

Supplementary Material for ETA: Efficiency through Thinking Ahead, A Dual Approach to Self-Driving with Large Models

Anonymous ICCV submission

Paper ID 5250

Abstract

In this supplementary material, we first review related work on using LLMs/VLMs in driving (Section 1). Next, we provide experimental details to ensure reproducibility (Section 2). Finally, we present additional quantitative and qualitative results in Section 3, including the comparison of models according to abilities and an exploration of alternative models by varying the large encoder in the base model.

1. Additional Related Work

LLMs/VLMs for Driving: There is a large increase in the use of LLMs/VLMs for driving. A line of work, represented by approaches like DriveMLM [12], LLM4AD/CarLLaVA [8], and FeD [16], uses LLMs for initialization to benefit from their pre-training on large datasets. A typical approach is to encode the scene using smaller models and feed the encoded visual features into the LLM to predict action. For fast inference, especially in closed-loop evaluation, these approaches prefer small language models, e.g., as small as 50M in [8]. In this line of work, LLMs are not fully utilized, i.e., for their reasoning capabilities. As reported in [8], the improvement mostly comes from the large-scale pre-training of the vision encoder rather than the LLM on top.

Another line of work, including approaches like DriveVLM [11] and DriveLM [9], aims to benefit from the reasoning capabilities of LLMs, e.g. with Chain-of-thought (CoT) reasoning. This kind of approach has the potential to fully utilize the power of LLMs for driving, e.g. by improving out-of-distribution cases with their pre-trained knowledge. However, multi-step reasoning causes high latency, restricting these models to open-loop evaluation. Similarly, our goal is to benefit from the pre-training knowledge of LLMs as much as possible, but differently, without sacrific-

ing real-time performance in closed-loop evaluation.

Other Uses of LLMs for Driving: Besides action, previous work also uses an LLM to detect hard cases in motion prediction [14] or to align features of a driving model with the features of an LLM [7]. Another work analyzes the capabilities of LLMs in terms of future prediction [10].

2. Experimental Details

Base Large Model: For the Large Model, we use the CLIP-ViT-L-336px encoder from the checkpoint of LLaVA 1.6 [6] 7B Vicuna, which has 24 layers with a hidden dimension size of 1024 and an input patch size of 14.

Small Model: For the Small Model, we use the first 8 layers of the same CLIP-ViT-L-336px encoder as the Base Large Model.

Action Prediction Model: For the Action Prediction Model, we use the autoregressive Llama architecture initialized from scratch, with 12 hidden layers and a hidden size of 768.

Forecasting Model: For the Forecasting Model, we use a lightweight transformer encoder with 2 layers. We concatenate the input features with the conditioning along the feature dimension, project first to the feature dimension size, then pass to the transformer encoder.

Training Setting: We use the Deepspeed zero2 setting during training for memory optimization. We use bfloat16 during training, and do not use any weight decay.

Data Bucketing: To increase the frequency of interesting data samples, we divide samples to buckets as in [8]. Additionally, we weight the buckets to oversample some buckets. The buckets we design are:

- **Acceleration from scratch:** The agent is currently stationary (speed < 0.05) and starting to accelerate.
- **Light acceleration:** The agent is slightly accelerating (0.2 < throttle < 0.5). We weight this bucket with a

Table 1. **Comparison to SOTA according to Distinct Abilities.** We compare our async model to state-of-the-art methods across distinct abilities, highlighting significant improvements in all challenging scenarios except traffic sign (T. Sign) over previous best results.

Method	Merging \uparrow	Overtaking \uparrow	E. Brake \uparrow	Give Way \uparrow	T. Sign \uparrow	Mean \uparrow	Latency \downarrow
AD-MLP [15]	0.00	0.00	0.00	0.00	0.00	0.00	3
UniAD-Base [1]	12.16	20.00	23.64	10.00	13.89	15.94	663.4
VAD [5]	7.14	20.00	16.36	20.00	20.22	16.75	278.3
TCP-traj [13]	12.50	22.73	52.72	40.00	46.63	34.92	83
ThinkTwice [3]	13.72	22.93	52.99	50.00	47.78	37.48	762
DriveTransformer-Large [4]	17.57	35.00	48.36	40.00	52.10	38.60	211.7
DriveAdapter [2]	14.55	22.61	54.04	50.00	50.45	38.33	931
Base Model (Ours)	35.24	54.70	66.67	80.00	59.43	59.21	120
Async Model (Ours)	26.66	50.42	60.13	80.00	43.64	52.17	50

weight of 2.

