# VADv2: End-to-End Autonomous Driving via Probabilistic Planning

**Anonymous authors**Paper under double-blind review

000

001

002003004

010 011

012

013

014

016

017

018

019

021

023

025

026027028

029

031

033

034

037

040

041

042

043

044

046

047

048

050

051

052

## **ABSTRACT**

Learning a human-like driving policy from large-scale driving demonstrations is promising, but the uncertainty and non-deterministic nature of planning make it challenging. Existing learning-based planning methods follow a deterministic paradigm to directly regress the action, failing to cope with the uncertainty problem. In this work, we propose a probabilistic planning model for end-to-end autonomous driving, termed VADv2. We resort to a probabilistic field function to model the mapping from the action space to the probabilistic distribution. Since the planning action space is a high-dimensional continuous spatiotemporal space and hard to tackle, we first discretize the planning action space to a large planning vocabulary and then tokenize the planning vocabulary into planning tokens. Planning tokens interact with scene tokens and output the probabilistic distribution of action. Mass driving demonstrations are leveraged to supervise the distribution. VADv2 achieves state-of-the-art closed-loop performance on the CARLA Town05 benchmark, significantly outperforming all existing methods. We also provide comprehensive evaluations on the NAVSIM dataset and a large-scale 3DGS-based benchmark, demonstrating its effectiveness in real-world applications. Code will be released to facilitate future research.

# 1 Introduction

End-to-end autonomous driving is an important research topic currently. A plethora of human driving demonstrations in real-world scenarios are readily accessible. It is very promising to derive a human-like driving policy for vehicle control from these extensive demonstrations. However, the uncertainty and non-deterministic nature of planning make it challenging to extract the driving knowledge from driving demonstrations. To illustrate such uncertainty, two scenarios are presented in Fig. 1 and explained as follows. 1) Following another vehicle: The human driver exhibits various reasonable driving maneuvers, such as maintaining the current lane or changing lanes to overtake. 2) Interaction with an oncoming vehicle: The human driver faces two potential driving maneuvers, i.e., yielding or overtaking. From a statistical perspective, the actions (including timing and speed) are highly stochastic, influenced by numerous latent factors that cannot be accurately modeled.

Existing learning-based planning methods (Jiang et al., 2023; Hu et al., 2022c; Jia et al., 2023b; Prakash et al., 2021b; Hu et al., 2022a; Zhang et al., 2021) follow a deterministic paradigm to directly regress the action. The regression target is the future trajectory in (Jiang et al., 2023; Hu et al., 2022c; Jia et al., 2023b; Prakash et al., 2021b) and control signal (acceleration and steering) in (Hu et al., 2022a; Zhang et al., 2021). Such a paradigm assumes there exists a deterministic relation between the driving scene and action, which is not the case. The variance of human driving behavior causes the ambiguity of the regression target. Especially when the feasible solution space is non-convex, *i.e.*, there exist multiple feasible solutions (see Fig. 1), the deterministic modeling cannot cope with non-convex cases and may output an in-between action, causing safety problems.

In this work, we propose probabilistic planning to cope with the uncertainty of planning. We model the planning policy as a scene-conditioned non-stationary stochastic process, formulated as  $p(\boldsymbol{a}|\boldsymbol{o})$ , where  $\boldsymbol{o}$  is the historical and current observations of the driving scene, and  $\boldsymbol{a}$  is a candidate planning action. Compared with deterministic planning, probabilistic planning is more robust against uncertainty in planning and able to model non-convex feasible solution space, and thus achieves more accurate and safer planning performance.

We resort to a probabilistic field function to model the mapping from the action space to the probabilistic distribution. Since the planning action space is a high-dimensional continuous spatiotemporal space and hard to tackle, we first discretize the planning action space to a large planning vocabulary and then tokenize the planning vocabulary into planning tokens. Planning tokens interact with scene tokens and output the probabilistic distribution of action. Mass driving demonstrations are leveraged to supervise the distribution.

Probabilistic planning has two other advantages. First, unlike deterministic planning which has to regress the optimal action according to scene information, probabilistic planning models the correlation between each action and the driving scene. It just ranks different actions and samples a high-scoring one. Such modeling is much simpler. Besides, probabilistic planning is flexible in the inference stage. It

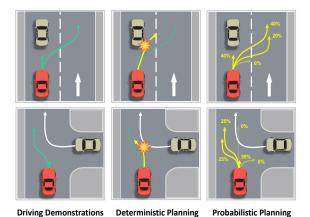


Figure 1: Uncertainty exists in planning. There doesn't exist a deterministic relation between driving scene and action. The deterministic planning fails to model such uncertainty especially when the feasible solution space is non-convex. VADv2 is based on probabilistic planning and learns the scene-conditioned probabilistic distribution of action from large-scale driving demonstrations.

outputs multi-mode planning results and is easy to combine with rule-based and optimization-based planning methods. We can flexibly add other candidate planning actions to the planning vocabulary and evaluate them because we model the distribution over the whole action space.

Based on the probabilistic planning, we present VADv2, an end-to-end driving model, which takes surround-view image sequence as input in a streaming manner, tokenizes sensor data and planning action space, outputs the probabilistic distribution of action, and samples one action to control the vehicle. Only with camera sensors, VADv2 achieves state-of-the-art closed-loop planning performance on the CARLA Town05 benchmark, significantly outperforming all existing methods, and also delivers leading planning results on the NAVSIM and our 3DGS-based benchmark. VADv2 runs stably in a fully end-to-end manner, even without a rule-based wrapper as a post-processing step to avoid infraction.

