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

Shaoyu Chen[1⋆], Bo Jiang[1⋆], Hao Gao[1], Bencheng Liao[1], Qing Xu[2],
Qian Zhang[2], Chang Huang[2], Wenyu Liu[1], Xinggang Wang[1,✉]
[1] Huazhong University of Science & Technology [2] Horizon Robotics
{shaoyuchen, bjiang, g_hao, bcliao, liuwy, xgwang}@hust.edu.cn
{qing.xu, qian01.zhang, chang.huang}@horizon.cc
https://github.com/hustvl/VAD
https://hgao-cv.github.io/VADv2

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. In this work, to cope with the uncertainty problem, we propose VADv2, an end-to-end driving model based on probabilistic planning. VADv2 takes multi-view image sequences as input in a streaming manner, transforms sensor data into environmental token embeddings, 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 performance on the CARLA Town05 benchmark, significantly outperforming all existing methods. It runs stably in a fully end-to-end manner, even without the rule-based wrapper. Closed-loop demos are presented at https://hgao-cv.github.io/VADv2.

1. Introduction

End-to-end autonomous driving is an important and popular field recently. Mass of human driving demonstrations are easily available. It seems promising to learn a human-like driving policy from large-scale demonstrations.

However, the uncertainty and non-deterministic nature of planning make it challenging to extract the driving knowledge from driving demonstrations. To demonstrate such uncertainty, two scenarios are presented in Fig. 1. 1) Following another vehicle. The human driver has diverse reasonable driving maneuvers, keeping following or changing lanes to overtake. 2) Interaction with the coming vehicle. The human driver has two possible driving maneuvers, yield or overtake. From the perspective of statistics, the action (including the timing and speed) is highly stochastic, affected by many latent factors that can not be modeled.

Existing learning-based planning methods [23, 19, 21, 40, 16, 54] follow a deterministic paradigm to directly regress the action. The regression target $\hat{a}$ is the future trajectory in [23, 19, 21, 40] and control signal (acceleration and steering) in [16, 54]. Such a paradigm assumes there exists a deterministic relation between environment 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 (see Fig. 1), the deterministic modeling cannot cope with non-convex cases and may output an in-between action, causing safety problems. Besides, such deterministic regression-based planner tends to output the dominant trajectory, which appears the most in the training data (like stop or go straight), and results in undesirable planning performance.

In this work, we propose probabilistic planning to cope with the uncertainty of planning. As far as we know, VADv2 is the first work to use probabilistic modeling to fit the continuous planning action space, which is different from previous practices that use deterministic modeling for planning. We model the planning policy as an environment-conditioned non-stationary stochastic process, formulated as $p(a|o)$, where o is the historical and current observations of the driving environment, and a is a candidate planning action. Compared with deterministic modeling, probabilistic modeling can effectively capture the uncertainty in planning and achieve more accurate and safe planning performance.

The planning action space is a high-dimensional continuous spatiotemporal space. We resort to a probabilistic field function to model the mapping from the action space to the probabilistic distribution. Since directly fitting the continuous planning action space is not feasible, we discretize the

⋆ Equal contribution
✉ Corresponding author: xgwang@hust.edu.cn
This work is still in process.