- Medium acceleration: The agent is slightly accelerating ($0.5 < \text{throttle} < 0.9$). We weight this bucket with a weight of 2.
- Strong acceleration: The agent is slightly accelerating ($\text{throttle} > 0.9$).
- Braking: The agent is braking.
- Coasting: The agent is at a non-zero speed, is not braking, and is not accelerating strongly ($\text{throttle} < 0.2$).
- Steering Left: The agent steering to the left. We weight this bucket with a weight of 3.
- Steering Right: The agent steering to the right. We weight this bucket with a weight of 3.
- Rear Vehicle Hazard: Assuming all agents maintain speed and heading, there is at least one other vehicle that will hit the ego vehicle from behind (rear 60 degrees).
- Front Vehicle Hazard: Assuming all agents maintain speed and heading, there is at least one other vehicle that will collide with the ego vehicle from the front (front 60 degrees).
- Side Vehicle Hazard: Assuming all agents maintain speed and heading, there is at least one other vehicle that will collide with the ego vehicle from either side (left and right, 120 degrees each).
- Stop Sign: The ego vehicle is currently within the range of a stop sign.
- Red Light: The ego vehicle is currently within the range of a red traffic light.
- Swerving Bucket: The current route contains a scenario which is one of:
 - Accident
 - BlockedIntersection
 - ConstructionObstacle
 - HazardAtSideLane
 - ParkedObstacle
 - VehicleOpensDoorTwoWays
- Pedestrian: Assuming all pedestrians maintain speed and heading, there is at least one pedestrian that will be hit by

the ego vehicle.

3. Additional Experiments

3.1. Ability Scores

We include the per-ability scores for all models in Table 1.

3.2. Additional Ablations

We include two additional ablations in Table 3 in addition to the ablations included in the main paper.

Action Mask Loss: We ablate the inclusion of the action mask loss **E** for the Base Model. Compared to the Async Model setting **C** in the main paper (Table 2), the decrease in the Driving Score is relatively lower.

Different Small Model: We experiment with using ViT-Base as the small model (**E**), despite it not being fast enough to meet real-time constraints. We find it less suitable as the small model, with Driving Score decreasing significantly (57.07 vs. 69.53).

Table 2. **Varying the Large Encoder in the Base Model.** We experiment with various sizes of large encoders in the base model to observe the differences when changing our default large encoder in **A** to smaller encoders (**B–E**) and larger encoders (**F, G**). While larger encoders generally perform better, smaller encoders achieve lower latencies, as expected. Notably, the model in **B** has a similar latency to our Async model but performs significantly worse (61.30 vs. 69.53). **Lat.** stands for latency in milliseconds.

ID	Model	Size	DS \uparrow	SR \uparrow (%)	Lat. \downarrow
A	CLIP-ViT-L	308M	74.33	48.94	120
B	CLIP-ViT-L8	114M	61.30	35.00	50
C	InternViT	304M	71.53	46.82	148
D	ViT-Base	86.9M	67.53	44.55	90
E	EVA02-Base	85.8M	65.20	37.73	70
F	CLIP ViT-H	631M	70.74	43.64	174
G	CLIP-ViT-g	1011M	73.19	45.91	235

Table 3. **Ablation Study.** We conduct an ablation study by removing each component of the model (A–C). Additionally, we report the performance of the small model alone (D) and further analyze forecasting by using ground truth features during both training and testing (E) or only during testing (F). ‘GT’ indicates ground truth. See the text for a detailed analysis.

ID	Method	DS \uparrow	SR \uparrow (%)	Efficiency \uparrow	Comfort \uparrow	Latency \downarrow
	Base Model (Ours)	74.33	48.33	186.04	25.77	120
	Async Model (Ours)	69.53	38.64	184.51	28.43	50
A	Without Forecasting	54.92	30.45	177.52	18.71	50
B	Without Small Model	42.49	17.27	170.94	14.10	30
C	Base Model Without Action Mask	70.02	87.26	178.37	29.97	120
D	Async Model Without Action Mask	42.67	17.27	167.32	27.99	50
E	ViT-Base as Small Model	57.07	27.73	175.58	35.18	90
F	Small Model Only	61.30	35.00	183.63	43.91	50
G	GT Forecasting	74.12	48.18	192.72	24.76	120
H	GT Forecasting (Test Time)	60.72	32.27	151.53	17.25	120

3.3. Additional Qualitative Analysis

We present two additional qualitative examples: one comparing the Async model to the version without the small model (B) in Fig. 1, and another examining lane change behavior between the Base model and the Async model in Fig. 2.

