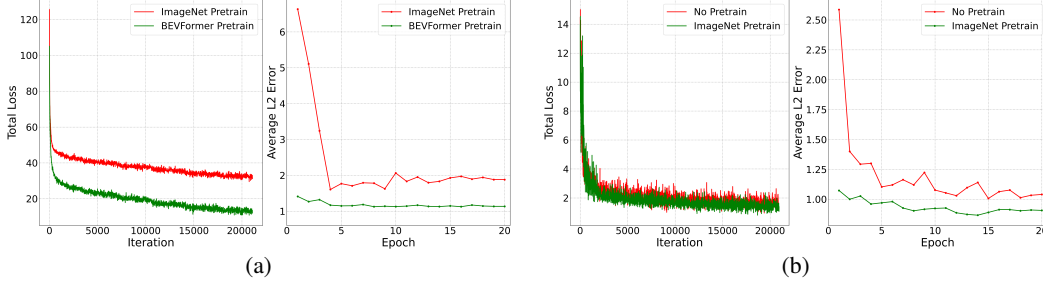


Figure 2: Optimization of UniAD (a) and our UAD (b) with different pre-trained backbone weights.

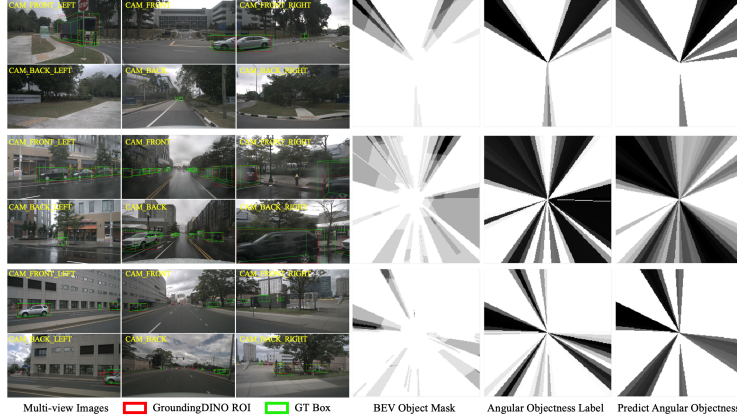


Figure 3: Visualization of the angular perception.

the safety-critical *turn left* and *turn right* scenes. The ROC curve shows the robustness of our angular perception pretext to classify the objects from complex environmental observations.

A.9 Influence of Pre-training

Pre-training the backbone network with fundamental tasks is a commonly used metric to benefit representation learning. As mentioned in “Different Backbones and Pre-trained Weights” of Sec. 4.4 in the manuscript, the performance of the previous SOTA method UniAD [6] dramatically degrades without the pre-trained weights from BEVFormer [8]. Here we further detail the influence by comparing the training losses and planning performances with different pre-trained weights in Fig. 2. Fig. 2a shows that the training losses increase by about 20 on average when replaced with the pre-trained weights from ImageNet [2]. Correspondingly, the average L2 error is significantly higher than the one with the pre-trained weights from BEVFormer. This reveals that UniAD heavily relies on the perceptive pre-training in BEVFormer to optimize modularized sub-tasks. In contrast, our UAD performs comparably even without any pre-training (see Fig. 2b), proving the effectiveness of our designs for robust optimization.

A.10 More Visualizations

Open-loop Planning We provide more visualizations about the predicted angular-wise objectness and planning results on nuScenes [1]. Fig. 3 compares the discrete objectness scores and ground truth, proving the effectiveness of our angular perception pretext to perceive the objects in each sector region. The planning results of previous SOTA methods (*i.e.*, UniAD [6] and VAD [7]) and our UAD are shown in Fig. 4. With the designed pretext and tailored training strategy, our method could plan a more reasonable ego trajectory under different driving scenarios, proving the effectiveness of our work. The third row shows the failure case of our planner. In this case, the ego car is given the “*Turn Right*” command when $t=0$ (*i.e.*, the first frame of the driving scenario), leading to ineffectiveness of our planner in learning helpful temporal information. A possible solution to deal with this is to apply an auxiliary trajectory prior for the first several frames, and we leave this to future work.