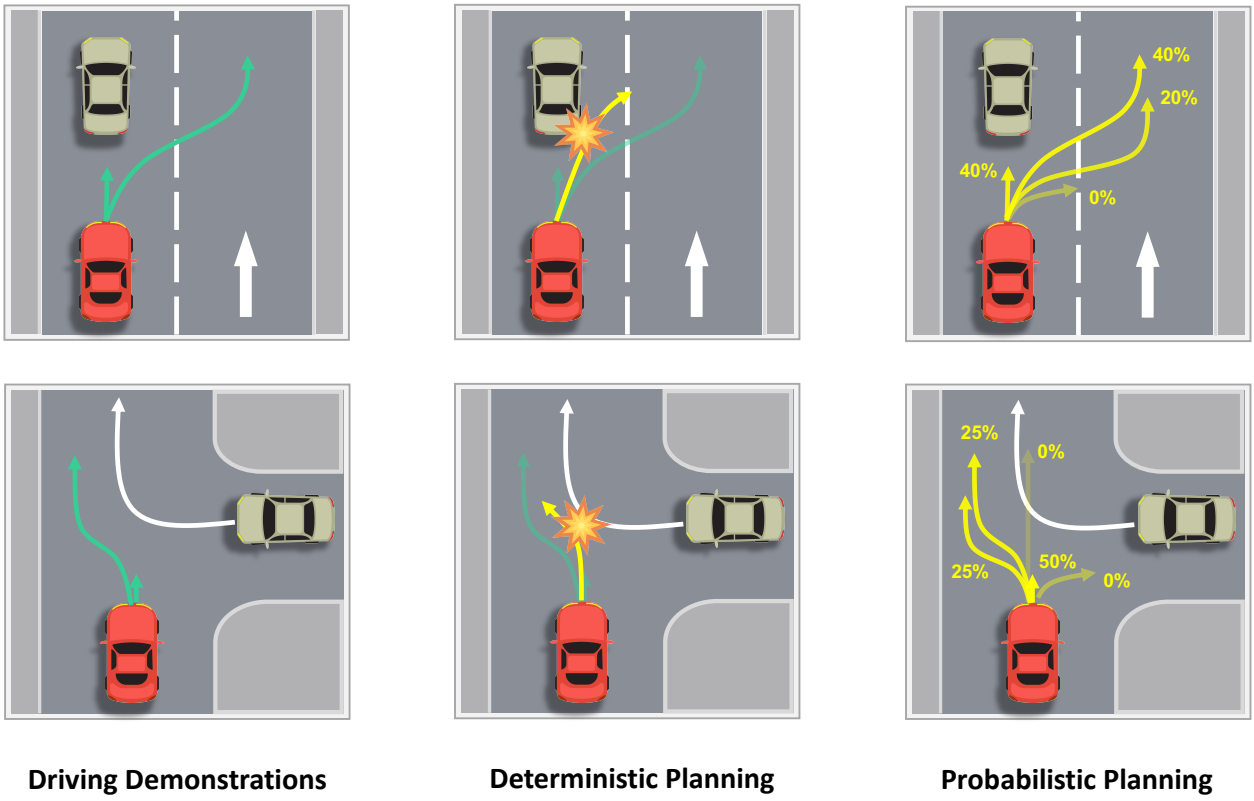

Figure 1. Uncertainty exists in planning. There doesn't exist a deterministic relation between environment and action. The deterministic planning fails to model such uncertainty especially when the feasible solution space is non-convex. VADv2 is based on probabilistic planning and learns the environment-conditioned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

planning action space to a large planning vocabulary and use mass driving demonstrations to learn the probability distribution of planning actions based on the planning vocabulary. For discretization, we collect all the trajectories in driving demonstrations and adopt the furthest trajectory sampling to select N representative trajectories which serve as the planning vocabulary.

Probabilistic planning has two other advantages. First, probabilistic planning models the correlation between each action and environment. Unlike deterministic modeling which only provides sparse supervision for the target planning action, probabilistic planning can provide supervision not only for the positive sample but also for all candidates in the planning vocabulary, which brings richer supervision information. Besides, probabilistic planning is flexible in the inference stage. It outputs multi-mode planning results and is easy to combine with rule-based and optimization-based planning methods. And 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, transforms sensor data into token embeddings, 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 performance on the CARLA Town05 benchmark, significantly outperforming all existing methods. Abundant closed-loop demos are presented at https://hgao-cv.github.io/VADv2. VADv2 runs stably in a fully end-to-end manner, even without the rule-based wrapper.

Our contributions are summarized as follows:

- We propose probabilistic planning to cope with the uncertainty 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 transforms sensor data into environmental token embeddings, outputs the probabilistic distribution of action, and samples one action to control the vehicle.
- In CARLA simulator, VADv2 achieves state-of-the-

art closed-loop performance on Town05 benchmark. Closed-loop demos show it runs stably in an end-to-end manner.

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** [38] is a pioneering work that achieves the perspective view to BEV transformation by explicitly predicting depth for image pixels. BEVFormer [26, 52], on the other hand, avoids explicit depth prediction by designing spatial and temporal attention mechanisms, and achieves impressive detection performance. Subsequent works [25, 48] continuously improve performance in downstream tasks by optimizing temporal modeling and BEV transformation strategies. In terms of vectorized mapping, HDMapNet [24] converts lane segmentation into vector maps through post-processing. VectorMapNet [32] predicts vector map elements in an autoregressive manner. MapTR [29, 30] introduces permutation equivalence and hierarchical matching strategies, significantly improving mapping performance. LaneGAP [28] introduces path-wise modeling for lane graphs.