Our contributions are summarized as follows:

- We propose probabilistic planning to cope with the uncertainty and non-deterministic nature of planning. We design a probabilistic field to map from the action space to the probabilistic distribution and learn the distribution of action from large-scale driving demonstrations.
- Based on the probabilistic planning, we present VADv2, an end-to-end driving model, which tokenizes sensor data and planning action space for interaction, outputs the probabilistic distribution of action, and samples one action to control the vehicle.
- VADv2 achieves state-of-the-art planning performance in both closed- and open-loop settings
  across multiple benchmarks. Abundant closed-loop simulations and real-world deployment results
  validate its effectiveness and stability in vehicle control.

# 2 RELATED WORK

**Perception.** Perception is the first step in achieving autonomous driving, and a unified representation of driving scenes is beneficial for easy integration into downstream tasks. Bird's Eye View (BEV) representation has become a common strategy in recent years, enabling effective scene feature encoding and multimodal data fusion. LSS (Philion & Fidler, 2020) is a pioneering work that achieves the perspective view to BEV transformation by explicitly predicting depth for image pixels. BEVFormer (Li et al., 2022c), on the other hand, avoids explicit depth prediction by designing spatial and temporal attention mechanisms. Subsequent works (Li et al., 2022b; Wang et al., 2023a) continuously optimize temporal modeling and BEV transformation strategies. In terms of vectorized

mapping, HDMapNet (Li et al., 2022a) converts lane segmentation into vector maps through post-processing. VectorMapNet (Liu et al., 2022) predicts vector map elements in an autoregressive manner. MapTR (Liao et al., 2022; 2023b) introduces permutation equivalence and hierarchical matching strategies, significantly improving mapping performance. LaneGAP (Liao et al., 2023a) introduces path-wise modeling for lane graphs.

Motion Prediction. Motion prediction aims to forecast future trajectories of other traffic participants, assisting the ego vehicle in making informed planning decisions. Traditional motion prediction methods utilizes input such as historical trajectories and high-definition maps to predict future trajectories (Chai et al., 2019; Gao et al., 2020; Liu et al., 2021). However, recent end-to-end methods (Hu et al., 2021; Zhang et al., 2022; Gu et al., 2022; Jiang et al., 2022) perform perception and motion prediction jointly. Some works (Hu et al., 2021; Zhang et al., 2022) predict future motion as dense occupancy and flow. Some other works (Gu et al., 2022; Jiang et al., 2022) predict agent-level multi-mode trajectories. Planning can draw inspiration from these approaches to model uncertainty, although the number of modes remains limited.

Planning. Learning-based planning has shown great potential recently due to its data-driven nature and impressive performance with increasing amounts of data. Early attempts (Pomerleau, 1988; Codevilla et al., 2019b; Prakash et al., 2021a) use a completely black-box spirit, where sensor data is directly used to predict control signals. However, this strategy lacks interpretability and is difficult to optimize. In addition, there are numerous studies combining reinforcement learning and planning (Chen et al., 2021; Zhang et al., 2021; Gao et al., 2025) by autonomously exploring driving behavior in closed-loop simulation environments. Imitation learning (Chekroun et al., 2021; Hu et al., 2022b; Ma et al., 2025; Liao et al., 2025) is another research direction, where models learn expert driving behavior to achieve good planning performance and develop a driving style close to that of humans. In recent years, end-to-end autonomous driving has emerged, integrating perception, motion prediction, and planning into a single model, resulting in a fully data-driven approach that demonstrates promising performance. UniAD (Hu et al., 2022c) cleverly integrates multiple perception and prediction tasks to enhance planning performance. VAD (Jiang et al., 2023) explores the potential of vectorized scene representation for planning and getting rid of dense maps.

Large Language Model in Autonomous Driving. Recent researches explore the combination of LLMs and autonomous driving (Sha et al., 2023; Xu et al., 2023; Mao et al., 2023). One line of work utilizes LLMs for driving scene understanding and evaluation through question-answering (Chen et al., 2023; Sima et al., 2024). Another approach goes a step further by directly utilizing LLMs for planning (Wang et al., 2023b; 2024). However, current LLM-based planning approaches inevitably suffer from limited inference speed, which constrains their practicality for real-time deployment in autonomous driving applications. VADv2 draws inspiration from GPT (Achiam et al., 2023) to cope with the uncertainty problem, which also exists in language modeling. Given a specific context, the next word is non-deterministic, LLM learns the context-conditioned probabilistic distribution of the next word from a large-scale corpus, and samples one word from the distribution. Inspired by LLM, VADv2 models the planning policy as a scene-conditioned nonstationary stochastic process. VADv2 discretizes the action space to generate a planning vocabulary, approximates the probabilistic distribution based on large-scale driving demonstrations, and samples one action from the distribution at each time step to control the vehicle.

# 3 VADv2

The overall framework of VADv2 is depicted in Fig. 2. VADv2 takes multi-view image sequences as input in a streaming manner, transforms sensor data into scene token embeddings, outputs the probabilistic distribution of action, and samples one action to control the vehicle. Large-scale driving demonstrations and scene constraints are used to supervise the predicted distribution.

#### 3.1 Scene Encoder

VADv2 uses a scene encoder to transform the sensor data into instance-level token embeddings  $E_{\text{scene}} \in \mathbb{R}^{M \times D}$  to explicitly extract high-level information, where M is the number of scene tokens

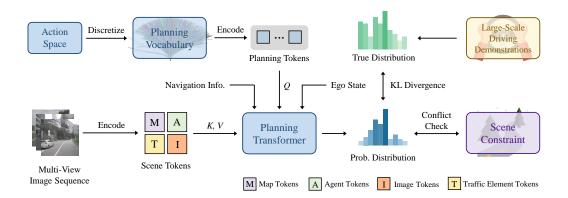


Figure 2: **Overall architecture of VADv2.** VADv2 takes multi-view image sequences as input in a streaming manner, tokenizes sensor data and planning action space, outputs the probabilistic distribution of action, and samples one action to control the vehicle. Large-scale driving demonstrations and scene constraints are used to supervise the predicted distribution.

and D is the feature dimension. Concretely,  $E_{\rm scene}$  includes four kinds of scene tokens: map tokens, agent tokens, traffic element tokens, and image tokens.

