Perception in Plan: Coupled Perception and Planning for End-to-End Autonomous Driving

Bozhou Zhang[1,2*], **Jingyu Li**[1,2*], **Nan Song**[1,2], **Li Zhang**[1,2†]

[1]School of Data Science, Fudan University
[2]Shanghai Innovation Institute
lizhangfd@fudan.edu.cn

Abstract

End-to-end autonomous driving has achieved remarkable advancements in recent years. Existing methods primarily follow a perception–planning paradigm, where perception and planning are executed sequentially within a fully differentiable framework for planning-oriented optimization. We further advance this paradigm through a "perception-in-plan" framework design, which integrates perception into the planning process. This design facilitates targeted perception guided by evolving planning objectives over time, ultimately enhancing planning performance. Building on this insight, we introduce **VeteranAD**, a coupled perception and planning framework for end-to-end autonomous driving. By incorporating multi-mode anchored trajectories as planning priors, the perception module is specifically designed to gather traffic elements along these trajectories, enabling comprehensive and targeted perception. Planning trajectories are then generated based on both the perception results and the planning priors. To make perception fully serve planning, we adopt an autoregressive strategy that progressively predicts future trajectories while focusing on relevant regions for targeted perception at each step. With this simple yet effective design, VeteranAD fully unleashes the potential of planning-oriented end-to-end methods, leading to more accurate and reliable driving behavior. Extensive experiments on the NAVSIM and Bench2Drive datasets demonstrate that our VeteranAD achieves state-of-the-art performance.

Code —
https://github.com/LogosRoboticsGroup/VeteranAD

Introduction

End-to-end autonomous driving (Hu et al. 2023; Jiang et al. 2023; Prakash, Chitta, and Geiger 2021) has made significant progress in recent years by unifying multiple tasks, including perception (Li et al. 2022b; Wang et al. 2023; Liao et al. 2023), prediction (Shi et al. 2022; Zhou et al. 2023), and planning (Dauner et al. 2023a; Cheng et al. 2024), into a unified framework. In this way, the end-to-end driving framework builds a fully differentiable learning system that ensures planning-oriented optimization. This enables impressive performance in both open-loop (Zheng et al. 2024; Sun et al. 2025; Zhang et al. 2025a) and closed-loop (Jia et al. 2023a,b, 2025) planning.

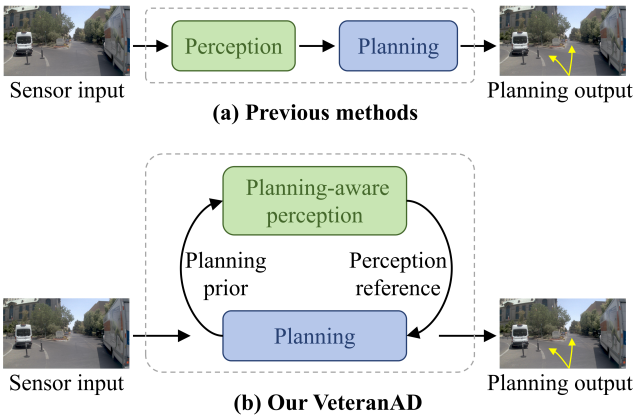

Figure 1: Comparison of end-to-end autonomous driving methods. **(a)** Previous approaches mainly follow a perception–planning paradigm, executing these modules sequentially. **(b)** In contrast, our VeteranAD *integrates perception into the planning process*, using planning priors to guide perception and leveraging targeted perception results to inform planning. This "perception-in-plan" paradigm enhances the planning-oriented framework. For clarity, other intermediate modules are omitted in the figure.

Mainstream end-to-end autonomous driving methods (Hu et al. 2023; Jiang et al. 2023; Sun et al. 2025; Chitta et al. 2023) typically adopt a sequential paradigm, where perception is followed by planning, as shown in Figure 1 (a). A Transformer-based architecture (Vaswani 2017) is often employed to make the entire pipeline differentiable, thereby enabling planning-oriented optimization. However, differentiability alone is insufficient to fully exploit the advantages of planning-oriented optimization in end-to-end autonomous driving, whose goal is to ensure that all preceding modules—such as perception—are optimized to better serve the planning process.

To address the aforementioned limitation, we propose a "perception-in-plan" paradigm, which integrates perception

*These authors contributed equally.
†Corresponding author.

into the planning process. In this way, the perception module operates in a targeted manner, aligned with the requirements of the planning process. Based on this paradigm, we introduce **VeteranAD**, as illustrated in Figure 1 (b). In our framework, perception and planning are tightly coupled. Multi-mode anchored trajectories are used as planning priors to guide the perception module in gathering traffic elements—such as lanes and surrounding agents—along the predicted trajectories, enabling holistic and targeted perception for planning. To fully inject perception into planning, we adopt an autoregressive strategy that progressively generates future trajectories. At each time step, guided by planning priors, the model focuses on relevant regions to perform targeted perception and generate the planning output for the corresponding step. Under this paradigm, we design two core modules: Planning-Aware Holistic Perception and Localized Autoregressive Trajectory Planning. The Planning-Aware Holistic Perception module operates across three dimensions: image features, bird's-eye-view (BEV) features, and surrounding agent features. This interaction enables a comprehensive understanding of traffic elements, including vehicles, lanes, and barriers. The Localized Autoregressive Trajectory Planning module decodes future trajectories in an autoregressive manner. It iteratively adjusts the anchor trajectories from near to far future based on perception results, ensuring context-aware and progressively refined planning. Through the above design, VeteranAD leverages trajectory priors to enable focused perception and progressive planning, thereby achieving strong end-to-end planning performance.