Motion Prediction. Motion prediction aims to forecast the future trajectories of other traffic participants in driving scenes, assisting the ego vehicle in making informed planning decisions. Traditional motion prediction task utilizes input such as historical trajectories and high-definition maps to predict future trajectories. However, recent developments in end-to-end motion prediction methods [17, 53, 14, 22] perform perception and motion prediction jointly. In terms of scene representation, some works adopt rasterized image representations and employ CNN networks for prediction [3, 37]. Other approaches utilize vectorized representations and employ Graph Neural Networks [27] or Transformer models [13, 33, 36] for feature extraction and motion prediction. Some works [17, 53] see future motion as dense occupancy and flow instead of agent-level future waypoints. Some motion prediction methods [14, 22] adopt Gaussian Mixture Model (GMM) to regress multi-mode trajectories. It can be applied in planning to model uncertainty. But the number of modes is 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 [39, 8, 41] 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 [46, 5, 4]. By autonomously exploring driving behavior in closed-loop simulation environments, these approaches achieve or even surpass human-level driving performance. However, bridging the gap between simulation and reality, as well as addressing safety concerns, poses challenges in applying reinforcement learning strategies to real driving scenarios. Imitation learning [2, 18, 19, 23] 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 [19] cleverly integrates multiple perception and prediction tasks to enhance planning performance. VAD [23] explores the potential of vectorized scene representation for planning and getting rid of dense maps.

Large Language Model in Autonomous Driving. The interpretability and logical reasoning abilities demonstrated by large language models (LLMs) can greatly assist in the field of autonomous driving. Recent research has explored the combination of LLMs and autonomous driving[7, 10, 12, 44, 51, 34, 31, 50, 49]. One line of work utilizes LLMs for driving scene understanding and evaluation through question-answering (QA) tasks. Another approach goes a step further by incorporating planning on top of LLM-based scene understanding. For instance, DriveGPT4 [51] takes inputs such as historical video and text (including questions and additional information like historical control signals). After encoding, these inputs are fed into an LLM, which predicts answers to the questions and control signals. LanguageMPC [44], on the other hand, takes in historical ground truth perception results and HD maps in the form of language descriptions. It then utilizes a Chain of Thought analysis approach to understand the scene, and the LLM finally predicts planning actions from a predefined set as output. Each action corresponds to a specific control signal for execution. VADv2 draws inspiration from GPT [42, 43, 1, 47] to cope with the uncertainty problem. Uncertainty also exists in language modeling. Given a specific context, the next word is non-deterministic and probabilistic. 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 an environment-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.

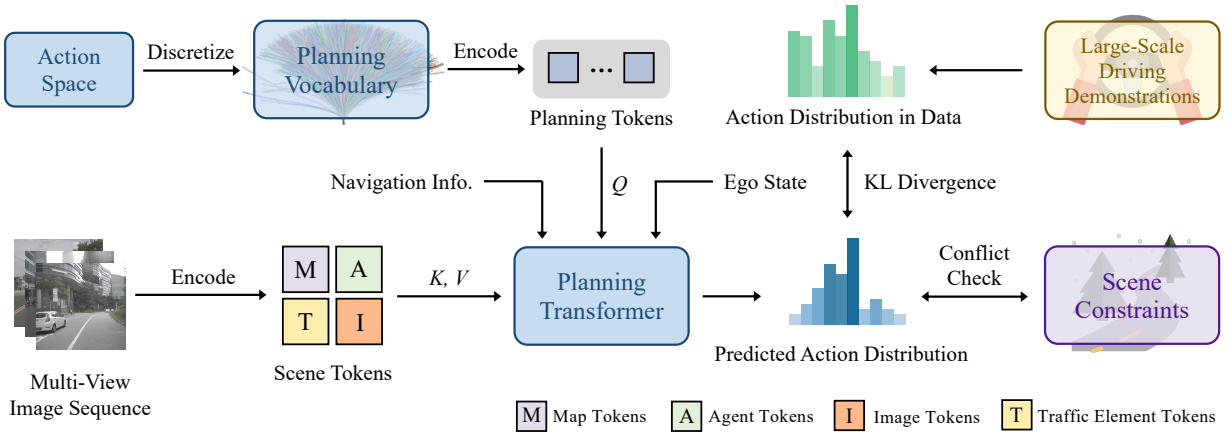

Figure 2. **Overall architecture of VADv2.** VADv2 takes multi-view image sequence as input in a streaming manner, transforms sensor data into environmental token embeddings, outputs the probabilistic distribution of action, and samples one action to control the vehicle. Large-scale driving demonstrations and scene constraints are used to supervise the predicted distribution.