**BEV Encoder.** VADv2 first employs a BEV encoder (Li et al., 2022c) to transform the multi-view image features from perspective view to Bird's Eye View, obtaining a feature map in the BEV space. This feature map will be used for learning instance-level map features and agent features.

**Map Tokens.** VADv2 utilizes a group of map tokens (Liao et al., 2023a;b) to learn the vectorized map elements of the driving scene from the BEV feature map, including lane centerline, lane divider, road boundary, and pedestrian crossing.

**Agent Tokens.** Besides, a group of agent tokens (Jiang et al., 2023) is adopted to predict traffic participants' motion information, including location, orientation, size, speed, and future trajectories.

**Traffic Element Tokens.** Traffic signals also play a vital role in planning. In CARLA, we consider two types of traffic signals: traffic light signals and stop signs. VADv2 further encodes the front view image features extracted from the image backbone with an MLP into traffic element tokens to predict the states of the traffic signals.

**Image Tokens.** Apart from the above instance-level tokens, VADv2 also utilizes the front-view image features as the image tokens. These image tokens provide a more dense scene feature for planning, containing rich information that is complementary to the instance-level tokens.

Map tokens, agent tokens, and traffic element tokens are supervised with corresponding supervision signals to ensure they explicitly encode corresponding high-level information. Besides, navigation information and ego state are also encoded into embeddings ( $E_{\rm navi}, E_{\rm state}$ ) with an MLP. In summary, the Scene Encoder transforms the sparse sensor data into more compact high-level scene features ( $E_{\rm scene}, E_{\rm navi}, E_{\rm state}$ ), which serve as the foundation for the following planning module.

#### 3.2 PROBABILISTIC PLANNING

We propose probabilistic planning to cope with the uncertainty of planning. We model the planning policy as a scene-conditioned nonstationary stochastic process, formulated as p(a|o), where o is the observed scene information and a is the action, represented as a waypoint sequence of future planning trajectory,

$$o = (E_{\text{scene}}, E_{\text{navi}}, E_{\text{state}}), \ o = (x_1, y_1, x_2, y_2, ..., x_T, y_T).$$
 (1)

T is the number of waypoints. Each waypoint  $(x_i, y_i)$  corresponds to a future timestamp  $t_i$ .

We approximate the probabilistic distribution of the planning action space based on large-scale driving demonstrations, and sample one action from the distribution at each time step to control the vehicle.

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244245

246

247

248249

250

251252253

254

255

256257

258

259

260

261262

263

264265

266

267 268

269

The planning action space is a high-dimensional continuous spatiotemporal space and hard to tackle. Thus, we discretize the planning action space to a large planning vocabulary V = $\{a^i\}^N$ , where N is the vocabulary size. Specifically, we collect all the planning actions in driving demonstrations as the planning action set  $\mathcal{S}$  and adopt the furthest trajectory sampling to select N representative actions to serve as the planning vocabulary. The vocabulary sampling algorithm is presented in Alg. 1. Each planning action in V is sampled from driving demonstrations and thus naturally satisfies the kinematic constraints of the ego vehicle, which means that when the action is converted into control signals (steer, throttle, and brake), the control signal values do not exceed the feasible range. By default, N is set to 4096.

The probability p(a) is assumed to be continuous with respect to a and insensitive to the little deviation of a, *i.e.*,  $\lim_{\Delta a \to 0} (p(a) - p(a + \Delta a)) = 0$ . Inspired by NeRF (Mildenhall et al., 2020), which models the continuous radiance field over the 5D

```
Algorithm 1: Planning vocabulary sampling.
```

```
Input: Planning action set S, Planning vocabulary
         size N
Output: Planning vocabulary V
Initialization: V \leftarrow \emptyset
for i = 1 to N - 1 do
     if \mathcal{V} == \emptyset then
          random select a action a in S
           \mathcal{V} \leftarrow \mathcal{V} \cup \{a\} add action to planning
            vocabulary
           \mathcal{S} \leftarrow \mathcal{S} \setminus \{a\} remove action from
            planning action set
     end
     max\_dis = 0
     for trajectory a in S do
           dis = calculate\_distance(a, V)
          if dis > max\_dis then
                max\_dis = dis
                \hat{a} = a update the currently furthest
                  trajectory
          end
     end
     \mathcal{V} \leftarrow \mathcal{V} \cup \{\hat{m{a}}\}
Return: V
```

**calculate\_distance** will calculate the distance between the endpoint of action a and the endpoints of all actions in set  $\mathcal{V}$ . It will return the minimum value among these distances as the result.

space, we resort to a probabilistic field to model the continuous mapping from the action space to the probabilistic distribution.

Concretely, we first encode each action (trajectory waypoint) into a high-dimensional planning token embedding E(a),

$$E(\mathbf{a}) = (\Gamma(x_i), \Gamma(y_i))_{i=1}^T, \Gamma(\text{pos}) = (\gamma(\text{pos}, j))_{j=0}^{L-1},$$
  

$$\gamma(\text{pos}, j) = (\cos(\text{pos}/10000^{2\pi j/L}), \sin(\text{pos}/10000^{2\pi j/L})).$$
(2)

 $\Gamma$  is an encoding function that maps each coordinate from  $\mathbb{R}$  into a high dimensional embedding space  $\mathbb{R}^{2L}$ , and is applied separately to each of the coordinate values of trajectory a. pos denotes the position (referring to the x or y coordinate of waypoint). We use these functions to map continuous coordinates into a higher dimensional space to better approximate a higher frequency field function.