Our **contributions** are summarized as follows: (i) We propose VeteranAD, a novel framework that follows a "perception-in-plan" paradigm, integrating perception into the planning process. (ii) We design two key modules: the Planning-Aware Holistic Perception module and the Localized Autoregressive Trajectory Planning module, which jointly couple perception and planning, fully unleashing the planning-oriented optimization advantages enabled by end-to-end autonomous driving. (iii) Extensive experiments on the NAVSIM and Bench2Drive datasets demonstrate that VeteranAD achieves state-of-the-art performance.

Related Work

End-to-end autonomous driving. In the early stages of autonomous driving, rule-based methods adopted a modular design (Treiber, Hennecke, and Helbing 2000; Thrun et al. 2006; Bacha et al. 2008; Dauner et al. 2023b), dividing the system into separate components—perception, prediction, planning, and control—connected via predefined rules. While interpretable, this architecture suffers from error propagation and limited scenario coverage. End-to-end planning methods (Prakash, Chitta, and Geiger 2021; Liao et al. 2025) gradually replace individual modules, such as perception and planning, with deep learning-based subnetworks, while retaining essential rule-based constraints. This paradigm has gained traction for its ability to integrate perception, prediction, and planning into a unified framework, removing the need for hand-crafted intermediate representations. Early works (Prakash, Chitta, and Geiger 2021; Chitta et al. 2023) typically bypassed intermediate tasks like perception and motion prediction. ST-P3 (Hu et al. 2022) was the first to introduce explicit intermediate representations in a surround-view camera-based framework. UniAD (Hu et al. 2023) further unified perception, prediction, and planning using transformer-based query interactions, achieving strong performance on the nuScenes (Caesar et al. 2020) benchmark. Recent advances explore diverse representations and learning paradigms. VAD (Jiang et al. 2023) proposed vectorized scene representations, while VADv2 (Chen et al. 2024) introduced probabilistic planning with a 4K trajectory vocabulary and conflict-aware loss, yielding state-of-the-art closed-loop performance on CARLA Town05 (Jia et al. 2023a,b). SparseDrive (Sun et al. 2025) enhances efficiency through sparse scene representations and a parallel motion planner. GenAD (Zheng et al. 2024) adopts a generative framework that unifies motion prediction and planning using an instance-centric scene representation and structured latent modeling via variational autoencoders.

Closed-loop and open-loop benchmarking. Closed-loop and open-loop benchmarks are commonly used to evaluate autonomous driving systems. Closed-loop evaluation simulates the full feedback loop—from sensor input to control execution—using tools such as nuPlan (Karnchanachari et al. 2024), Waymax (Gulino et al. 2023), CARLA (Dosovitskiy et al. 2017), Bench2Drive (Jia et al. 2024), and MetaDrive (Li et al. 2022a). These simulators enable measurement of driving metrics such as collision rate and ride comfort. However, simulating realistic traffic behavior and sensor data remains challenging. Graphics-based rendering introduces domain gaps (Ljungbergh et al. 2024), while data-driven sensor simulation suffers from limited visual quality (Amini et al. 2020, 2022; Wang et al. 2022a). Open-loop evaluation tests trajectory prediction on offline datasets like nuScenes (Caesar et al. 2020), without interaction with the environment. Due to the lack of standardized planning metrics, prior works often rely on custom implementations, leading to inconsistent results (Weng et al. 2024; Li et al. 2024c).

Methodology

Preliminary

Task formulation. End-to-end autonomous driving takes sensor data (such as camera and LiDAR) as input and generates future planning trajectories as output. The planning task typically involves generating multi-mode trajectories to represent multiple possible future plans. Auxiliary tasks, such as detection, map segmentation, and motion prediction for surrounding agents, are also integrated into the end-to-end models to help the model better learn scene features for safe planning results.

Framework overview. The framework of our VeteranAD is illustrated in Figure 2. It comprises three main components: an image encoder, the Planning-Aware Holistic Perception module, and the Localized Autoregressive Trajectory Planning module. First, the image encoder extracts fea-