3. Method

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 environmental 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

Information in the image is sparse and low-level. We use an encoder to transform the sensor data into instance-level token embeddings E_{env}, to explicitly extract high-level information. E_{env} includes four kinds of token: map token, agent token, traffic element token, and image token. VADv2 utilizes a group of map tokens [30, 29, 28] to predict the vectorized representation of the map (including lane centerline, lane divider, road boundary, and pedestrian crossing). Besides, VADv2 uses a group of agent tokens [22, 26] to predict other traffic participants' motion information (including location, orientation, size, speed, and multi-mode future trajectories). Traffic elements also play a vital role in planning. VADv2 transforms sensor data into traffic element tokens to predict the states of traffic elements. In CARLA, we consider two types of traffic signals: traffic light signals and stop signs. Map tokens, agent tokens, and traffic element tokens are supervised with corresponding supervision signals to make sure they explicitly encode corresponding high-level information. We also take image tokens as scene representations for planning, which contain rich information and are complementary to the instance-level tokens above. Besides, navigation information and ego state are also encoded into embeddings $\{E_{\text{navi}}, E_{\text{state}}\}$ with an MLP.

3.2. Probabilistic Planning

We propose probabilistic planning to cope with the uncertainty of planning. We model the planning policy as an environment-conditioned nonstationary stochastic process, formulated as $p(a|o)$. We approximate the planning action space as a probabilistic distribution based on large-scale driving demonstrations, and sample one action from the distribution at each time step to control the vehicle.

The planning action space is a high-dimensional continuous spatiotemporal space $\mathbb{A} = \{a | a \in \mathbb{R}^{2\text{T}}\}$. Since directly fitting the continuous planning action space is not feasible, we discretize the planning action space to a large planning vocabulary $\mathbb{V} = \{a^i\}^N$. Specifically, we collect all the planning actions in driving demonstrations and adopt the furthest trajectory sampling to select N representative actions to serve as the planning vocabulary. Each trajectory in $\mathbb{V}$ is sampled from driving demonstrations and thus naturally satisfies the kinematic constraints of the ego vehicle, which means that when the trajectory 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.

We represent each action in the planning vocabulary as a waypoint sequence $a = (x_1, y_1, x_2, y_2, ..., x_{\text{T}}, y_{\text{T}})$. Each waypoint corresponds to a future timestamp. 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 [35], which models the continuous radiance field over the 5D space (x, y, z, θ, ϕ), we resort to a probabilistic field to model the continuous mapping from the action space $\mathbb{A}$ to the probabilistic distribution $\{p(a)|a \in \mathbb{A}\}$. We encode each action (trajectory)

into a high-dimensional planning token embedding $E(a)$, use a cascaded Transformer decoder for interaction with environmental information E_{env}, and combine with Navigation information E_{navi} and ego state E_{state} to output the probability, *i.e.*,

$$
\begin{aligned}
&p(a) = \text{MLP}(\text{Transformer}(E(a), E_{\text{env}}) + E_{\text{navi}} + E_{\text{state}}),\\
&q = E(a), k = v = E_{\text{env}},\\
&a = (x_1, y_1, x_2, y_2, ..., x_{\text{T}}, y_{\text{T}}),\\
&E(a) = \text{Cat}\Big[\Gamma(x_1), \Gamma(y_1), \Gamma(x_2), \Gamma(y_2), ..., \Gamma(x_{\text{T}}), \Gamma(y_{\text{T}})\Big],\\
&\Gamma(pos) = \text{Cat}\Big[\gamma(pos, 0), \gamma(pos, 1), ..., \gamma(pos, L-1)\Big],\\
&\gamma(pos, j) = \text{Cat}\Big[\cos(pos/10000^{2\pi j/L}, \sin(pos/10000^{2\pi j/L}\Big].
\end{aligned}
\tag{1}
$$

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

3.3. Training

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

$$\mathcal{L} = \mathcal{L}_{\text{distribution}} + \mathcal{L}_{\text{conflict}} + \mathcal{L}_{\text{token}}. \tag{2}$$

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 distribution of the data.

$$\mathcal{L}_{\text{distribution}} = D_{\text{KL}}(p_{\text{data}} || p_{\text{pred}}). \tag{3}$$

In the training phase, the ground truth trajectory is added to the planning vocabulary as the positive sample. Other trajectories are regarded as negative samples. We assign different loss weights to negative trajectories. Trajectories close to the ground truth trajectory are less penalized.