Then, we use a cascaded Transformer decoder  $\phi$  for interaction with scene information  $E_{\rm scene}$ , and combine with navigation information  $E_{\rm navi}$  and ego state  $E_{\rm state}$  to output the probability of each action,

$$p(\mathbf{a}) = \sigma(\text{MLP}(\phi(E(\mathbf{a}), E_{\text{scene}}) + E_{\text{navi}} + E_{\text{state}})). \tag{3}$$

 $\sigma$  is the sigmoid function. In the Transformer decoder  $\phi$ , E(a) serves as query, and  $E_{\text{scene}}$  serves as key and value. E(a),  $E_{\text{navi}}$ ,  $E_{\text{state}}$ , and the output of MLP are with the same dimension  $(1 \times D)$ .

#### 3.3 Training

We train VADv2 with three kinds of supervision, distribution loss, conflict loss, and scene token loss.

**Distribution Loss.** We learn the probabilistic distribution from large-scale driving demonstrations. KL divergence is used to minimize the difference between the predicted distribution and the distri-

bution of the data.  $p_{\text{data}}(a)$  is estimated through occurrence frequency in demonstrations. Since  $p_{\text{data}}(a)$  is fixed,  $p_{\text{data}}(a) \cdot \log p_{\text{data}}(a)$  is a constant and can be omitted.

$$\mathcal{L}_{\text{distribution}} = D_{\text{KL}}(p_{\text{data}}||p_{\text{pred}}) = -\sum_{\boldsymbol{a} \in \mathcal{V}} p_{\text{data}}(\boldsymbol{a}) \cdot \log p_{\text{pred}}(\boldsymbol{a}). \tag{4}$$

For each frame in the demonstrations, we select the action from the planning vocabulary that has the lowest L2 distance to the ground-truth action. This best-matched action is assigned a label of 1, and all other actions are assigned 0. Over all the frames, the occurrence frequency  $p_{\rm data}(a)$  of one action a is then estimated by counting how often each action is the best match, normalized by the total number of frames. This modeling is analogous to the standard formulation used in Large Language Models, where the ground-truth token is labeled as 1 and others as 0, and cross-entropy loss is used to minimize the divergence between the predicted distribution and the empirical distribution.

**Conflict Loss.** We use the driving scene constraints to help model learn important prior knowledge about driving and regularize the predicted distribution. Specifically, if one action in the planning vocabulary conflicts with other agents' ground truth future motion or road boundary, the action is regarded as a negative sample, and we impose a significant loss to reduce the probability of this action,

$$\mathcal{L}_{\text{conflict}} = \sum_{\boldsymbol{a} \in \mathcal{V}} \mathbb{1}_{\text{conflit}}(\boldsymbol{a}) \cdot \log p_{\text{pred}}(\boldsymbol{a}).$$
 (5)

 $\mathbb{1}_{\text{conflit}}(a)$  is the indicator function, whose value is 1 if conflict happens to a, otherwise is 0.

**Scene Token Loss.** Map, agent, and traffic-element tokens are supervised with corresponding supervision signals to ensure they explicitly encode the relevant high-level information.

The loss of map tokens is the same with MapTRv2 (Liao et al., 2023b).  $l_1$  loss is adopted to calculate the regression loss between the predicted map points and the ground truth map points. Focal loss is used as the map classification loss.

The loss of agent tokens is composed of the detection loss and the motion prediction loss (Jiang et al., 2023).  $l_1$  loss is used as the regression loss to predict agent attributes (location, orientation, size, etc.), and focal loss to predict agent classes. For each agent who has matched with a ground truth agent, we predict K future trajectories and use the trajectory that has the minimum final displacement error (minFDE) as a representative prediction. Then we calculate  $l_1$  loss between this representative trajectory and the ground truth trajectory as the motion regression loss. Besides, focal loss is adopted as the multi-modal motion classification loss.

Traffic element tokens consist of two parts: the traffic light token and the stop sign token. On one hand, we send the traffic light token to an MLP to predict the state of the traffic light (yellow, red, and green) and whether the traffic light affects the ego vehicle. On the other hand, the stop sign token is also sent to an MLP to predict the overlap between the stop sign area and the ego vehicle. Focal loss is used to supervise these predictions. The final loss can be denoted as:

$$\mathcal{L} = \mathcal{L}_{\text{distribution}} + \mathcal{L}_{\text{conflict}} + \mathcal{L}_{\text{token}}.$$
 (6)

# 3.4 Inference

In closed-loop inference, it's flexible to get the driving policy from the distribution. Intuitively, we sample the action with the highest probability at each time step, and use the PID controller to convert the selected action to control signals (steer, throttle, and brake).

In real-world applications, there are more robust strategies to make full use of the probabilistic distribution. A good practice is, sampling top-K actions as proposals, and adopting a rule-based wrapper for filtering proposals and an optimization-based post-solver for fine-grained trajectory refinement. Besides, the probability of the action reflects how confident the end-to-end model is, and can be regarded as the judgment condition to switch between conventional rule-based planning and control and learning-based planning and control.

324 325

Table 1: Closed-loop evaluation on the Town05 Long benchmark.