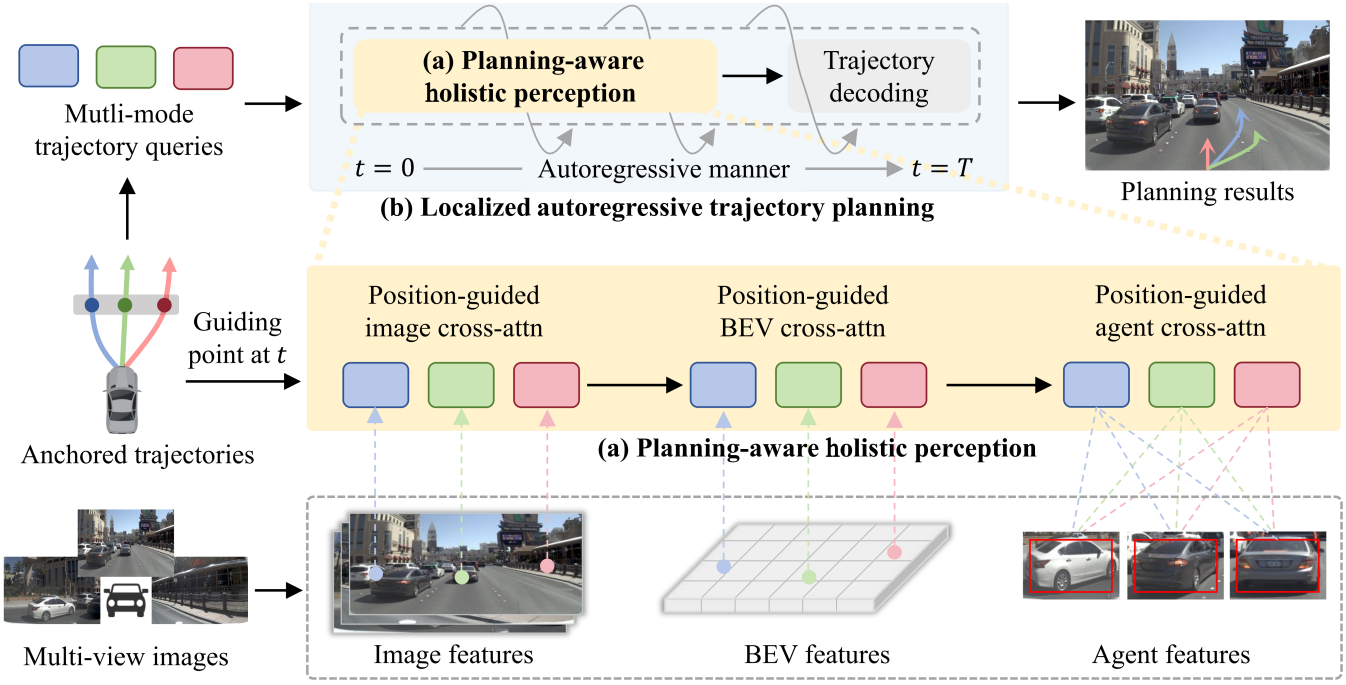

Figure 2: Overview of the **VeteranAD** framework. Multi-mode planning queries are initialized from the anchored trajectories. And multi-view images are processed by the encoder to generate image features, BEV features, and agent features. **(a)** The Planning-Aware Holistic Perception module takes the planning queries and performs cross-attention with image features, BEV features, and surrounding agent features, guided by the points on the anchored trajectories. **(b)** The Localized Autoregressive Trajectory Planning module adjusts the planning trajectories, derived from the anchored trajectories, using the outputs from (a). It performs perception at each time step to generate the corresponding planning waypoint in an autoregressive manner, resulting in the complete planning trajectory.

tures from multi-view images, producing image features, BEV features, and surrounding agent features. Next, multi-mode trajectory queries are initialized from the anchored trajectories. The Planning-Aware Holistic Perception module performs position-guided cross-attention between the trajectory queries and the extracted image, BEV, and agent features. The Localized Autoregressive Trajectory Planning module then operates in an autoregressive manner, performing perception at each time step and adjusting the anchored trajectory point accordingly, ultimately generating the complete planning output.

Image Encoding

Given multi-view images $I \in \mathbb{R}^{N_{\text{img}} \times 3 \times H \times W}$, where N_{img} denotes the number of camera views, the image encoder (He et al. 2016) first extracts multi-view image features, denoted as F_{img}. Then, bird's-eye-view (BEV) features F_{BEV} are generated from image features using the LSS (Philion and Fidler 2020) method. A simple multi-layer perceptron (MLP) decoder is then applied to decode the BEV features into a BEV segmentation map, which is supervised using the ground truth segmentation map. The surrounding agent features F_{agent} are initialized and interact with BEV features through Transformer (Vaswani 2017) blocks. A simple MLP decoder then decodes the agent features into bounding boxes, which are supervised using the ground truth bounding boxes of the surrounding agents. The process is shown below:

$$\begin{aligned} F_{\text{img}} &= \text{ImageEncoder}(I), \\ F_{\text{BEV}} &= \text{ImageToBEV}(F_{\text{img}}), \\ F_{\text{agent}} &= \text{Transformer}(\text{Q} = F_{\text{agent}}, \text{K}, \text{V} = F_{\text{BEV}}). \end{aligned} \tag{1}$$

After obtaining these features, the multi-mode trajectory queries $Q_{\text{traj}} \in \mathbb{R}^{M \times C}$ are initialized from the anchored trajectories, where M denotes the number of planning modes and C represents the feature channels. The anchored trajectories are clustered from the ground truth planning trajectories following previous work (Sun et al. 2025; Liao et al. 2025).

Planning-Aware Holistic Perception

The perception module enables the trajectory queries to comprehensively capture the scene and traffic elements, such as lanes, vehicles, pedestrians, and barriers, ensuring accurate and safe planning. Given the trajectory queries $Q_{\text{traj}} \in \mathbb{R}^{M \times C}$, three types of cross-attention are employed to interact with image features, BEV features, and agent features.

The position-guided image cross-attention and position-guided BEV cross-attention are designed to selectively

gather features along potential planning trajectories. First, the guiding points $P_{\mathrm{t}} \in \mathbb{R}^{M\times 3}$ at time t are extracted from the anchored trajectories, which serve as the planning prior. These guiding points are then projected onto both the image and BEV coordinates. Following previous works (Wang et al. 2022b; Liu et al. 2023), they serve as reference points for cross-attention between the trajectory queries and the image and BEV features. The process is shown below:

$$\begin{aligned} Q_{\mathrm{traj}} &= \mathrm{CrossAttn}(\mathrm{Q}=Q_{\mathrm{traj}}, \mathrm{K}, \mathrm{V}=F_{\mathrm{img}}), \\ Q_{\mathrm{traj}} &= \mathrm{CrossAttn}(\mathrm{Q}=Q_{\mathrm{traj}}, \mathrm{K}, \mathrm{V}=F_{\mathrm{BEV}}). \end{aligned} \tag{2}$$

The position-guided agent cross-attention is designed to effectively differentiate the importance of surrounding agents based on their distance. As introduced in the image encoding section, bounding boxes are decoded, allowing the positions of agents to be obtained. The pairwise relative distances between the surrounding agents and the ego agent are then computed using the guiding points and the decoded agent positions.

