2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Unveiling the Surprising Efficacy of Navigation Understanding in End-to-End Autonomous Driving

Zhihua Hua[1,3], Junli Wang[3,4], Pengfei Li[3], Qihao Jin[1,3], Bo Zhang[2], Kehua Sheng[2], Yilun Chen[3], Zhongxue Gan[1*], and Wenchao Ding[1*]

***Abstract*— Global navigation information and local scene understanding are two crucial components of autonomous driving systems. However, our experimental results indicate that many end-to-end autonomous driving systems tend to over-rely on local scene understanding while failing to utilize global navigation information. These systems exhibit weak correlation between their planning capabilities and navigation input, and struggle to perform navigation-following in complex scenarios. To overcome this limitation, we propose the Sequential Navigation Guidance (SNG) framework, an efficient representation of global navigation information based on real-world navigation patterns. The SNG encompasses both navigation paths for constraining long-term trajectories and turn-by-turn (TBT) information for real-time decision-making logic. We constructed the SNG-QA dataset, a visual question answering (VQA) dataset based on SNG that aligns global and local planning. Additionally, we introduce an efficient model SNG-VLA that fuses local planning with global planning. The SNG-VLA achieves state-of-the-art performance through precise navigation information modeling without requiring auxiliary loss functions from perception tasks. Project page: SNG-VLA**

I. Introduction

In recent years, end-to-end autonomous driving systems have garnered significant attention from researchers [1], [2]. The end-to-end paradigm simplifies traditional modular systems, is better suited to data-driven training approaches, and demonstrates enhanced generalization performance [3].

Global Navigation information plays a pivotal role in end-to-end autonomous driving systems [4], providing essential directional references for trajectory planning. Unlike prediction [5], which generates multimodal trajectory forecasts, planning requires explicit navigation inputs to produce deterministic driving trajectories [6], [7]. Despite the critical role of global navigation information in end-to-end autonomous driving systems, we have made a surprising observation: *removing or corrupting navigation information in existing end-to-end driving methods minimally affects planning performance and, in some cases, even improves performance*. For instance, as illustrated in Fig. 1, our experiments with the Transfuser [8] on the NAVSIM [9] benchmark demonstrate

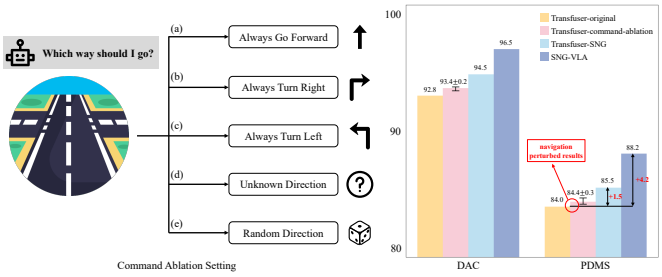


Fig. 1: We demonstrate the impact of introducing perturbations to the driving command and the effectiveness of the SNG. The command ablation experiments comprise five distinct experimental settings, with Transfuser-command-ablation representing the statistical results across all five experiments. Detailed data are presented in Table IV. The utilization of random or erroneous navigation information demonstrates minimal impact on model performance. SNG exhibits significant performance improvement over the baseline approach.

that complete removal of navigation information paradoxically yields superior results. This phenomenon contradicts basic driving logic, as one would anticipate a substantial decline in planner performance when explicit navigation information is absent. This unexpected outcome raises a critical question: *Do current end-to-end autonomous driving systems truly understand and utilize global navigation information?*

Our answer is unequivocally negative. Current research [10]–[12] predominantly employs driving commands (such as "Turn Left", "Go Forward", "Turn Right", "None") to represent global navigation information, utilizing one-hot encoding to discretize driving behaviors into finite categories. However, as shown in Fig. 2, this approach exhibits the following limitations: (1) the annotation process relies on a fixed temporal horizon or spatial intervals [6], [9], which can lead to ambiguous interpretations in complex scenarios. In the roundabout, the significant lateral displacement of the vehicle going forward caused the driving command to be incorrectly labeled as a "Turn Left". (2) this representation suffers from oversimplification. In beyond visual range (BVR) scenarios [13], a vehicle must change lanes in advance to execute a turn at a distant intersection. However, as the global navigation command is labeled as "Go Forward", it creates causal confusion in the model when encountering lane-changing behaviors present in expert trajectories. Consequently, numerous end-to-end autonomous driving systems

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62403142, and in part by the Science and Technology Commission of Shanghai Municipality under Grant 24511103100.

[1] College of Intelligent Robotics and Advanced Manufacturing, Fudan University, China {zhhua24}@m.fudan.edu.cn, {ganzhongxue, dingwenchao}@fudan.edu.cn
[2] Didi Chuxing
[3] Institute for AI Industry Research (AIR), Tsinghua University
[4] Institute of Automation, Chinese Academy of Sciences
* Corresponding authors: Wenchao Ding and Zhongxue Gan

fail to effectively utilize navigation information, and their performance likely stems from overfitting to specific input channels [14], [15].