> 344 345

352 353 354

355

351

361

362

363

364

372

373

374

375376

377

Method Modality RC ↑ IS ↑ Reference DS ↑ Transfuser (Prakash et al., 2021b) C+L TPAMI 22 31.0 47.5 0.77 ThinkTwice (Jia et al., 2023b) C+L CVPR 23 95.5 70.9 0.75 C+L ICCV 23 71.9 97.3 0.74 DriveAdapter+TCP (Jia et al., 2023a) DriveMLM (Wang et al., 2023b) C+L arXiv 23 76.1 98.1 0.78 CILRS (Codevilla et al., 2019a)  $\mathbf{C}$ CVPR 19 7.8 10.3 0.75 C LBC (Chen et al., 2020) CoRL 20 12.3 31.9 0.66 C Roach (Zhang et al., 2021) ICCV 21 41.6 96.4 0.43 C ST-P3 (Hu et al., 2022b) ECCV 22 11.5 83.2 C NeurIPS 22 MILE (Hu et al., 2022a) 61.1 97.4 0.63 C Interfuser (Shao et al., 2023) CoRL 22 68.3 95.0 C VAD (Jiang et al., 2023) ICCV 23 30.3 75.2 C Rao et al. (2024) **TIV 24** 74.9 94.6 0.77 C 73.6 DriveCoT (Wang et al., 2024) arXiv 24 96.8 0.76 C LeapVAD (Ma et al., 2025) arXiv 25 73.7 95.7 0.78 85.1 VADv2 Ours 98.4 0.87

Table 2: End-to-end planning results on the NAVSIM navtest split with closed-loop metrics.

Method	Reference	Modality	NC ↑	DAC ↑	TTC ↑	Comf ↑	EP↑	PDMS ↑
Transfuser (Prakash et al., 2021b)	PAMI 23	C+L	97.7	92.8	92.8	100	79.2	84.0
DRAMA (Yuan et al., 2024)	arXiv 23	C+L	98.0	93.1	94.8	100	80.1	85.5
Hydra-MDP (Li et al., 2024)	arXiv 24	C+L	98.3	96.0	94.6	100	78.7	86.5
DiffusionDrive (Liao et al., 2025)	CVPR 25	C+L	98.2	96.2	94.7	100	82.2	88.1
UniAD (Hu et al., 2022c)	CVPR 23	С	97.8	91.9	92.9	100	78.8	83.4
LTF (Prakash et al., 2021b)	PAMI 23	C	97.4	92.8	92.4	100	79.0	83.8
PARA-Drive (Weng et al., 2024)	CVPR 24	C	97.9	92.4	93.0	99.8	79.3	84.0
VADv2	Ours	C	98.1	94.8	94.3	100	80.6	86.2

#### 4 EXPERIMENTS

# 4.1 EXPERIMENTAL SETTINGS

**CARLA Benchmark.** We first adopt the widely used CARLA (Dosovitskiy et al., 2017) simulator to evaluate the performance of VADv2. Following common practice, we use Town05 benchmarks for closed-loop evaluation. Specifically, each benchmark contains several pre-defined driving routes. Town05 Long consists of 10 routes, each route is about 1km in length. Town05 Short consists of 32 routes, each route is 70m in length. Town05 Long validates the comprehensive capabilities of the model, while Town05 Short focuses on evaluating the model's performance in specific scenarios, such as lane changing before intersections. The simulation and control frequency for closed-loop inference is 10 Hz. VADv2 takes a multi-view image sequence as input in a streaming manner and plans a 3-second future trajectory. The trajectory consists of 6 waypoints (*i.e.*, T=6). The time interval between two adjacent waypoints is 0.5s. The default feature dimension D for VADv2 is set to 256. All experiments are conducted based on 16 NVIDIA 4090 GPUs.

CARLA data. As for generating the driving demonstration data, we use the official autonomous agent of CARLA to collect training data by randomly generating driving routes in Town03, Town04, Town06, Town07, and Town10. We collect approximately 3 million clips for training. For each clip, we save 6-camera surround-view image sequences at 10Hz for the past 1.6 seconds, along with information on traffic signals, traffic participants, and the state of the ego vehicle. Additionally, we obtain the vectorized maps for training the online mapping module by preprocessing the Open-StreetMap (Haklay & Weber, 2008) format maps provided by CARLA. The maps are provided only as ground truth during training, and VADv2 does not use high-definition maps for evaluation.

**NAVSIM and 3DGS-based Benchmarks.** To further validate generalization ability in real-world scenarios, we also evaluate VADv2 on the NAVSIM (Dauner et al., 2024) and a large-scale 3DGS-

Table 3: Closed-loop quantitative comparisons with other methods on the 3DGS-based benchmark.

Method	Reference	CR↓	DCR↓	SCR↓	DR↓	PDR↓	HDR↓	ADD↓
TransFuser VAD	TPAMI 22 ICCV 23	$\frac{0.320}{0.335}$	$\frac{0.273}{0.273}$	0.047 0.062	<b>0.235</b> 0.314	0.188 0.255	<b>0.047</b> 0.059	<b>0.263</b> 0.304
GenAD	ECCV 24	0.341	0.299	0.042	0.291	0.160	0.131	0.265
VADv2	Ours	0.270	0.240	0.030	0.243	0.139	0.104	0.273

Table 4: **Ablation for design choices.** "Dist. Loss" denotes Distribution Loss. "Traf. Token" denotes Traffic Element Token.