The relative distances are first encoded by an MLP to obtain the relative distance feature F_{DisRel}. This feature is then concatenated with the agent features and trajectory queries to form the distance-aware agent features F_{AgRel}. Inspired by previous works (Zhou et al. 2023; Zhang et al. 2024), cross-attention is subsequently applied to enable interaction between the trajectory queries and the distance-aware agent features, after aligning their dimensions. The overall process is illustrated below:

$$\begin{aligned} F_{\mathrm{AgRel}} &= \mathrm{Concat}(Q_{\mathrm{traj}}, F_{\mathrm{agent}}, F_{\mathrm{DisRel}}), \\ Q_{\mathrm{traj}} &= \mathrm{CrossAttn}(\mathrm{Q}=Q_{\mathrm{traj}}, \mathrm{K}, \mathrm{V}=F_{\mathrm{AgRel}}). \end{aligned} \tag{3}$$

Localized Autoregressive Trajectory Planning

The trajectory planning module aims to use the anchored trajectories as coarse planning trajectories and incorporate scene features to generate the final planning trajectories. For the anchored multi-mode trajectories over the future T steps, we obtain the trajectory points set $\{P_1, ..., P_{\mathrm{T}}\}$, where $P_{\mathrm{t}} \in \mathbb{R}^{M\times 3}$ is the same as mentioned above at time t. These trajectory points serve as guiding points for trajectory planning. The process operates in an autoregressive manner. At each time step t, the module takes the trajectory queries Q_{traj} and the guiding point P_{t} as input, while the Planning-Aware Holistic Perception module interacts with the trajectory queries and the scene features. Then, an MLP trajectory decoder is employed to predict the future trajectory point at time step t. The model estimates the offset ΔP_{ft} to refine the guiding point, producing the final planned trajectory point P_{ft}, as shown below:

$$P_{\mathrm{ft}} = \Delta P_{\mathrm{ft}} + P_{\mathrm{t}}. \tag{4}$$

Finally, we obtain the planned trajectory point set $\{P_{\mathrm{f1}}, ..., P_{\mathrm{fT}}\}$, which forms the final planning trajectories $P_{\mathrm{f}} \in \mathbb{R}^{M\times T\times 3}$. At the final time step T, the module decodes the classification score $S_{\mathrm{f}} \in \mathbb{R}^{M\times 1}$ for the multi-mode trajectories. To model the movement of trajectory points, we employ Motion-Aware Layer Normalization (Wang et al. 2023) to transform trajectory queries from time $t-1$ to time t, conditioned on the guiding points at time t, inspired by previous works (Wang et al. 2023; Song et al. 2024).

End-to-end Learning

The loss function consists of four components: the BEV segmentation map loss $\mathcal{L}_{\mathrm{BEV}}$, the agent bounding box loss $\mathcal{L}_{\mathrm{agent}}$, the planning regression loss $\mathcal{L}_{\mathrm{reg}}$, and the planning classification loss $\mathcal{L}_{\mathrm{cls}}$. The BEV segmentation map loss is calculated using cross-entropy loss. The agent bounding box loss is divided into L1 loss for box position regression and binary cross-entropy loss for box label classification. The planning regression loss is L1 loss, while the planning classification loss is computed using Focal loss. The overall loss function for end-to-end training is as follows:

$$\mathcal{L}_{total} = \lambda_1\mathcal{L}_{BEV} + \lambda_2\mathcal{L}_{agent} + \lambda_3\mathcal{L}_{reg} + \lambda_4\mathcal{L}_{cls}, \tag{5}$$

where λ_1, λ_2, λ_3, and λ_4 are the balancing factors.

Experiments

Experimental Settings

Datasets. NAVSIM (Dauner et al. 2024) is a large-scale real-world autonomous driving dataset designed for non-reactive simulation and benchmarking. It focuses on challenging scenarios involving dynamic intention changes, while filtering out trivial cases such as stationary or constant-speed driving. NAVSIM provides sensor data from cameras and LiDAR, along with annotated HD maps and object bounding boxes at 2 Hz. It is split into two subsets: navtrain (1,192 scenarios) for training and validation, and navtest (136 scenarios) for testing.

Bench2Drive (Jia et al. 2024) is the first closed-loop evaluation benchmark tailored for end-to-end autonomous driving. It addresses the limitations of traditional open-loop evaluations by offering a more realistic and interactive testing setup. The training set contains 10,000 short clips generated in the CARLA v2 (Dosovitskiy et al. 2017) simulator, while the evaluation set includes 220 independent short routes.

Evaluation metrics. For the NAVSIM dataset, we evaluate our method using the PDM Score (PDMS) as defined in the official benchmark. PDMS consists of several subscores: No At-Fault Collisions (NC), Drivable Area Compliance (DAC), Time-to-Collision (TTC), Comfort (Comf.), and Ego Progress (EP). For the Bench2Drive dataset, we follow the official evaluation protocols. In open-loop evaluation, we use the Average L2 Error. For closed-loop evaluation, we adopt the Driving Score and Success Rate. Further details are provided in the appendix.

Implementation details. The model is trained using 8 NVIDIA GeForce RTX 3090 GPUs, with a total batch size of 32 for 16 epochs. The learning rate and weight decay are set to 2×10^{-4} and 1×10^{-4}, respectively, and the model is optimized with AdamW. For fair comparison, the image backbone follows prior works and adopts ResNet-34 (He et al. 2016). The input consists of three images, front-right, front, and front-left, which are resized to 768×432. The