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

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

The loss of map tokens is the same with MapTRv2 [30]. 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, which is the same with VAD [23]. 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.

3.4. Inference

In closed-loop inference, it's flexible to get the driving policy π_{model} 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 trajectory 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 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 PnC and learning-based PnC.

4. Experiments

4.1. Experimental Settings

The widely used CARLA [11] simulator is adopted to evaluate the performance of VADv2. Following common practice, we use Town05 Long and Town05 Short 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

Method	Modality	Reference	Driving Score ↑	Route Completion ↑	Infraction Score ↑
CILRS [9]	C	CVPR 19	7.8	10.3	0.75
LBC [6]	C	CoRL 20	12.3	31.9	0.66
Roach [54]	C	ICCV 21	41.6	96.4	0.43
Transfuser† [40]	C+L	TPAMI 22	31.0	47.5	0.77
ST-P3 [18]	C	ECCV 22	11.5	83.2	-
VAD [23]	C	ICCV 23	30.3	75.2	-
ThinkTwice [21]	C+L	CVPR 23	70.9	95.5	0.75
MILE [16]	C	NeurIPS 22	61.1	97.4	0.63
Interfuser [45]	C	CoRL 22	68.3	95.0	-
DriveAdapter+TCP [20]	C+L	ICCV 23	71.9	97.3	0.74
DriveMLM [49]	C+L	arXiv	76.1	98.1	0.78
VADv2	C	Ours	85.1	98.4	0.87

Table 1. **Closed-loop evaluation on Town05 Long benchmark.**

Method	Modality	Driving Score ↑	Route Completion ↑
CILRS [9]	C	7.5	13.4
LBC [6]	C	31.0	55.0
Transfuser [40]	C+L	54.5	78.4
ST-P3 [18]	C	55.1	86.7
VAD [23]	C	64.3	87.3
VADv2	C	89.7	93.0

Table 2. **Closed-loop evaluation on Town05 Short benchmark.**

evaluating the model's performance in specific scenarios, such as lane changing before intersections.

We use the official autonomous agent of CARLA to collect training data by randomly generating driving routes in Town03, Town04, Town06, Town07, and Town10. The data is sampled at a frequency of 2Hz, and we collect approximately 3 million frames for training. For each frame, we save 6-camera surround-view images, traffic signals, information about other traffic participants, and the state information of the ego vehicle. Additionally, we obtain the vectorized maps for training the online mapping module by preprocessing the OpenStreetMap [15] format maps provided by CARLA. It is important to note that the map information was only provided as ground truth during training, and VADv2 does not utilize any high-definition map in closed-loop evaluation.

4.2. Metrics

For closed-loop evaluation, we use the official metrics of CARLA. Route Completion indicates the percentage of the route distance completed by an agent. Infraction Score indicates the degree of infractions happening along the route. Typical infractions include running red lights, collisions with pedestrians, *etc.*. Each type of infraction has a corresponding penalty coefficient, with more infractions happening, Infraction Score becomes lower. Driving Score serves as the product between the Route Completion and the Infraction Score, which is the main metric for evaluation. In benchmark evaluation, most works adopt a rule-based wrapper to reduce the infraction. For fair comparisons with other methods, we follow the common practice of adopting a rule-based wrapper over the learning-based policy.

For open-loop evaluation, L2 distance and collision rate are adopted to show which degree the learned policy drives similar to the expert demonstrations. In ablation experiments, we adopt open-loop metrics for evaluation, considering open-loop metrics are fast to calculate 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.

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

On the Town05 Long benchmark, VADv2 achieved a Drive Score of 85.1, a Route Completion of 98.4, and an Infraction Score of 0.87, as shown in Tab. 1. Compared to the previous state-of-the-art method [49], VADv2 achieves a higher Route Completion while significantly improving Drive Score by 9.0. It is worth noting that VADv2 only utilizes cameras as perception input, whereas [49] utilizes both cameras and LiDAR. Furthermore, compared to the previous best method [45] which only relies on cameras, VADv2 demonstrates even greater advantages, with a remarkable increase in Drive Score of up to 16.8.