ID	Dist.	Conflict	Agent	Map	Traf.	Image		L2 (m) ↓		Co	llision (%	<i>6</i> )↓
ID	Loss	Loss	Token	Token	Token	Token	1s	2s	3s	1s	2s	3s
1		✓	✓	✓	1	✓	1.415	2.310	3.153	0.698	0.755	0.746
2	✓		✓	✓	✓	✓	0.086	0.173	0.291	0.000	0.012	0.039
3	✓	✓		✓	✓	✓	0.089	0.190	0.327	0.015	0.047	0.085
4	/	/	✓		✓	✓	0.086	0.191	0.332	0.005	0.034	0.070
5	/	✓	/	✓		✓	0.082	0.171	0.295	0.000	0.017	0.051
6	/	✓	/	✓	✓		0.083	0.170	0.293	0.000	0.010	0.039
7	✓	✓	✓	✓	✓	✓	0.082	0.169	0.290	0.000	0.010	0.039

based (Kerbl et al., 2023) benchmarks. we collect 2000 hours of real-world human driving demonstrations for training, and utilizing 337 reconstructed 3DGS (3D Gaussian Splatting) environments for closed-loop evaluation. Each environment features an 8s scenario capturing interactions in dense traffic with potential collision risks, providing a representative segment of real-world driving behaviors and multi-agent interactions.

The photorealistic 3DGS reconstruction enables accurate agent trajectory modeling and dynamic environment rendering, providing a testbed that closely mirrors real-world driving conditions. More details of the 3DGS-based benchmark can be found in Appendix A due to page limits. We also deploy VADv2 on real-world vehicles. The results are presented in the supplementary material.

**LLM Usage.** We only use LLMs to check grammar and polish writing in this paper, and the authors take full responsibility for all content.

#### 4.2 METRICS

On the CARLA benchmark, we employ its official closed-loop metrics, Route Completion (RC), Infraction Score (IS) and Driving Score (DS). DS is the product between RC and IS, which serves as the main metric. In benchmark evaluation, most works adopt a rule-based wrapper to reduce the infraction. For a fair comparison, we follow the common practice of adopting a rule-based wrapper over the learning-based policy, which is similar to Transfuser (Prakash et al., 2021b). We also conduct open-loop evaluation using the L2 displacement error and collision rate. In most ablations, we adopt open-loop metrics by default because they are faster to evaluate and more stable. We use the official autonomous agent of CARLA to generate the validation set on the Town05 Long benchmark for open-loop evaluation, and the results are averaged over all validation samples.

On the NAVSIM benchmark, its official closed-loop metrics such as PDMS are adopted. On the 3DGS-based benchmark, we use safety-critical metrics grounded in real-world driving analytics: Collision Ratio (CR, sum of Dynamic and Static Collision Ratios) evaluates interaction safety in dense traffic, while Deviation Ratio (DR, combining Positional and Heading Deviation Ratios) and Average Deviation Distance (ADD) jointly assess trajectory consistency with expert human demonstrations.

#### 4.3 Comparisons with State-of-the-Art Methods

On the Town05 Long benchmark in Tab. 1, VADv2 achieves a DS of 85.1, a RC of 98.4, and an IS of 0.87. Relative to the former state-of-the-art method DriveMLM (Wang et al., 2023b), VADv2 achieves a higher RC while significantly improving DS by 9.0. It is worth noting that VADv2 only

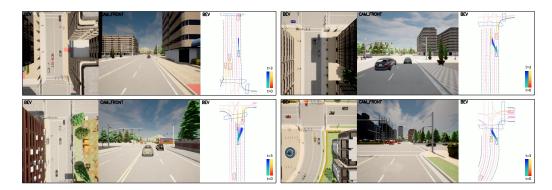


Figure 4: Qualitative results of VADv2 on the CARLA Town05 Long benchmark.

utilizes cameras as perception input, whereas DriveMLM uses both cameras and LiDAR. Furthermore, compared to the previous best camera-based method (Rao et al., 2024), VADv2 demonstrates even greater advantages, with a remarkable increase in DS of up to 10.2.

On the NAVSIM benchmark in Tab. 2, VADv2 also achieves state-of-the-art camera-based planning performance. In comparison with the strongest prior approach (Weng et al., 2024), PDMS greatly improved from 84.0 to 86.2. Additionally, Tab. 3 presents the results on the 3DGS-based benchmark. VADv2 reduces the Collision Ratio to 0.270, a 15.6% improvement over TransFuser (0.320), while maintaining a competitive Deviation Ratio of 0.243 that approaches the best reported performance. These results demonstrate robust safety in real-world dynamic interactions.

# 4.4 ABLATION STUDY

**Key Modules.** Tab. 4 shows the ablation experiments of the key modules in VADv2. 50k clips of driving demonstrations are used in training. The model performs poorly in terms of planning accuracy without the supervision of expert driving behavior provided by the Distribution Loss (ID 1). The Conflict Loss provides critical prior information about driving; therefore, removing it (ID 2) also affects the model's planning accuracy. Scene tokens encode important scene elements into high-dimensional features, and the planning tokens interact with the scene tokens to learn both dynamic and static information about the driving scene. When any type of scene token is missing, the model's planning performance will be affected (ID 3-ID 6). The best planning performance is achieved when the model incorporates all of the aforementioned designs (ID 7). More experiments and ablations are presented in Appendix A due to page limits.

### 4.5 VISUALIZATION

Fig. 4 shows qualitative results of VADv2. The top left image illustrates multi-modal planning trajectories at different driving speeds. The top right shows predictions of both forward creeping and left lane-changing. The bottom left presents a right lane-change at an intersection, where VADv2 predicts trajectories for both going straight and changing lanes. The bottom right depicts a lane-change with a vehicle in the target lane, and VADv2 generates multiple reasonable trajectories.

