OCCDRIVER: FUTURE OCCUPANCY GUIDED DUAL-BRANCH TRAJECTORY PLANNER IN AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Trajectory planning for autonomous driving is challenging due to agents' behavioral uncertainty and intricate multi-agent interaction modeling. Most existing studies generate trajectories without explicitly exploiting possible scene evolution, while world models predict consequences from ego behavior, enabling more informed planning decisions. Inspired by the world model, we propose OccDriver, a novel rasterized-to-vectorized dual-branch framework for trajectory planning. This pipeline performs a coarse-to-fine trajectory decoding process: The vectorized branch first generate multimodal coarse trajectories; Then the rasterized branch predicts future scene evolutions conditioned on each coarse trajectory via occupancy flow prediction; Lastly, the vectorized branch leverages intuitive future interaction evolution of each modality from the rasterized branch and produces refined trajectories. Several cross-modality (occupancy and trajectory) losses are further introduced to improve the consistency between trajectory and occupancy prediction. Additionally, we apply a contingency objective in both occupancy space, considering marginal and joint occupancy distributions in different planning scopes. Our model is assessed on the large-scale real-world nuPlan dataset and its associated planning benchmark. Experiments show that OccDriver achieves state-of-the-art in both Non-Reactive and Reactive closed-loop performance.

1 INTRODUCTION

Trajectory planning for autonomous driving confronts intrinsic challenges due to the complexity of multi-agent interactions and pervasive uncertainty Djuric et al. (2020); Xu et al. (2014)—from sensor noise to high-level behavioral unpredictability. Deep learning methods Bansal et al. (2018); Guo et al. (2023); Scheel et al. (2022) have emerged as a promising alternative to rule-based systems Bouchard et al. (2022); Treiber et al. (2000); Yi et al. (2018). However, effectively modeling the dynamic and uncertain interplay among agents in future scenarios remains a formidable task. In this work, we tackle this problem from two key perspectives: predictive modeling of future interactions and the choice of its representation space.

Representation space is crucial in planning, transforming raw sensor inputs into structured features that capture environmental contexts. Figure. 1 presents a comparison of different representation paradigms. Rasterized representations Hu et al. (2021); Kim et al. (2022); Hu et al. (2023b), by modeling occupancy over spatiotemporal grids, offer robustness against occlusions and a probabilistic view of joint future states Liu et al. (2023a); Mahjourian et al. (2022). Unfortunately, this approach incurs discretization artifacts and loses fine individual context and geometric details. In contrast, vectorized methods Jiang et al. (2023); Zhou et al. (2022; 2023) provide high-precision trajectory generation by capturing detailed individual semantics but tend to oversimplify evolving future interactions and require substantial manual feature engineering to approximate uncertainty Khaitan et al. (2021); Chen et al. (2024). To overcome these limitations, we propose a hybrid approach that retains the probabilistic strengths of rasterized joint modeling while preserving the individual fidelity of vectorized representations, supporting probabilistic interaction modeling and more interaction-informed trajectory planning.

The problem of interaction modeling has been extensively studied in recent years. Multi-agent reinforcement learning methods Kiran et al. (2021); Liu et al. (2022) are developed to simulate

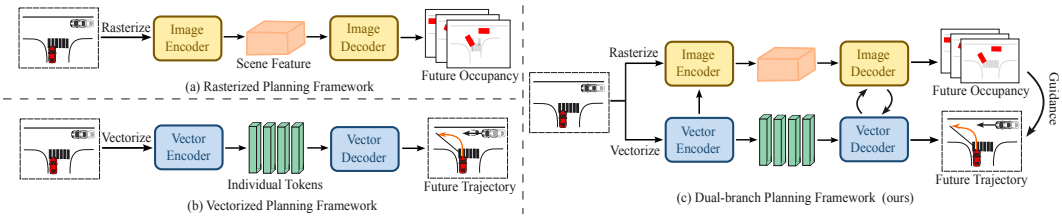


Figure 1: Illustration of different representation paradigms: (a) rasterized framework models the scene-level joint dynamics in spatiotemporal grids; (b) vectorized framework performs individual-level trajectory planning; (c) our proposed dual-branch framework integrates scene-level and individual-level information, further leveraging future scene as planning guidance.

interactions and learn policies through trial and error. However, they often struggle with scalability and environmental non-stationarity. Graph neural networks (GNNs) Mo et al. (2022); Sheng et al. (2022) excel in capturing relational dependencies, yet suffer from message-passing limitations and escalating computational costs as the number of agents increases. Beyond encoding mechanisms, decoding mechanisms e.g., game theoretic approaches Huang et al. (2023); Wang et al. (2021) and tree policy planning Huang et al. (2024) have been explored. However, these methods typically lack robustness, limiting practicality for real-world deployment. Nevertheless, all these methods perform forward-only planning without correction ability when poor rollout occurs, often necessitating a strong trajectory scoring module. Our method first plans multimodal coarse ego trajectories in vectorized branch. Then the rasterized branch constructs probabilistic occupancy maps conditioned on each coarse trajectory, capturing future scene evolution resulting from each ego action. Lastly, the interactive knowledge embedded in the occupancy space is distilled into the vectorized branch for trajectory guidance.