To address these limitations and enhance the navigation semantic comprehension capabilities of end-to-end autonomous driving systems, we propose a novel paradigm of Sequential Navigation Guidance (SNG), inspired by real-world navigation patterns [16]. The SNG effectively represents navigation information by integrating static global path planning with dynamic high-level guidance: (1) Navigation Path: A predefined trajectory segment extracted from the global path, serving as a reference line for planning; (2) Real-time Turn-by-Turn (TBT) Information: A comprehensive set of high-level guidance cues comprising current driving actions with associated distance and time estimations, future actions, and corresponding supplementary actions, which collectively inform the planning process. Notably, both types of information can be conveniently acquired through navigation APIs and are readily available off-the-shelf in practical deployment.

To further align local planning with global planning, we propose SNG-QA, which imposes additional constraints on local planning within both the reasoning and action spaces. The SNG-QA dataset comprises 100K QA pairs that decompose the reasoning process into hierarchical planning components. The local planning reasoning process integrates both global planning outcomes and local scene understanding. We utilize NAVSIM [9] and its annotations to construct the SNG-QA pairs according to predefined task formats. We introduce an efficient model SNG-VLA. The model employs multi-modal fusion encoder and a unified transformer backbone, which can efficiently handle the constraints between global navigation information and local scene inputs. The model autoregressively generates textual reasoning and planning trajectories.

The proposed method demonstrates remarkable efficacy in modeling global navigation information, offering a plug-and-play solution that significantly enhances the planning capabilities of end-to-end autonomous driving systems. Our model also achieves state-of-the-art performance on both the Bench2Drive [17] a closed-loop benchmark based on Carla [18] and the NAVSIM [9] a real-world evaluation benchmark. Our contributions can be summarized as follows:

1. To address the limitations of current global navigation representation, we propose a novel Sequential Navigation Guidance approach to structure global navigation information, offering both long-term trajectory constraints and real-time decision-making logic.
2. We propose the SNG-QA dataset, which partitions the reasoning process into global and local planning components to enhance local planning rationality and ensure consistency between local and global planning.
3. We develop an effective model that, without incorporating auxiliary tasks and utilizing precise navigation information, achieves state-of-the-art (SOTA) performance in both Bench2Drive and NAVSIM benchmarks.

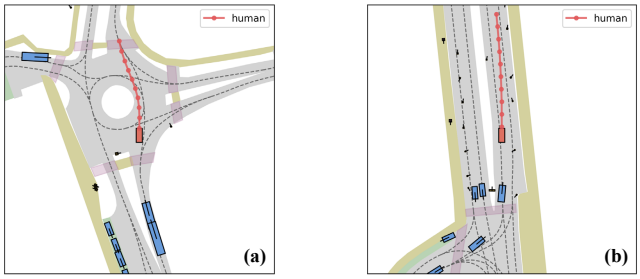


Fig. 2: We demonstrate erroneous examples in the annotation process of driving commands. (a) Incorrect annotation occurs in roundabouts due to longitudinal displacement. (b) The annotation information fails to comprehend lane-changing behavior, resulting in causal confusion.

II. Related Work

A. End-to-end autonomous driving

Traditional autonomous driving systems are often composed of multiple modular components [19], [20], whereas end-to-end autonomous driving enables a direct mapping from raw sensor data to planning trajectories [21], [22]. Most methods [23], [24] and dataset [25] utilize driving commands as global navigation information. UniAD [21] has significantly enhanced the performance of autonomous driving systems by integrating multiple modules into an end-to-end framework. VAD [6] employs a fully vectorized approach to model driving scenarios, ensuring planning safety while improving operational efficiency. BEVPlanner [15] transforms sensor inputs into BEV (Bird's Eye View) features, serving as an intermediate representation within the end-to-end architecture. GenAD [12] introduces a novel generative framework that aids planning tasks by predicting the dynamic interactions between the ego vehicle and the environment. However, driving command constrains the practical performance of these methods.

B. Multimodal large models for planning

Multimodal large models facilitate seamless interaction and understanding across diverse data types, driving transformative innovations in fields such as natural language processing and beyond [26]. Given the necessity of processing sensor data from multiple modalities and performing joint planning in autonomous driving systems, multimodal large models naturally serve as an effective backbone for such systems. DriveGPT4 [27] leverages multimodal large models to process multi-frame video and text inputs, simultaneously outputting reasoning processes during planning, thereby significantly enhancing the interpretability and interactivity of autonomous driving systems. LMDrive [28] unifies multimodal sensor data into a textual feature space, greatly improving the interactivity of autonomous driving systems and demonstrating exceptional performance in CARLA [18] closed-loop evaluation. Therefore, we propose an efficient pipeline based on a multimodal large model, capable of pro-