# 5 CONCLUSIONS AND LIMITATIONS

In this work, we present VADv2, an end-to-end driving model based on probabilistic planning. In the CARLA simulator, VADv2 runs stably and achieves state-of-the-art closed-loop performance, significantly outperforming all existing methods. Comprehensive experiments on the NAVSIM and the 3DGS-based benchmark also demonstrate its impressive planning capability. The feasibility of this probabilistic planning paradigm is primarily validated. However, both simulator-based and 3DGS-based closed-loop environments still present limitations, such as overly simplistic agent behaviors and insufficient scene realism, which may restrict the performance of VADv2. In future work, we plan to explore how large-scale real-world driving data can be leveraged to further enhance planning performance and bridge the gap between simulation and real-world deployment.

## REFERENCES

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.
- Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019.
  - Raphael Chekroun, Marin Toromanoff, Sascha Hornauer, and Fabien Moutarde. Gri: General reinforced imitation and its application to vision-based autonomous driving. *arXiv preprint arXiv:2111.08575*, 2021.
- Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. 2020.
- Dian Chen, Vladlen Koltun, and Philipp Krähenbühl. Learning to drive from a world on rails. In *ICCV*, 2021.
- Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. *arXiv preprint arXiv:2310.01957*, 2023.
- Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. 2019a.
- Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *ICCV*, 2019b.
- Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *NeurIPS*, 2024.
- Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *CoRL*, 2017.
- Hao Gao, Shaoyu Chen, Bo Jiang, Bencheng Liao, Yiang Shi, Xiaoyang Guo, Yuechuan Pu, Haoran Yin, Xiangyu Li, Xinbang Zhang, et al. Rad: Training an end-to-end driving policy via large-scale 3dgs-based reinforcement learning. *arXiv preprint arXiv:2502.13144*, 2025.
- Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *CVPR*, 2020.
- Junru Gu, Chenxu Hu, Tianyuan Zhang, Xuanyao Chen, Yilun Wang, Yue Wang, and Hang Zhao. Vip3d: End-to-end visual trajectory prediction via 3d agent queries. *arXiv preprint arXiv:2208.01582*, 2022.
- Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive computing*, 2008.
- Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *ICCV*, 2021.
- Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zak Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022a.
- 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*, 2022b.
- 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. *CVPR2023*, 2022c.

- 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. 2023a.
  - 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*, 2023b.
    - Bo Jiang, Shaoyu Chen, Xinggang Wang, Bencheng Liao, Tianheng Cheng, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, and Chang Huang. Perceive, interact, predict: Learning dynamic and static clues for end-to-end motion prediction. *arXiv preprint arXiv:2212.02181*, 2022.
    - 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. *ICCV*, 2023.
    - Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
    - Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, 2022a.
    - Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv* preprint arXiv:2206.10092, 2022b.
    - Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.
    - 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. *arXiv preprint arXiv:2203.17270*, 2022c.
    - Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. *arXiv* preprint arXiv:2208.14437, 2022.
    - Bencheng Liao, Shaoyu Chen, Bo Jiang, Tianheng Cheng, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Lane graph as path: Continuity-preserving path-wise modeling for online lane graph construction. *arXiv preprint arXiv:2303.08815*, 2023a.
    - Bencheng Liao, Shaoyu Chen, Yunchi Zhang, Bo Jiang, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Maptrv2: An end-to-end framework for online vectorized hd map construction. *arXiv* preprint arXiv:2308.05736, 2023b.
    - Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *CVPR*, 2025.
    - Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *CVPR*, 2021.
  - Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. *arXiv preprint arXiv:2206.08920*, 2022.
  - Yukai Ma, Tiantian Wei, Naiting Zhong, Jianbiao Mei, Tao Hu, Licheng Wen, Xuemeng Yang, Botian Shi, and Yong Liu. Leapvad: A leap in autonomous driving via cognitive perception and dual-process thinking. *arXiv preprint arXiv:2501.08168*, 2025.
    - Jiageng Mao, Yuxi Qian, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv* preprint arXiv:2310.01415, 2023.

- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 2020.
  - Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020.
    - Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. NeurIPS, 1988.
    - Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *CVPR*, 2021a.
    - Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. 2021b.
    - Zhongyu Rao, Yingfeng Cai, Hai Wang, Yubo Lian, Yilin Zhong, Long Chen, and Yicheng Li. Enhancing autonomous driving: A low-cost monocular end-to-end framework with multi-task integration and temporal fusion. *IEEE Transactions on Intelligent Vehicles*, 2024.
    - Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. Languagempc: Large language models as decision makers for autonomous driving. *arXiv preprint arXiv:2310.03026*, 2023.
    - Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Conference on Robot Learning*, pp. 726–737. PMLR, 2023.
    - Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *ECCV*, 2024.
    - Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. *arXiv preprint arXiv:2303.11926*, 2023a.
    - Tianqi Wang, Enze Xie, Ruihang Chu, Zhenguo Li, and Ping Luo. Drivecot: Integrating chain-of-thought reasoning with end-to-end driving. *arXiv preprint arXiv:2403.16996*, 2024.
    - Wenhai Wang, Jiangwei Xie, ChuanYang Hu, Haoming Zou, Jianan Fan, Wenwen Tong, Yang Wen, Silei Wu, Hanming Deng, Zhiqi Li, et al. Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving. *arXiv preprint arXiv:2312.09245*, 2023b.
    - Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *CVPR*, 2024.
    - Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kenneth KY Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *arXiv preprint arXiv:2310.01412*, 2023.
    - Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. *arXiv preprint arXiv:2408.03601*, 2024.
    - 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.
    - Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. 2021.

Table 5: Closed-loop results on Town05 Short.

Method	Modality	DS ↑	RC↑
CILRS	С	7.5	13.4
LBC	C	31.0	55.0
Transfuser	C+L	54.5	78.4
ST-P3	C	55.1	86.7
VAD	C	64.3	87.3
LeapVAD	C	88.2	99.5
VADv2	C	89.7	93.0

Table 6: Ablation on vocabulary size.