We present the results for all publicly available works on the Town05 Short benchmark in Tab. 2. 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 con-

ID	Dist. Loss	Conflict Loss	Agent Token	Map Token	Traf. Elem. Token	Image Token	L2 (m) ↓ 1s	2s	3s	Collision (%) ↓ 1s	2s	3s
1		✓	✓	✓	✓	✓	1.415	2.310	3.153	0.698	0.755	0.746
2	✓		✓	✓	✓	✓	0.086	0.173	0.291	0.0	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

Table 3. **Ablation for design choices.** "Dist. Loss" denotes Distribution Loss. "Traf. Elem. Token" denotes Traffic Element Token.

gested traffic flow and lane changing before intersections. In comparison to the previous result [23], VADv2 significantly improves Drive Score and Route Completion by 25.3 and 5.7 respectively, demonstrating the comprehensive driving ability of VADv2 in complex driving scenarios.

4.4. Ablation Study

Tab. 3 shows the ablation experiments of the key modules in VADv2. 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, so without the Conflict Loss (ID 2), the model's planning accuracy is also affected. 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).

4.5. Visualization

Fig. 3 presents some qualitative results of VADv2. The first image showcases multi-modal planning trajectories predicted by VADv2 at different driving speeds. The second image showcases VADv2's predictions of both forward creeping and multi-modal left-turn trajectories in a lane-changing scenario. The third image depicts a right lane-changing scenario at an intersection, where VADv2 predicts multiple trajectories for both going straight and changing lanes to the right. The final image demonstrates a lane-changing scenario where there is a vehicle in the target lane, and VADv2 predicts multiple reasonable lane-changing trajectories.

5. Conclusion

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. The feasibility of this probabilistic paradigm is primarily validated. However, its effectiveness in more complicated real-world scenarios remains unexplored, which is the future work.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[2] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *CVPR*, 2021. 3

[3] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019. 3

[4] 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. 3

[5] Dian Chen, Vladlen Koltun, and Philipp Krähenbühl. Learning to drive from a world on rails. In *ICCV*, 2021. 3

[6] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. 2020. 6

[7] 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. 3

[8] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *ICCV*, 2019. 3

[9] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. 2019. 6

[10] Xinpeng Ding, Jianhua Han, Hang Xu, Wei Zhang, and Xiaomeng Li. Hilm-d: Towards high-resolution understanding in multimodal large language models for autonomous driving. *arXiv preprint arXiv:2309.05186*, 2023. 3