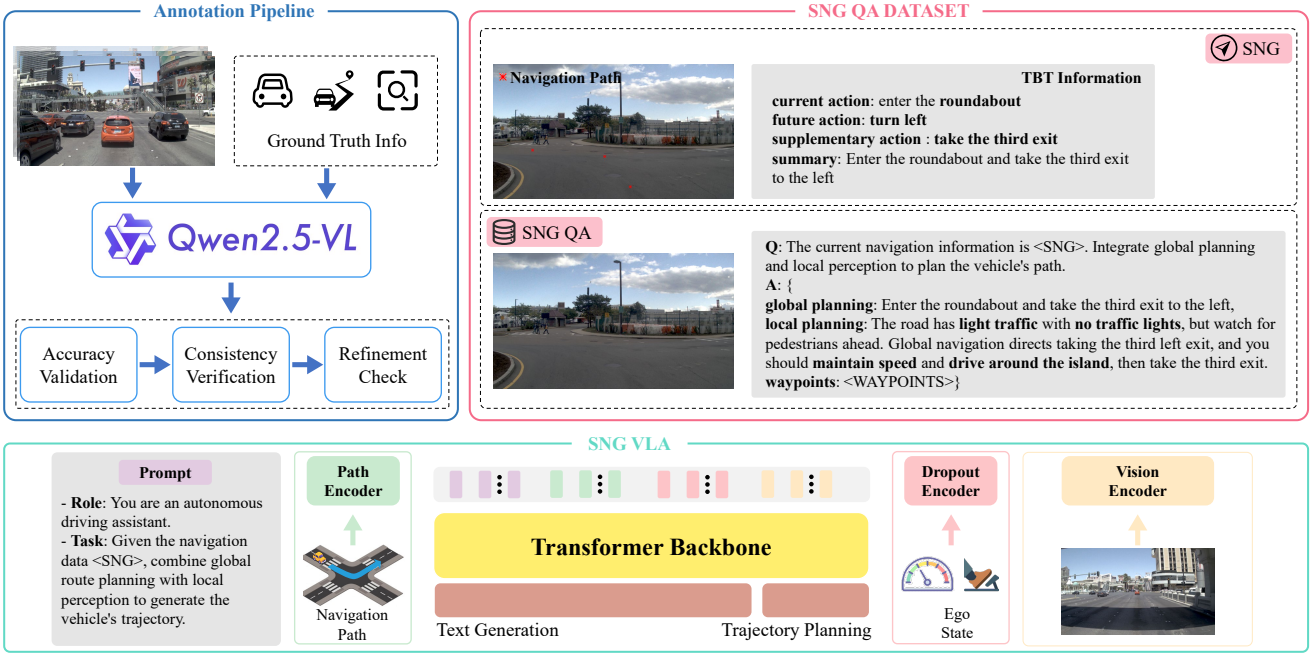


Fig. 3: Overview of our pipeline. Sequential navigation guidance is consists of navigation path and TBT information. SNG-QA comprises three subtasks: global planning, local planning, and trajectory planning. The architecture of our model is divided into two parts: the multimodal feature fusion encoder and the unified transformer backbone.

cessing data from various sensor modalities and performing end-to-end planning tasks.

C. Navigation information for planning

Navigation information plays a critical role in autonomous driving planning. Current end-to-end benchmarks primarily rely on driving commands as navigation inputs. In real-world datasets [25], [29], [30], driving commands (e.g., "Turn Left") are implicitly inferred from expert trajectories to model navigation information. Although simulators [17], [18] provide waypoints between the current position and the target location, they still use discrete driving commands as the primary input. Several methods, such as UniAD [21] and BEVPlanner [15], embed navigation commands into latent spaces as additional model inputs. ST-P3 [7] samples multiple trajectories and filters them based on geometric features aligned with driving commands, while VAD [6] generates results for all command categories and selects the corresponding trajectory as the final output. TCP [23] and TransFuser [8], [9] concatenate driving commands with ego states as conditional inputs. However, relying solely on driving commands to model navigation information leads to intent ambiguity and deviations from real-world scenarios. To address these limitations, we propose integrating TBT (Turn-by-Turn) instructions and navigation paths to model more accurate navigation information, thereby improving planning rationality, safety, and human-like interaction.

III. Methods

The pipeline of our method is shown in Fig. 3. Specifically, we first introduce the modeling approaches for sequential navigation guidance in Section III-A. Subsequently, we present the construction methodology of SNG-QA in Section III-B. In Section III-C, we detail our model architecture, which comprises a multimodal feature fusion encoder described in Section III-C.1 and a transformer-based decoder outlined in Section III-C.2.

A. Modeling Global Navigation Information

In real-world driving scenarios, most driving behaviors are guided by specific navigation information, often facilitated by tools such as Google Maps [16]. Navigation information typically comprises two key components: a pre-planned global route R, generated using A*, and real-time turn-by-turn (TBT) information I. The global route, when transformed from the world coordinate system to the vehicle coordinate system, serves as a reference line for the vehicle's direction of driving. Simultaneously, TBT information, which includes high-level textual prompts, provides immediate guidance for local maneuvers.

We construct SNG by integrating the navigation path and turn-by-turn (TBT) information, as illustrated in Fig. 3. Specifically, we select road centerlines within a 40m range ahead of each vehicle as references and sampled them to generate navigation path $P = \{(\hat{x}_1, \hat{y}_1), (\hat{x}_2, \hat{y}_2), \ldots, (\hat{x}_{N_p}, \hat{y}_{N_p})\}$, N_p represents the number of navigation points. To simulate real-world localization errors and mitigate the influence of privileged information in navigation paths on model performance, we introduced substantial noise to the sampled navigation paths.