References

- [1] 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 *CVPR*, 2023. 2
- [2] Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. DriveAdapter: breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *ICCV*, 2023. 2
- [3] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *CVPR*, 2023. 2
- [4] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. DriveTransformer: unified transformer for scalable end-to-end autonomous driving. In *ICLR*, 2025. 2
- [5] 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. In *ICCV*, 2023. 2
- [6] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 1
- [7] Chenbin Pan, Burhaneddin Yaman, Tommaso Nesti, Abhirup Mallik, Alessandro G Allievi, Senem Velipasalar, and Liu Ren. VLP: vision language planning for autonomous driving. In *CVPR*, 2024. 1
- [8] Katrin Renz, Long Chen, Ana-Maria Marcu, Jan Hünermann, Benoit Hanotte, Alice Karnsund, Jamie Shotton, Elahe Arani, and Oleg Sinavski. CarLLaVA: Vision language models for camera-only closed-loop driving, 2024. 1
- [9] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. DriveLM: Driving with graph visual question answering. In *ECCV*, 2024. 1
- [10] Shiva Sreeram, Tsun-Hsuan Wang, Alaa Maalouf, Guy Roman, Sertac Karaman, and Daniela Rus. Probing multimodal LLMs as world models for driving, 2024. 1
- [11] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Zhiyong Zhao, Yang Wang, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. DriveVLM: The convergence of autonomous driving and large vision-language models. In *CoRL*, 2024. 1
- [12] Wenhai Wang, Jiangwei Xie, ChuanYang Hu, Haoming Zou, Jianan Fan, Wenwen Tong, Yang Wen, Silei Wu, Hanming Deng, Zhiqi Li, Hao Tian, Lewei Lu, Xizhou Zhu, Xiaogang Wang, Yu Qiao, and Jifeng Dai. DriveMLM: Aligning multimodal large language models with behavioral planning states for autonomous driving, 2023. 1
- [13] 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. In *NeurIPS*, 2022. 2
- [14] Yi Yang, Qingwen Zhang, Kei Ikemura, Nazre Batool, and John Folkesson. Hard cases detection in motion prediction by vision-language foundation models. pages 2405–2412, 2024. 1
- [15] 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 nuscenets, 2023. 2
- [16] Jimuyang Zhang, Zanming Huang, Arijit Ray, and Eshed Ohn-Bar. Feedback-guided autonomous driving. In *CVPR*, 2024. 1



Figure 1. **Turn Failure.** In the left column, we show the initial state of the scene. In this scenario, the agent is performing a right turn at an intersection. In the middle, the Async Model can maneuver properly, performing a correct turn. On the right, the model without the small encoder (**B**) is unable to perform a smooth turn, performing a tighter turn resulting in a crash.

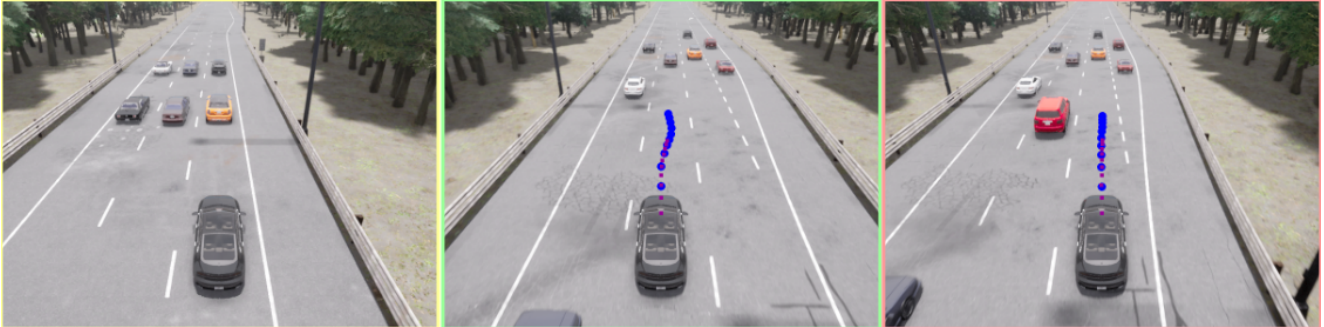


Figure 2. **Lane Change - Base model vs. Async.** In the left column, we show the initial state of the ego vehicle in traffic. In this scenario, the ego vehicle is required to switch the left lane. In the middle, the Large model is able to correctly make the lane change while avoiding collision. Interestingly, the Async model on the right ignores the requested lane change and continues driving straight. Due to the leaderboard evaluation method, both models get a perfect driving score in this scenario.