Size	L2 (m) ↓			Collision (%) ↓			
Size	1s	2s	3s	1s	2s	3s	
256	0.110	0.207	0.337	0.000	0.019	0.057	
512	0.099	0.189	0.313	0.000	0.022	0.045	
1024	0.093	0.175	0.293	0.000	0.020	0.044	
2048	0.088	0.173	0.294	0.000	0.017	0.041	
4096	0.082	0.169	0.290	0.000	0.010	0.039	

Table 7: Ablation on the amount of training clips.

Amount		L2 (m) ↓	,	Collision (%) ↓		
Amount	1s	2s	3s	1s	2s	3s
$1 \times 10^{5}$	0.121	0.264	0.461	0.015	0.061	0.107
$3 \times 10^5$	0.082	0.169	0.290	0.000	0.010	0.039
$1 \times 10^{6}$	0.073	0.153	0.267	0.000	0.008	0.027
$3 \times 10^6$	0.072	0.133	0.225	0.000	0.000	0.007

Table 8: Ablation on planning manners.

Planning	DS ↑	RC↑	L2(m) @3s↓	Collision(%) @3s↓
Deterministic	74.6	95.1	0.223	0.006
Probabilistic	85.1	98.4	0.225	0.007

## A APPENDIX

## A.1 CLOSED-LOOP RESULTS ON THE TOWN05 SHORT BENCHMARK

We present the results for all publicly available works on the Town05 Short benchmark in Tab. 5. Compared to the Town05 Long benchmark, the Town05 Short benchmark focuses more on evaluating the ability of models to perform specific driving behaviors, such as lane changing in congested traffic flow and lane changing before intersections. VADv2 achieves the best performance on the main DS metric, while LeapVAD has a higher RC but a lower DS, indicating more infractions during evaluation. This demonstrates the comprehensive driving ability of VADv2 in complex driving scenarios.

#### A.2 ABLATION ON VOCABULARY SIZE

We ablate about the vocabulary size in Tab. 6. With the vocabulary size increasing, both L2 and collision metrics become better. A larger vocabulary size can better represent the action space with less discretization error.

#### A.3 ABLATION STUDY ON THE AMOUNT OF TRAINING CLIPS

Tab. 7 is the ablation experiments about the amount of clips of driving demonstrations used for training the end-to-end model. As expected, the model achieves better L2 and collision metrics with the data amount increasing.

# A.4 PROBABILISTIC PLANNING vs. DETERMINISTIC PLANNING

Ablation experiments about modeling manners of planning are presented in Tab. 8. In terms of deterministic modeling, following common practice (Jiang et al., 2023; Hu et al., 2022c; Prakash et al., 2021b), we change the planning head to an MLP, which directly regresses the future trajectory based on scene tokens, navigation information, and ego state. The results show that deterministic planning and probabilistic planning achieves similar performance in open-loop evaluation. However, in closed-loop evaluation, probabilistic planning is significantly better. Deterministic planning cannot cope with the uncertainty problem of planning and is more unstable in closed-loop evaluation.

#### A.5 More evaluation details on the 3DGS-based benchmark

Tab. 9 reports the inference latency, model parameters, and hardware platforms of baseline methods such as TransFuser (Prakash et al., 2021b) and VAD (Jiang et al., 2023). To ensure a fair comparison and accurately evaluate the planning performance improvements of our proposed strategy, we adopt the same perception backbone across all methods.

Table 9: More evaluation details on the 3DGS-based benchmark.

Method	Reference	Parameters	Latency	Collision (%)	Platform
TransFuser	TPAMI 22	0.36B	118ms	0.320	RTX 4090
VAD	ICCV 23	0.36B	118ms	0.335	RTX 4090
GenAD	ECCV 24	0.38B	121ms	0.341	RTX 4090
VADv2	Ours	0.40B	125ms	0.270	RTX 4090

As a result, the observed differences in inference latency are mainly attributed to variations in the design of the planning modules. While VADv2 introduces some computational overhead due to the use of planning vocabulary, its inference latency remains comparable to that of other baseline methods, and it significantly improves the primary collision rate metric.

Table 10: Quantitative analysis of the real-world 3DGS-based dataset.

Dataset	Duration	Environment	Straight	Turning
nuScenes	5.55h	Urban	92.80%	7.20%
NAVSIM	120h	Urban	66.40%	33.60%
Ours	2000h	Urban, Highway, Rural	52.50%	47.50%

Table 11: More details of the real-world 3DGS-based validation dataset.

Scenario	Type	Percentage
Sunny	Weather	74.78%
Night & Rainy	Weather	25.22%
Crowded Road	Precise Behavior	6.23%
Narrow Road	Precise Behavior	6.82%
Intersection	Precise Behavior	38.58%
Cut-in	Interactive Scenario	9.79%
Ped. Crossing	Interactive Scenario	9.20%

## A.6 More dataset details of the 3DGS-based benchmark

We provide additional details of our real-world dataset and include comparisons with other popular benchmarks in Tab. 10. The nuScenes dataset contains a high proportion of straight-driving scenarios, while NAVSIM includes more turning scenarios but still has a relatively small overall data volume. In contrast, our dataset offers a clear advantage in both data scale and scene diversity.

Tab. 11 highlights the wide range of challenging test scenarios in our dataset, supporting more robust and comprehensive evaluation. In addition, we evaluate the reconstruction quality of driving scenes under different weather conditions in our 3DGS-based environment. The PSNR (Peak Signal-to-Noise Ratio) metrics for sunny, rainy, and nighttime conditions were 29.5, 28.8, and 28.2, respectively, demonstrating a leading level of performance.