The TBT information includes current driving actions with associated distance and time estimations, future actions,

and future supplementary action. Drawing from real-world navigation systems, we categorize driving actions into eight distinct types: turn left, turn right, execute U-turn, proceed straight, keep left, keep right, enter roundabout, and none. We employ the vehicle's route and camera data feed as inputs to VLM for predicting both current and future driving actions. The duration of each current action is calculated based on the future trajectory and instantaneous vehicle speed. We define nine categories of supplementary actions—including entering highways, tunnels, right-turn lanes, left-turn lanes, etc.—to provide contextual information for future action prediction. When the VLM identifies ambiguity in a future action description, it incorporates an appropriate supplementary action for disambiguation. The entire annotation framework is implemented using Qwen2.5 VL 72B [31].

B. SNG-QA dataset

High-quality question-answering (QA) datasets are crucial for enhancing Vision-Language-Action (VLA) models' scene understanding and reasoning capabilities [11], [32]. Driving behavior reasoning represents a key task in visual question answering for autonomous driving, playing a vital role in ensuring that models correctly interpret scenes and formulate reasonable planning strategies. However, existing VQA datasets [11], [33] typically rely on local perception and expert trajectories for annotating driving behavior inferences, thereby neglecting the role of global navigation information in the annotation process. For instance, in scenarios such as beyond visual range (BVR) lane changes or roundabout exit selections, models struggle to accurately interpret expert trajectory intentions when relying solely on local perception.

To address this limitation, we developed an automated annotation workflow based on Qwen 2.5 VL 72B [31] that divides driving behavior reasoning into three stages: global navigation information summarization, local planning, and trajectory point generation. The model first generates corresponding summaries based on the input SNG. To ensure that local planning fully comprehends both global navigation information and local scene understanding, we employ global navigation information and object detection labels as prompts to guide the model in generating causal explanations for local planning decisions. To guarantee the quality of the generated textual explanations, we implemented a three-stage validation process encompassing accuracy verification, consistency validation, and language refinement. Based on NAVSIM [9], we constructed an inference annotation dataset comprising approximately 100,000 samples.

C. SNG-VLA

We employ LLaVA [39] as the backbone, integrating a large language model, Qwen2.5 [31], and a vision encoder, SigLIP [40]. Following the method [41], we incorporate additional encoders specifically designed for the navigation path and ego state, thereby augmenting the model's capacity to process multimodal inputs. Besides, our model achieves adequate real-time performance.

1) Scene representation: For a driving scenario, the input is represented as TBT information I, navigation path $P = \{(\hat{x}_1, \hat{y}_1), (\hat{x}_2, \hat{y}_2), \ldots, (\hat{x}_{N_P}, \hat{y}_{N_p})\}$, front view images $V = (M^1, M^2, ...M^{N_M})$, and the vehicle's ego state $S_t = (v_x^t, v_y^t, a_x^t, a_y^t)$. The TBT information I is encoded into a $F_T \in \mathbb{R}^{N_T \times H}$ through LLM tokenizer, where N_T denotes the number of text tokens and H corresponds to the feature dimension of the LLM's transformer backbone. Similarly, P is encoded into $F_P \in \mathbb{R}^{N_P \times H}$ through multi-layer perception (MLP) layers. We use a pre-trained SigLIP vision encoder [40] to extract features F'_M from multi-view images, which are then projected into $F_M \in \mathbb{R}^{N' \times H}$ via a linear transformation. To mitigate overfitting on ego state inputs [14], [15], we adopt an attention-based state dropout encoder (SDE) inspired by [42], which applies dropout to each state channel and processes ego state to $F_E \in \mathbb{R}^{4 \times H}$.

2) Transformer Decoder: After obtaining the driving scenario representations, all features are concatenated with the waypoint query Q_W into F and fed as input to the transformer backbone. Following the Simlingo [33], the model first auto-regressively generates language predictions that represent the response to the task prompt. Subsequently, in an additional forward pass, it generates waypoint hidden states.

$$F = \text{Concat}(F_T, F_P, F_M, F_E, Q_W) \tag{1}$$

The hidden states output by the transformer backbone interact with the navigation query through a cross-attention module, followed by MLP layers to predict the trajectory. The loss function consists of the L1 loss between the predicted and the ground truth trajectory.

$$\mathcal{L} = \|\hat{\tau} - \tau\| \tag{2}$$

where the $\hat{\tau}$ denotes the predicted trajectory and τ denotes the future ground truth trajectory.

IV. Experiments

We evaluate our method in Bench2Drive [17], a closed-loop evaluation benchmark under CARLA Leaderboard 2.0 [18] for end-to-end autonomous driving. The base set, consisting of 1,000 clips, is used for training, while the model is evaluated on 220 official routes. Additionally, we conduct closed-loop experiments in the NAVSIM benchmark [9] to evaluate its performance in real-world scenarios.