Method	Input	NC ↑	DAC ↑	TTC ↑	Comf. ↑	EP ↑	PDMS ↑
VADv2-$\mathcal{V}_{8192}$ (Chen et al. 2024)	C & L	97.2	89.1	91.6	**100**	76.0	80.9
Hydra-MDP-$\mathcal{V}_{8192}$ (Li et al. 2024b)	C & L	97.9	91.7	92.9	**100**	77.6	83.0
UniAD (Hu et al. 2023)	Camera	97.8	91.9	92.9	**100**	78.8	83.4
LTF (Prakash, Chitta, and Geiger 2021)	Camera	97.4	92.8	92.4	**100**	79.0	83.8
PARA-Drive (Weng et al. 2024)	Camera	97.9	92.4	93.0	99.8	79.3	84.0
Transfuser (Prakash, Chitta, and Geiger 2021)	C & L	97.7	92.8	92.8	**100**	79.2	84.0
DRAMA (Yuan et al. 2024)	C & L	98.0	93.1	94.8	**100**	80.1	85.5
Hydra-MDP++ (Li et al. 2024a)	Camera	97.6	96.0	93.1	**100**	80.4	86.6
DiffusionDrive (Liao et al. 2025)	C & L	98.2	96.2	94.7	**100**	<u>82.2</u>	88.1
WoTE (Li et al. 2025)	C & L	<u>98.5</u>	<u>96.8</u>	<u>94.9</u>	<u>99.9</u>	81.9	<u>88.3</u>
VeteranAD (Ours)	Camera	**99.1**	**98.3**	**96.1**	**100**	**83.1**	**90.2**

Table 1: Performance comparison *on the NAVSIM (Dauner et al. 2024) dataset for the navtest split* using closed-loop metrics. "C & L" indicates the use of both camera and LiDAR. The **best** and <u>second best</u> results are highlighted in **bold** and <u>underline</u>.

Method	Open-loop	Closed-loop	
	Average L2 Error ↓	Driving Score ↑	Success Rate (%) ↑
AD-MLP (Zhai et al. 2023)	3.64	18.05	0.00
VAD (Jiang et al. 2023)	0.91	42.35	15.00
Dual-AEB (Zhang et al. 2025b)	-	45.23	10.00
UniAD-Tiny (Hu et al. 2023)	0.80	40.73	13.18
UniAD-Base (Hu et al. 2023)	0.73	45.81	16.36
DriveTransformer (Jia et al. 2025)	<u>0.62</u>	63.46	**35.01**
TCP* (Wu et al. 2022)	1.70	40.70	15.00
TCP-ctrl* (Wu et al. 2022)	-	30.47	7.27
TCP-traj* (Wu et al. 2022)	1.70	59.90	30.00
ThinkTwice* (Jia et al. 2023b)	0.95	62.44	31.23
DriveAdapter* (Jia et al. 2023a)	1.01	<u>64.12</u>	33.08
VeteranAD (Ours)	**0.60**	**64.22**	<u>33.85</u>

Table 2: Performance comparison *on CARLA v2 using the Bench2Drive (Jia et al. 2024) benchmark* under both open-loop and closed-loop evaluations. "*" denotes expert feature distillation.

number of planning modes is set to 20. The balancing factors for loss calculation, λ_1, λ_2, λ_3, and λ_4, are all set to 10 for NAVSIM and 1 for Bench2Drive. The anchored trajectories are clustered using K-Means, following the same procedure as in previous works (Sun et al. 2025; Liao et al. 2025).

Comparison with State of the Art

As shown in Table 1, our VeteranAD is compared with state-of-the-art methods on the NAVSIM navtest split. Using the same ResNet-34 backbone, VeteranAD achieves a PDM Score (PDMS) of 90.2, significantly outperforming previous learning-based methods. With only camera input, VeteranAD surpasses UniAD by 6.8 PDMS, demonstrating its superior performance. Even compared to top-performing methods such as DiffusionDrive and WoTE, VeteranAD achieves higher scores across all evaluation metrics. These results highlight the effectiveness of our "perception-in-plan" design for end-to-end planning. We further evaluate our method on CARLA v2 using the Bench2Drive benchmark under both open-loop and closed-loop settings. As shown in Table 2, VeteranAD achieves an average L2 distance of 0.60 in the open-loop evaluation, outperforming all baselines. In the closed-loop evaluation, VeteranAD delivers competitive performance, on par with state-of-the-art methods such as DriveTransformer and DriveAdapter. These strong results demonstrate the effectiveness and generalization capability of our approach.

Ablation Study

Effects of components. Table 3 presents an ablation study of the Planning-Aware Holistic Perception module and the Localized Autoregressive Trajectory Planning module. The results include the performance of the full model as well as the effect of each module individually. In the first row, replacing position-guided attention with vanilla attention in the perception module leads to a drop in PDMS, highlighting the importance of using guiding points from anchored trajectories as planning priors. The second row shows that removing the guiding points and directly outputting planning trajectories—rather than predicting offsets—also results in significant performance degradation. These results suggest

Holistic perception	Trajectory planning	NC ↑	DAC ↑	TTC ↑	Comf. ↑	EP ↑	PDMS ↑
	✓	98.5	96.0	94.9	100	80.8	87.5
✓		98.6	96.7	95.2	100	81.9	88.4
✓	✓	99.1	98.3	96.1	100	83.1	90.2

Table 3: Ablation study on the components, including the Planning-Aware Holistic Perception module and the Localized Autoregressive Trajectory Planning module.

Img-attn	BEV-attn	Agent-attn	PDMS ↑
	✓	✓	89.7
✓		✓	89.1
✓	✓		89.4
✓	✓	✓	90.2

Table 4: Ablation study on cross-attention with image, BEV, and agent features.

Decoding type	DAC ↑	TTC ↑	EP ↑	PDMS ↑
NAR	97.1	94.9	82.1	88.6
AR	98.3	96.1	83.1	90.2

Table 5: Ablation study on the planning process in autoregressive (AR) and non-autoregressive (NAR) modes within the Localized Autoregressive Trajectory Planning module.

that the guiding points from anchored trajectories play a crucial role in accurate planning. When both modules are applied simultaneously, as shown in the third row, the PDMS reaches 90.2, demonstrating their complementary strengths and overall effectiveness.