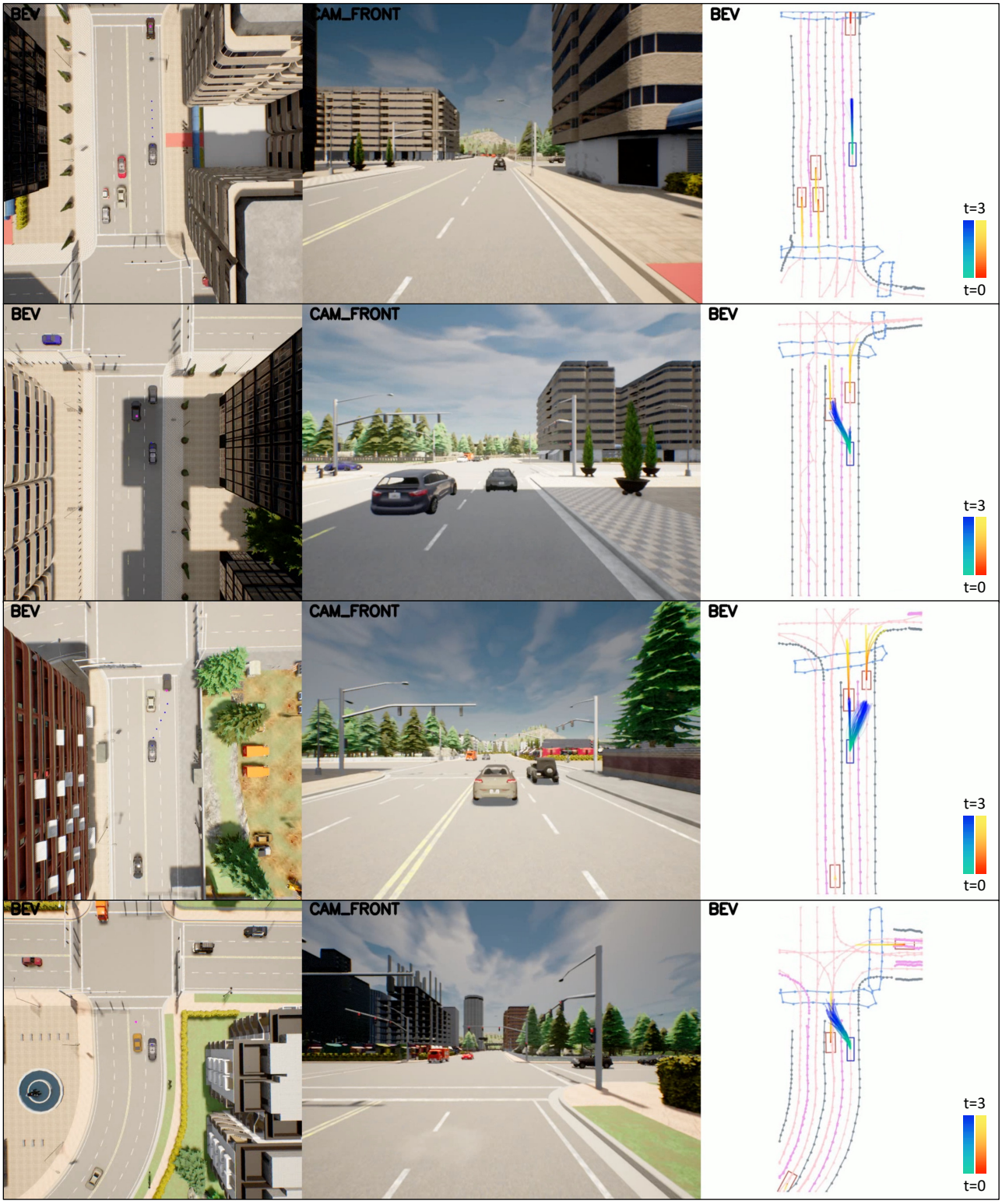

Figure 3. Qualitative results of VADv2.

[11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 5

[12] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive like a human: Rethinking autonomous driving with large language models. *arXiv preprint arXiv:2307.07162*, 2023. 3

[13] 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. 3

[14] 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. 3

[15] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive computing*, 2008. 6

[16] 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)*, 2022. 1, 6

[17] 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. 3

[18] 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*, 2022. 3, 6

[19] 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*, 2022. 1, 3

[20] 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. 2023. 6

[21] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *CVPR*, 2023. 1, 6

[22] 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. 3, 4

[23] 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. 1, 3, 5, 6, 7

[24] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, 2022. 3

[25] 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*, 2022. 3

[26] 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*, 2022. 3, 4

[27] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020. 3

[28] 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*, 2023. 3, 4

[29] 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. 3, 4

[30] 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*, 2023. 3, 4, 5

[31] Jiaqi Liu, Peng Hang, Jianqiang Wang, Jian Sun, et al. Mtd-gpt: A multi-task decision-making gpt model for autonomous driving at unsignalized intersections. *arXiv preprint arXiv:2307.16118*, 2023. 3

[32] Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. *arXiv preprint arXiv:2206.08920*, 2022. 3

[33] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *CVPR*, 2021. 3

[34] Jiageng Mao, Yuxi Qian, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*, 2023. 3

[35] 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. 4

[36] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. *arXiv preprint arXiv:2106.08417*, 2021. 3

[37] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *CVPR*, 2020. 3

[38] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 3

[39] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *NeurIPS*, 1988. 3

[40] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. 2021. 1, 6

[41] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *CVPR*, 2021. 3

[42] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 3

[43] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 3

[44] 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. 3

[45] 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*, pages 726–737. PMLR, 2023. 6

[46] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *CVPR*, 2020. 3

[47] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[48] 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*, 2023. 3

[49] 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*, 2023. 3, 6

[50] Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. Dilu: A knowledge-driven approach to autonomous driving with large language models. *arXiv preprint arXiv:2309.16292*, 2023. 3

[51] 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. 3

[52] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. *arXiv preprint arXiv:2211.10439*, 2022. 3

[53] 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. 3

[54] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 6