A. Implementation Details

We employ pre-trained Qwen2.5-0.5B as the transformer backbone and pre-trained SigLIP-So400M as the vision encoder, with a patch size of 14 and an image size of 384. In the state dropout encoder (SDE), we apply a dropout rate of 0.5 to the four ego state channels. The visual inputs consist of front and rear camera images, which undergo additional downsampling after passing through the vision encoder. The SNG-QA task is employed exclusively in the NAVSIM experiment. Due to real-time constraints in CARLA and the availability of detailed scenario labels, we generate SNGs through rule-based methods in our Bench2Drive experiments

Method	Input	Navigation	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
UniAD [21]	C & L	CM	97.8	91.9	92.9	**100**	78.8	83.4
PARA-Drive [10]	C & L	CM	97.9	92.4	93.0	99.8	79.3	84.0
LTF [8]	C & L	CM	97.4	92.8	92.4	**100**	79.0	83.8
Transfuser [8]	C & L	CM	97.7	92.8	92.8	**100**	79.2	84.0
Transfuser†	C & L	SNG	97.8	94.5	93.5	**100**	80.0	85.5
DRAMA [34]	C & L	CM	98.0	93.1	**94.8**	**100**	80.1	85.5
Hydra-MDP [35]	C & L	CM	98.3	96.0	94.6	**100**	78.7	86.5
DiffusionDrive [36]	C & L	CM	98.2	96.2	94.7	**100**	82.2	88.1
SNG-VLA	C-single	SNG	**98.9**	**96.5**	92.9	**100**	**83.8**	**88.24**
SNG-VLA-QA	C-single	SNG	98.4	96.7	93.1	100	83.4	88.21

TABLE I: **Comparison on NAVSIM navtest split with closed-loop metrics**. † represents the results with SNG input. CM represents Driving Command. C-single represents front view image. PDM score (PDMS) [9] is weighted aggregation of several sub-scores: no at-fault collisions (NC), drivable area compliance (DAC), time-to-collision (TTC), comfort (Comf.), and ego progress (EP).

Method	**Open-loop Metric**	**Closed-loop Metric**				**Latency**
	Avg. L2 ↓	Driving Score ↑	Success Rate (%) ↑	Efficiency ↑	Comfortness ↑	
AD-MLP [14]	3.64	18.05	0.00	48.45	22.63	3ms
UniAD-Tiny [21]	0.80	40.73	13.18	123.92	**47.04**	420.4ms
UniAD-Base [21]	0.73	45.81	16.36	129.21	43.58	663.4ms
VAD [6]	0.91	42.35	15.00	157.94	46.01	278.3ms
DriveTransformer-Large [37]	**0.62**	63.46	35.01	100.64	20.78	211.7ms
SNG-VLA	0.82	**67.17**	**35.90**	**158.58**	22.30	159.6ms
TCP* [23]	1.70	40.70	15.00	54.26	47.80	83ms
TCP-ctrl* [23]	–	30.47	7.27	55.97	**51.51**	83ms
TCP-traj* [23]	1.70	59.90	30.00	76.54	18.08	83ms
TCP-traj w/o distillation [23]	1.96	49.30	20.45	**78.78**	22.96	83ms
ThinkTwice* [38]	0.95	62.44	31.23	69.33	16.22	762ms
DriveAdapter* [24]	1.01	**64.22**	**33.08**	70.22	16.01	931ms

TABLE II: **Open-loop and Closed-loop Results in Bench2Drive**. All results are trained on the base training set. Avg. L2 is averaged over the predictions in 2 seconds under 2Hz. * denotes expert feature distillation. Latency is measured on the A6000.

Method	**Ability ↑**					
	M	O	EB	GW	TS	Mean
AD-MLP [14]	0.0	0.0	0.0	0.0	4.4	0.9
UniAD-Tiny [21]	8.9	9.3	20.0	20.0	15.4	14.7
UniAD-Base [21]	14.1	17.8	21.7	10.0	14.2	15.6
VAD [6]	8.1	24.4	18.6	20.0	19.2	18.1
DriveTransformer [37]	17.6	**35.0**	**48.4**	40.0	**52.1**	**38.6**
SNG-VLA	**33.8**	11.1	46.6	**50.0**	50.0	38.1
TCP* [23]	16.12	20.0	20.0	10.0	7.0	14.6
TCP-ctrl* [23]	10.3	4.4	10.0	10.0	6.5	8.2
TCP-traj* [23]	8.9	24.3	**51.7**	40.0	46.3	34.2
ThinkTwice* [38]	27.4	18.4	35.8	**50.0**	54.2	37.2
DriveAdapter* [24]	**28.8**	**26.4**	48.8	**50.0**	**56.4**	**42.1**

TABLE III: **Multi-Ability Results in Bench2Drive**. All results are trained on the base training set. * denotes expert feature distillation.M: Merging, O: Overtaking, EB: Emergency Brake, GW: Give Way, TS: Traffic Sign, M: Mean.

without employing SNG-QA. We use a learning rate of 1e-6 with a cosine annealing schedule and a warmup ratio of 0.03. The model is trained on 8 × NVIDIA A100 GPU 80G with a per-GPU batch size of 8 for 10 epochs.