Effects of different attention types. We investigate the impact of different attention types applied to image features, BEV features, and agent features, with the results summarized in Table 4. Removing any single attention mechanism leads to a performance drop, with the most significant decline observed when BEV attention is removed, highlighting the critical role of road information in planning. Each type of attention captures interactions with specific traffic elements, such as lanes, surrounding agents, and static obstacles. The combination of all three attention mechanisms yields the best performance, as shown in the last row.

AR $v.s.$ **NAR.** The autoregressive (AR) decoding process underpins the "perception-in-plan" framework by progressively predicting future trajectories while performing targeted perception at each step. We conduct an ablation study by replacing the AR approach with a non-autoregressive (NAR) strategy in the trajectory planning module, as shown in Table 5. In the NAR setting, trajectory queries interact with scene features simultaneously in a one-shot manner, guided by all points in the anchored trajectories—thus following the conventional "perception-planning" paradigm. The results show that the AR approach consistently outperforms the NAR counterpart. This is because, in the AR setting, trajectory queries focus on one trajectory point at a time, enabling step-wise adjustment and tighter coupling between perception and planning. In contrast, the NAR method processes all points in parallel, making perception less responsive to planning intent and resulting in suboptimal performance. This substantial improvement highlights the effectiveness of the AR-based, planning-oriented design in our framework.

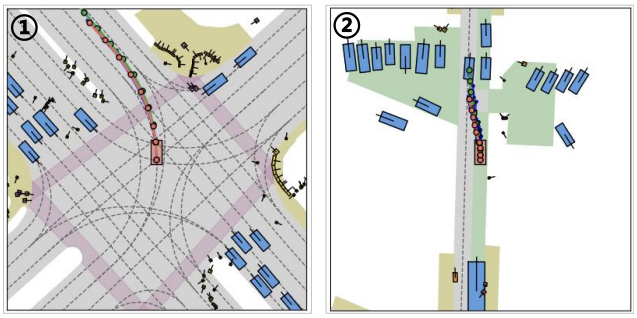

Figure 3: Qualitative results *on the NAVSIM dataset*. The ground-truth trajectory is shown in green, and the planned trajectory is shown in orange.

Comparison on nuScenes Dataset

To further validate the effectiveness and generalization of our VeteranAD, we conduct an open-loop planning experiment on the nuScenes (Caesar et al. 2020) dataset, with the results presented in Table 6. We integrate our design on top of VAD (Jiang et al. 2023) and follow its training and inference procedures. As shown, our approach reduces the average L2 Displacement Error by 0.10 m and lowers the average Collision Rate by 27.2% compared to VAD.

Efficiency Analysis

We compare our VeteranAD with the state-of-the-art method DiffusionDrive (Liao et al. 2025), following its official training and inference protocols. Our model requires approximately 8 hours to train, compared to 9 hours for DiffusionDrive. At inference time, VeteranAD achieves an average latency of 22.3 ms, while DiffusionDrive runs at 18.4 ms. Despite comparable efficiency in both training and inference, VeteranAD achieves significantly better performance. All experiments are conducted on NVIDIA GeForce RTX 3090 GPUs for fair comparison.

Method	L2 (m) ↓				Col. Rate (%) ↓				FPS
	$1s$	$2s$	$3s$	Avg.	$1s$	$2s$	$3s$	Avg.	
VAD (Jiang et al. 2023)	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22	4.5
VAD + VeteranAD	$0.33_{-0.08}$	$0.59_{-0.11}$	$0.94_{-0.11}$	$0.62_{-0.10}$	$0.03_{-0.04}$	$0.12_{-0.05}$	$0.34_{-0.07}$	$0.16_{-0.06}$	4.3

Table 6: Performance comparison of open-loop planning results *on the nuScenes (Caesar et al. 2020) validation dataset*.