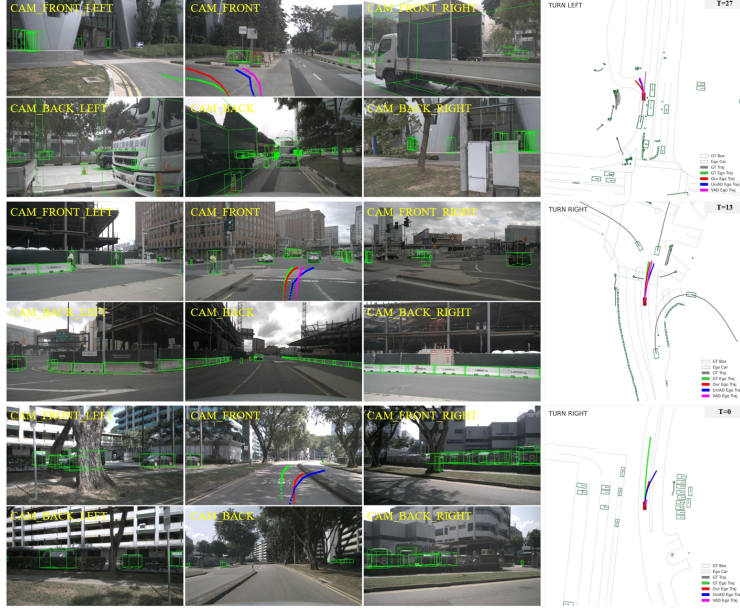


Figure 4: Visualization of the planning results. The first two rows show the success of our method in safe planning in complex scenarios, while the third row exhibits a failure case of our planner when no temporal information could be acquired when $t=0$.

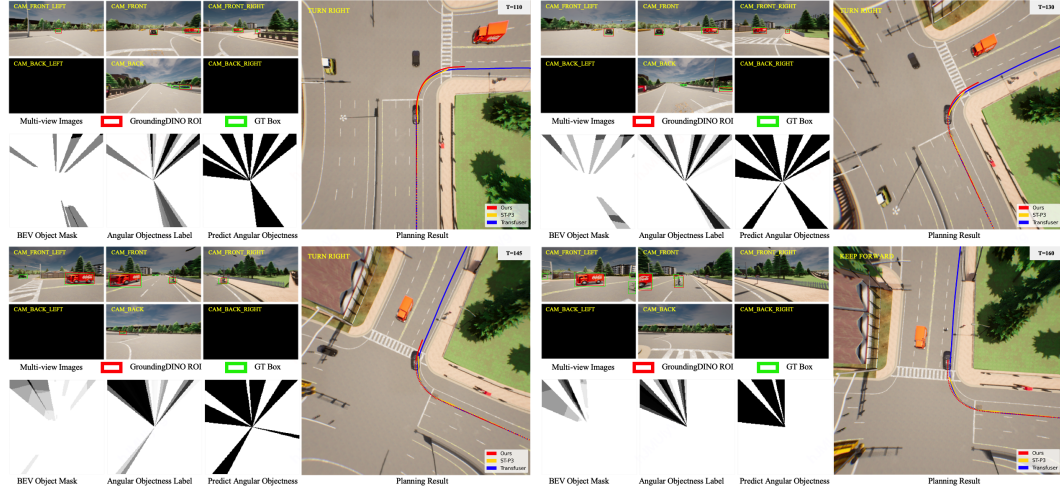


Figure 5: Visualization of angular perception and planning in Carla.

Closed-loop Simulation Fig. 5 visualizes the predicted objectness and planning results in the Town05 Long benchmark of CARLA [3]. Following the setting of ST-P3 [5] in closed-loop evaluation, we collect visual observations from the cameras of “CAM_FRONT”, “CAM_FRONT_LEFT”, “CAM_FRONT_RIGHT” and “CAM_BACK”. It shows that the sector regions in which the surrounding objects exist are successfully captured by our UAD, proving the effectiveness and robustness of our design. Notably, the missed objects by GroundingDINO [10], *e.g.*, the black car in the camera of “CAM_FRONT_LEFT” at $t = 145$, are surprisingly perceived and marked in the corresponding sector. This demonstrates our method has the capability of learning perceptive knowledge in a data-driven manner, even with coarse supervision by the generated 2D pseudo boxes from GroundingDINO.

References

- [1] 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 multimodal dataset for autonomous driving. In *CVPR*, 2020.
- [2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*. Ieee, 2009.
- [3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, 2017.
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [5] 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 *ECCV*. Springer, 2022.
- [6] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, 2023.
- [7] 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. *arXiv preprint arXiv:2303.12077*, 2023.
- [8] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*. Springer, 2022.
- [9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*. Springer, 2014.
- [10] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.