B. Main results

We compare our method with other E2E-AD methods in both Bench2Drive and NAVSIM. Table I presents the results on NAVSIM [9]. After adding SNG, Transfuser's performance has been significantly enhanced, with improvements observed in both DAC and TTC. The performance of Hydra-MDP [35] is enhanced through additional training to optimize for the EP evaluation metric, utilizing supplementary supervision and weighted confidence post-processing. Despite this, our model achieves SOTA performance without relying on any supervision from perception tasks. Our model exhibits improvement on the DAC (drivable area compliance) metric, which underscores the enhanced efficacy of the proposed SNG. The SNG-VLA-QA model can perform reasoning tasks while maintaining planning performance.

Table II and Table III present the results in Bench2Drive. Models based on driving commands exhibit insufficient modeling of navigation information, leading to target loss during trajectory planning. Consequently, these models generate

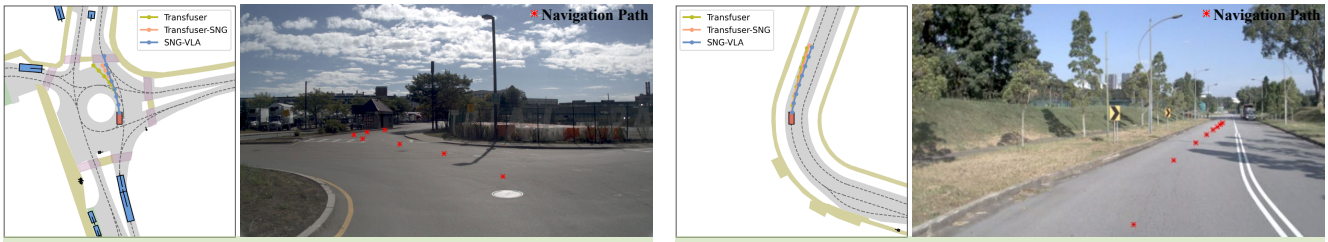


Fig. 4: Qualitative analysis of real-world scenarios. Navigation paths are augmented with substantial noise before being fed into the model to mitigate the influence of privileged information.

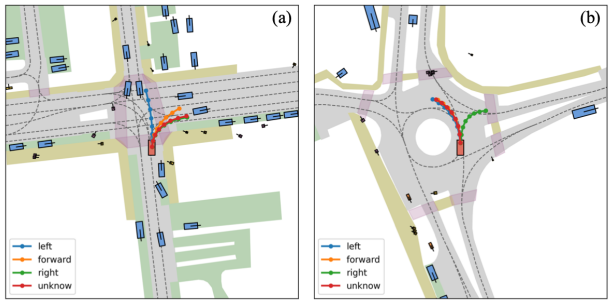


Fig. 5: We demonstrate the impact of introducing noise to the driving command on the predicted trajectory during the inference process of the Transfuser [8]. The experimental results are based on the official checkpoint [9].

trajectories that deviate randomly from the intended path. Furthermore, cumulative errors in closed-loop experiments further degrade the performance of driving command-based methods, leading to poor performance in key metrics such as task completion rate and driving score. In contrast, our SNG-based model has significantly outperformed existing methods. Compared to UniAD-Base [21], our method surpasses it by 46.6% and 119.4% in terms of Driving Score and Success Rate, respectively. In the mean Multi-Ability score, our method outperforms VAD [6] by 110.7%. We employed a smaller backbone with a single vision encoder that processes only the front view image, achieving favorable latency performance without text generation tasks.

C. Qualitative Results

Fig. 4 presents the qualitative experimental results of our model. In both scenarios, transfuser-sng demonstrates superior performance compared to transfuser. The incorporation of SNG enables the model to more effectively comprehend global navigation information and subsequently predict the planning trajectory in accordance with the navigation guidance. Similarly, SNG-VLA exhibits remarkable capabilities in complex scene understanding and navigation following.

D. Ablation study

1) Driving command fails to model navigation information: Undoubtedly, clear navigation information plays an important role in autonomous driving systems, providing essential guidance for determining the vehicle's direction of driving. However, as shown in Table IV, after introducing varying levels of noise into the driving command, the model can yield results that are comparable to, or surpass the official results on metrics on PDM score and others. The use of random or fixed driving commands has minimal impact on model performance. We present qualitative results in Fig. 5. In an open intersection scenario where the expert trajectory demonstrates a right turn, the model fails to execute "Go Forward" and "Turn Left" commands. In the roundabout scenario where the expert trajectory performs a left turn, when given a "Turn Right" command, the model generates a counter-flow right turn trajectory that could cause severe accidents. Notably, in both scenarios, when the driving command is set to "Unknown" the model's output performance is comparable to that achieved with the correct driving commands.

2) Driving command vs. Sequential navigation guidance: As illustrated in Table V (ID 0-4), the results obtained without any navigation information (ID 0) and with driving commands (ID 1) exhibit no significant difference. The performance achieved by solely utilizing TBT information (ID 2) surpasses that of both (ID 0) and (ID 1). Notably, when employing only two points spaced 20 meters apart as the navigation path (ID 3), the model's performance is comparable to that of (ID 2), indicating that the navigation path can provide the model with effective spatial reference under sparse sampling. The model achieves its optimal performance when both the navigation path and TBT information are used as sequential navigation guidance (ID 4). These results demonstrate that sequential navigation guidance models navigation information more effectively than driving commands alone. Specifically, the navigation path provides long-term trajectory constraints, while TBT information offers real-time decision-making logic, such as road traffic conditions. The synergy between these two elements mitigates the limitations associated with relying on a single modality.