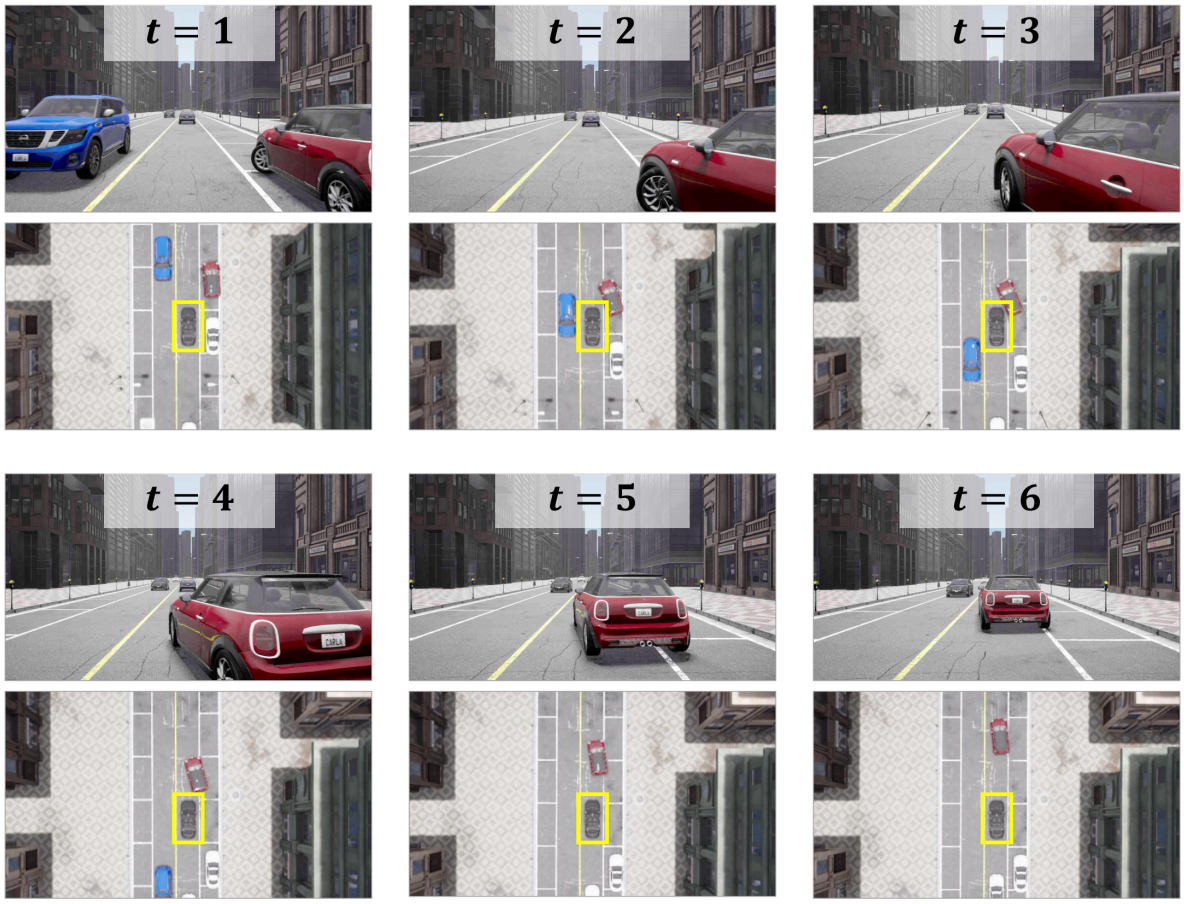

Figure 4: Qualitative results *on the Bench2Drive dataset*. The top image shows the front-view camera perspective, while the bottom image presents the BEV. In the BEV view, the ego vehicle is marked with a yellow box. In this scenario, our model successfully slows down and avoids a collision as a previously parked vehicle begins to merge into the road.

Qualitative Results

As shown in Figure 3, on the NAVSIM dataset, our model accurately plans complex maneuvers such as left turns and lane changes. As shown in Figure 4, during the closed-loop simulation on the Bench2Drive dataset, our model slows down to yield to a parked vehicle that begins merging into the road. Additional visualizations and failure cases are provided in the appendix.

Conclusion

In this work, we propose VeteranAD, an end-to-end autonomous driving framework built upon the proposed "perception-in-plan" paradigm, which tightly integrates perception into the planning process. By leveraging multi-mode anchored trajectories as planning priors, the Planning-Aware Holistic Perception module enables targeted and comprehensive understanding of the traffic scene, while the Localized Autoregressive Trajectory Planning module progressively refines future trajectories based on perception feedback. Extensive experiments on NAVSIM and Bench2Drive demonstrate that VeteranAD achieves state-of-the-art performance.

Limitations and future work. The limitations of our model stem from its restricted capacity for closed-loop simulation, a common challenge for imitation learning-based end-to-end approaches. In the future, incorporating reinforcement learning may help enhance planning performance.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (Grant No. 62376060).

References

Amini, A.; Gilitschenski, I.; Phillips, J.; Moseyko, J.; Banerjee, R.; Karaman, S.; and Rus, D. 2020. Learning robust control policies for end-to-end autonomous driving from data-driven simulation. *IEEE RA-L*.

Amini, A.; Wang, T.-H.; Gilitschenski, I.; Schwarting, W.; Liu, Z.; Han, S.; Karaman, S.; and Rus, D. 2022. Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles. In *ICRA*.

Bacha, A.; Bauman, C.; Faruque, R.; Fleming, M.; Terwelp, C.; Reinholtz, C.; Hong, D.; Wicks, A.; Alberi, T.; Anderson, D.; et al. 2008. Odin: Team victortango's entry in the darpa urban challenge. *Journal of field Robotics*.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*.

Chen, S.; Jiang, B.; Gao, H.; Liao, B.; Xu, Q.; Zhang, Q.; Huang, C.; Liu, W.; and Wang, X. 2024. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint*.

Cheng, J.; Chen, Y.; Mei, X.; Yang, B.; Li, B.; and Liu, M. 2024. Rethinking imitation-based planners for autonomous driving. In *ICRA*.

Chitta, K.; Prakash, A.; Jaeger, B.; Yu, Z.; Renz, K.; and Geiger, A. 2023. TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving. *IEEE TPAMI*.

Dauner, D.; Hallgarten, M.; Geiger, A.; and Chitta, K. 2023a. Parting with Misconceptions about Learning-based Vehicle Motion Planning. In *CoRL*.

Dauner, D.; Hallgarten, M.; Geiger, A.; and Chitta, K. 2023b. Parting with misconceptions about learning-based vehicle motion planning. In *CoRL*.

Dauner, D.; Hallgarten, M.; Li, T.; Weng, X.; Huang, Z.; Yang, Z.; Li, H.; Gilitschenski, I.; Ivanovic, B.; Pavone, M.; et al. 2024. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *NeurIPS*.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An open urban driving simulator. In *CoRL*.

Gulino, C.; Fu, J.; Luo, W.; Tucker, G.; Bronstein, E.; Lu, Y.; Harb, J.; Pan, X.; Wang, Y.; Chen, X.; et al. 2023. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. *NeurIPS*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hu, S.; Chen, L.; Wu, P.; Li, H.; Yan, J.; and Tao, D. 2022. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*.

Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; et al. 2023. Planning-oriented autonomous driving. In *CVPR*.

Jia, X.; Gao, Y.; Chen, L.; Yan, J.; Liu, P. L.; and Li, H. 2023a. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *ICCV*.

Jia, X.; Wu, P.; Chen, L.; Xie, J.; He, C.; Yan, J.; and Li, H. 2023b. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *CVPR*.

Jia, X.; Yang, Z.; Li, Q.; Zhang, Z.; and Yan, J. 2024. Bench2Drive: Towards Multi-Ability Benchmarking of Closed-Loop End-To-End Autonomous Driving. In *NeurIPS*.

Jia, X.; You, J.; Zhang, Z.; and Yan, J. 2025. DriveTransformer: Unified Transformer for Scalable End-to-End Autonomous Driving. In *ICLR*.

Jiang, B.; Chen, S.; Xu, Q.; Liao, B.; Chen, J.; Zhou, H.; Zhang, Q.; Liu, W.; Huang, C.; and Wang, X. 2023. Vad: Vectorized scene representation for efficient autonomous driving. In *ICCV*.

Karnchanachari, N.; Geromichalos, D.; Tan, K. S.; Li, N.; Eriksen, C.; Yaghoubi, S.; Mehdipour, N.; Bernasconi, G.; Fong, W. K.; Guo, Y.; et al. 2024. Towards learning-based planning: The nuPlan benchmark for real-world autonomous driving. In *ICRA*.

Li, K.; Li, Z.; Lan, S.; Liu, J.; Xie, Y.; zhizhong zhang; Wu, Z.; Yu, Z.; and Alvarez, J. M. 2024a. Hydra-MDP++: Advancing End-to-End Driving via Hydra-Distillation with Expert-Guided Decision Analysis. https://openreview.net/forum?id=5xfAcRHfgP.

Li, Q.; Peng, Z.; Feng, L.; Zhang, Q.; Xue, Z.; and Zhou, B. 2022a. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE TPAMI*.

Li, Y.; Wang, Y.; Liu, Y.; He, J.; Fan, L.; and Zhang, Z. 2025. End-to-End Driving with Online Trajectory Evaluation via BEV World Model. In *ICCV*.

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

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; and Dai, J. 2022b. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*.

Li, Z.; Yu, Z.; Lan, S.; Li, J.; Kautz, J.; Lu, T.; and Alvarez, J. M. 2024c. Is ego status all you need for open-loop end-to-end autonomous driving? In *CVPR*.

Liao, B.; Chen, S.; Wang, X.; Cheng, T.; Zhang, Q.; Liu, W.; and Huang, C. 2023. Maptr: Structured modeling and learning for online vectorized hd map construction. In *ICLR*.

Liao, B.; Chen, S.; Yin, H.; Jiang, B.; Wang, C.; Yan, S.; Zhang, X.; Li, X.; Zhang, Y.; Zhang, Q.; and Wang, X. 2025. DiffusionDrive: Truncated Diffusion Model for End-to-End Autonomous Driving. In *CVPR*.

Liu, H.; Teng, Y.; Lu, T.; Wang, H.; and Wang, L. 2023. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *ICCV*.

Ljungbergh, W.; Tonderski, A.; Johnander, J.; Caesar, H.; Åström, K.; Felsberg, M.; and Petersson, C. 2024. NeuroNCAP: Photorealistic Closed-loop Safety Testing for Autonomous Driving. In *ECCV*.

Philion, J.; and Fidler, S. 2020. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*.

Prakash, A.; Chitta, K.; and Geiger, A. 2021. Multi-Modal Fusion Transformer for End-to-End Autonomous Driving. In *CVPR*.

Shi, S.; Jiang, L.; Dai, D.; and Schiele, B. 2022. Motion transformer with global intention localization and local movement refinement. In *NeurIPS*.

Song, N.; Zhang, B.; Zhu, X.; and Zhang, L. 2024. Motion Forecasting in Continuous Driving. In *NeurIPS*.

Sun, W.; Lin, X.; Shi, Y.; Zhang, C.; Wu, H.; and Zheng, S. 2025. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *ICRA*.

Thrun, S.; Montemerlo, M.; Dahlkamp, H.; Stavens, D.; Aron, A.; Diebel, J.; Fong, P.; Gale, J.; Halpenny, M.; Hoffmann, G.; et al. 2006. Stanley: The robot that won the DARPA Grand Challenge. *Journal of field Robotics*.

Treiber, M.; Hennecke, A.; and Helbing, D. 2000. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*.

Vaswani, A. 2017. Attention is all you need. In *NeurIPS*.

Wang, S.; Liu, Y.; Wang, T.; Li, Y.; and Zhang, X. 2023. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *ICCV*.

Wang, T.-H.; Amini, A.; Schwarting, W.; Gilitschenski, I.; Karaman, S.; and Rus, D. 2022a. Learning interactive driving policies via data-driven simulation. In *ICRA*.

Wang, Y.; Guizilini, V. C.; Zhang, T.; Wang, Y.; Zhao, H.; and Solomon, J. 2022b. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*.

Weng, X.; Ivanovic, B.; Wang, Y.; Wang, Y.; and Pavone, M. 2024. Para-drive: Parallelized architecture for real-time autonomous driving. In *CVPR*.

Wu, P.; Jia, X.; Chen, L.; Yan, J.; Li, H.; and Qiao, Y. 2022. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *NeurIPS*.

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

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

Zhang, B.; Song, N.; Jin, X.; and Zhang, L. 2025a. Bridging Past and Future: End-to-End Autonomous Driving with Historical Prediction and Planning. In *CVPR*.

Zhang, L.; Li, P.; Liu, S.; and Shen, S. 2024. Simpl: A simple and efficient multi-agent motion prediction baseline for autonomous driving. *IEEE RA-L*.

Zhang, W.; Li, P.; Wang, J.; Sun, B.; Jin, Q.; Bao, G.; Rui, S.; Yu, Y.; Ding, W.; Li, P.; et al. 2025b. Dual-AEB: Synergizing Rule-Based and Multimodal Large Language Models for Effective Emergency Braking. In *ICRA*.

Zheng, W.; Song, R.; Guo, X.; Zhang, C.; and Chen, L. 2024. Genad: Generative end-to-end autonomous driving. In *ECCV*.

Zhou, Z.; Wang, J.; Li, Y.-H.; and Huang, Y.-K. 2023. Query-centric trajectory prediction. In *CVPR*.