3) Optimal combination of navigation path & TBT information: We further investigated the optimal combination of navigation paths and TBT information, as practical operations often face challenges in consistently obtaining

Command	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
Original	97.7	92.8	92.8	100	79.2	84.0
None	97.7	93.5	92.9	100	78.7	84.4
Random	97.8	93.4	93.3	100	79.2	84.7
Left	97.4	93.4	92.5	100	79.2	84.3
Right	97.7	93.5	93.2	100	78.9	84.4
Forward	97.7	93.2	93.1	100	78.9	84.2

TABLE IV: **Command ablation results in NAVSIM**. We conducted experiments on NAVSIM [9] using Transfuser [8] by modifying the input driving commands. Original: ground truth driving command; None: no driving command added; Random: random driving command; Left, Right, Forward: fixed driving commands. All results are obtained under identical training parameters. The introduction of noise into the driving command has minimal impact on the final planning performance.

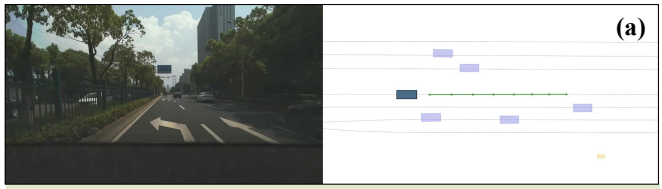
ID	Navigation Path	TBT Information	Driving Command	NC↑	DAC↑	PDMS↑
0	–	–	–	97.2	95.1	85.9
1	–	–	✓	97.5	95.3	86.1
2	–	✓	–	97.6	95.2	86.4
3	2×20	–	–	97.8	95.1	86.4
4	2×20	✓	–	97.5	96.1	87.6
5	4×10	–	–	97.5	**96.6**	87.7
6	4×10	✓	–	**98.9**	96.5	**88.2**
7	8×5	–	–	97.5	96.2	87.2
8	8×5	✓	–	97.5	96.6	87.6

TABLE V: **Ablation of navigation information representation.** We conduct ablation studies on the sampling interval of the navigation path, TBT information and driving commands.

sufficiently dense navigation paths or accurate TBT information. As shown in Table V (ID 3-8), we systematically evaluated the impact of varying densities of navigation path points and the inclusion of TBT information on the results. Comparing (ID 3, 5, 7), it's shown that 4 navigation points spaced 10 meters apart yielded the best performance. Both excessively sparse and overly dense configurations led to diminished performance. Sparse navigation points fail to accurately model the reference path, while overly dense points impose excessive constraints on the model, resulting in poor performance in scenarios such as obstacle avoidance. Across (ID 3-8), the inclusion of TBT information consistently improved performance under varying navigation point densities. In conclusion, a moderate-density navigation path combined with TBT information serves as an effective SNG setting, optimally modeling navigation information and maximizing the model's planning performance.

E. Real world experiments

To further evaluate our proposed SNG and SNG-VLA in real-world scenarios, we established a validation platform. The platform is equipped with a LiDAR, the Innovusion Falcon 300, and a surround-view camera system comprising five AR0820 cameras with a 120-degree horizontal field of view (HFOV) and two AR0820 cameras with a 70-degree HFOV. The onboard computing unit consists of dual Orin



Global planning: Go straight and turn left at the next intersection, keep to the left lane.
Local planning: The road ... Global navigation directs ... You should **maintain moderate speed**, **follow the left-turn lane markings**, and **keep to the left lane** after turning while watching for any pedestrians near the roadside.

Global planning: Turn left, then go straight.
Local planning: The road ... Global navigation directs you to turn left. **Maintain moderate speed**, **follow left-turn lane markings**, and keep left after turning. **Watch for electric bikes and the construction area on the right**. Stay alert for pedestrians near the roadside.

Fig. 6: Qualitative analysis of real-world scenarios.

modules. We collect an in-house dataset and conduct experiments. We present the model's reasoning process and planned trajectories in Fig. 6. Our method performs excellently in selecting turning lanes and providing warnings for special scenarios, pedestrians, and electric vehicles.

V. Conclusions

In this study, we investigate the limitations of end-to-end autonomous driving systems in utilizing navigation information and introduce a novel representation, termed Sequential Navigation Guidance, which integrates long-term trajectory constraints and real-time decision logic. Our model SNG-VLA, achieves superior performance in closed-loop evaluations without the supervision from perception tasks. Experiments conducted in real-world scenarios further confirm the robustness and practical applicability of our approach.

References

[1] T. Wu, A. Luo, R. Huang, H. Cheng, and Y. Zhao, "End-to-end driving model for steering control of autonomous vehicles with future spatiotemporal features," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 950–955.

[2] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 537–547, 2022.

[3] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 164–10 183, 2024.

[4] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 163–168.

[5] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2090–2096.

[6] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8340–8350.

[7] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, "St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," in *European Conference on Computer Vision*. Springer, 2022, pp. 533–549.

[8] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 12 878–12 895, 2023.

[9] D. Dauner, M. Hallgarten, T. Li, X. Weng, Z. Huang, Z. Yang, H. Li, I. Gilitschenski, B. Ivanovic, M. Pavone, A. Geiger, and K. Chitta, "Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[10] X. Weng, B. Ivanovic, Y. Wang, Y. Wang, and M. Pavone, "Paradrive: Parallelized architecture for real-time autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 15 449–15 458.

[11] C. Sima, K. Renz, K. Chitta, L. Chen, H. Zhang, C. Xie, J. Beißwenger, P. Luo, A. Geiger, and H. Li, "Drivelm: Driving with graph visual question answering," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024, pp. 256–274.

[12] W. Zheng, R. Song, X. Guo, C. Zhang, and L. Chen, "Genad: Generative end-to-end autonomous driving," in *Proceedings of the European Conference on Computer Vision (ECCV)*, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds., 2024, pp. 87–104.

[13] Q. Peng, C. Bai, G. Zhang, B. Xu, X. Liu, X. Zheng, C. Chen, and C. Lu, "Navigscene: Bridging local perception and global navigation for beyond-visual-range autonomous driving," *arXiv preprint arXiv:2507.05227*, 2025.

[14] J.-T. Zhai, Z. Feng, J. Du, Y. Mao, J.-J. Liu, Z. Tan, Y. Zhang, X. Ye, and J. Wang, "Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes," *arXiv preprint arXiv:2305.10430*, 2023.

[15] Z. Li, Z. Yu, S. Lan, J. Li, J. Kautz, T. Lu, and J. M. Alvarez, "Is ego status all you need for open-loop end-to-end autonomous driving?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 864–14 873.

[16] G. Svennerberg, *Beginning google maps API 3*. Apress, 2010.

[17] X. Jia, Z. Yang, Q. Li, Z. Zhang, and J. Yan, "Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving," in *NeurIPS 2024 Datasets and Benchmarks Track*, 2024.

[18] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[19] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[20] M. Naumann and C. Stiller, "Aib-mdp: Continuous probabilistic motion planning for automated vehicles by leveraging action independent belief spaces," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 6373–6380.

[21] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 853–17 862.

[22] S. Casas, A. Sadat, and R. Urtasun, "Mp3: A unified model to map, perceive, predict and plan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 403–14 412.

[23] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao, "Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6119–6132, 2022.

[24] X. Jia, Y. Gao, L. Chen, J. Yan, P. L. Liu, and H. Li, "Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 7953–7963.

[25] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[26] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[27] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y. K. Wong, Z. Li, and H. Zhao, "Drivegpt4: Interpretable end-to-end autonomous driving via large language model," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, pp. 8186–8193, 2024.

[28] H. Shao, Y. Hu, L. Wang, G. Song, S. L. Waslander, Y. Liu, and H. Li, "Lmdrive: Closed-loop end-to-end driving with large language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 15 120–15 130.

[29] K. T. e. a. H. Caesar, J. Kabzan, "Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," in *CVPR ADP3 workshop*, 2021.

[30] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[31] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei *et al.*, "Qwen2. 5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.

[32] Z. Zhou, T. Cai, S. Z. Zhao, Y. Zhang, Z. Huang, B. Zhou, and J. Ma, "Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning," *arXiv preprint arXiv:2506.13757*, 2025.

[33] K. Renz, L. Chen, E. Arani, and O. Sinavski, "Simlingo: Vision-only closed-loop autonomous driving with language-action alignment," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 11 993–12 003.

[34] C. Yuan, Z. Zhang, J. Sun, S. Sun, Z. Huang, C. D. W. Lee, D. Li, Y. Han, A. Wong, K. P. Tee *et al.*, "Drama: An efficient end-to-end motion planner for autonomous driving with mamba," *arXiv preprint arXiv:2408.03601*, 2024.

[35] Z. Li, K. Li, S. Wang, S. Lan, Z. Yu, Y. Ji, Z. Li, Z. Zhu, J. Kautz, Z. Wu *et al.*, "Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation," *arXiv preprint arXiv:2406.06978*, 2024.

[36] B. Liao, S. Chen, H. Yin, B. Jiang, C. Wang, S. Yan, X. Zhang, X. Li, Y. Zhang, Q. Zhang, and X. Wang, "Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving," *arXiv preprint arXiv:2411.15139*, 2024.

[37] X. Jia, J. You, Z. Zhang, and J. Yan, "Drivetransformer: Unified transformer for scalable end-to-end autonomous driving," *arXiv preprint arXiv:2503.07656*, 2025.

[38] X. Jia, P. Wu, L. Chen, J. Xie, C. He, J. Yan, and H. Li, "Think twice before driving: Towards scalable decoders for end-to-end autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 21 983–21 994.

[39] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu *et al.*, "Llava-onevision: Easy visual task transfer," *arXiv preprint arXiv:2408.03326*, 2024.

[40] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 11 975–11 986.

[41] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[42] J. Cheng, Y. Chen, X. Mei, B. Yang, B. Li, and M. Liu, "Rethinking imitation-based planners for autonomous driving," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 123–14 130.