Beyond the framework, we propose a suite of specialized loss functions, leveraging future occupancy as planning guidance. Occupancy interference loss captures ego and agents' exclusivity in occupancy space, which can be seen as ego planning in occupancy space. Occupancy guidance loss enforces explicit consistency between trajectories and predicted occupancy, which bridges the gap between scene's occupancy probabilistic modeling and trajectory planning, ensuring effective information transfer between the two branches.

Our framework also integrates contingency planning objective Cui et al. (2021); Li et al. (2023b); Liu et al. (2024a) utilizing both marginal and joint occupancy distribution to balance safety and efficiency in dynamic traffic environments. In short-term, we estimate marginal occupancy probabilities of key interactive agents, enabling the ego vehicle to respond swiftly to imminent risks. For long-term planning, we estimate joint occupancy probabilities to construct a modality-consistent traffic evolution, ensuring scene-compliance and avoiding over-conservative behavior.

Our contributions are summarized as follows:

1) We propose a dual-branch transformer framework for coarse-to-fine trajectory planning, where a rasterized branch serves as a 2D occupancy world model by predicting future scene evolution conditioned on coarse trajectories and guiding fine-grained trajectory planning.

2) We introduce several specialized losses to facilitate effective information transfer between vectorized and rasterized branches, imposing intuitive guidance on planning.

3) We incorporate a contingency planning strategy that leverages short-term marginal occupancy distribution for risk-sensitive planning and long-term joint occupancy distribution for scene-compliant behavior generation.

4) Extensive evaluation on the NuPlan dataset, achieving state-of-the-art planning performance on both reactive and non-reactive closed-loop metrics.

2 Related Work

2.1 Representation Space in Motion Planning

Imitation-based planning method Hu et al. (2022a); Cheng et al. (2024b) has attracted lots of research interest due to the accessibility of massive real-world expert driving data Caesar et al. (2020; 2021);

Ettinger et al. (2021). It can be categorized into two branches according to the representation paradigm: rasterized and vectorized approaches.

Rasterized Approaches project scene context into discret bird-eye-view(BEV) images Hu et al. (2022b); Li et al. (2024c) and encode it with off-the-shelf image feature extraction methods Liu et al. (2021b); Dosovitskiy et al. (2020). Earlier research uses CNN Renz et al. (2022); Song et al. (2020) for feature encoding and trajectory decoding, while recent research utilizes transformer structure Chen et al. (2021); Huang et al. (2022); Chitta et al. (2022); Zhang et al. (2022); Yao et al. (2023) for better performance. Some works decode agents' future movement as future occupancy and flow fields Liu et al. (2023a); Kim et al. (2022), providing a dense and intuitive scene representation, yet compromising fewer individual details and limited receptive field.

Vectorized Approaches yield impressive performance because of vector's concise but effective representation capacity for scene semantics. Building on vectorized representation of traffic scenes Gao et al. (2020), researchers have used structures like graph neural networks for interaction modeling and DETR-style transformers Wang et al. (2023); Achaji et al. (2022) for query-based decoding. Additionally, instead of fully learnable queries, anchor-based Afshar et al. (2024); Li et al. (2023a); Chen et al. (2024) queries are used to decode multi-modal trajectories with explicit patterns.

OccDriver combines the advantages of vectorized and rasterized representations. The rasterized branch predicts future occupancy and flow fields to represent future scene evolution, serving as guidance for trajectory planning. The comparison between our work and other occupancy-assisted planning methods is provided in Appendix A.2.

2.2 Contingency Planning

Contingency planning Li et al. (2023c) is traditionally framed as a tree-structured trajectory optimization problem, where each branch represents a possible scenario and a shared short-term segment ensures safety across all cases. While theoretically complete, this approach suffers from exponential complexity, as branches grow combinatorially with interactive agents' intentions, requiring a complicated safety-evaluation pruning strategy. Also, each agent's intention is reduced to a deterministic approximation, resulting in the loss of trajectory-level multi-modality.

OccDriver addresses these challenges by formulating contingency planning within a dense probabilistic occupancy space. Rather than explicitly constructing and pruning a scenario tree, it directly predicts multiple scene-level rollouts of interactive agents' joint occupancy distribution, which also mitigates the limitations of deterministic scenario approximation.

3 Method

The overall framework of OccDriver is illustrated in Fig. 2. We first introduce problem formulation in Sec .3.1. Then, we demonstrate OccDriver's dual-branch network architecture in Sec. 3.2. Marginal occupancy distribution prediction is presented in Sec. 3.3. Finally, in Sec. 3.4, we propose several specialized training losses.

3.1 Problem Formulation

Our research is dedicated to the task of trajectory planning. Our model input $\mathbf{X}$ composes of states of ego vehicle $\mathbf{E}$ and dynamic agents $\mathbf{A}$ over historical horizon T_h, states of static objects $\mathbf{S}$ and a High-Definition map $\mathbf{M}$. The objective is to plan ego vehicle's M-modal future states $\mathbf{Y} = \{(y_i, \pi_i) \mid i = 1...M\}$, where y is trajectory over future horizon T_f and π is the confidence score. With the integration of the occupancy branch, our model is updated with rasterized inputs and outputs. $\mathbf{E}^0$, $\mathbf{A}^0$ and $\mathbf{M}$ at current step are projected into occupancy grid $\mathbf{O}_e^0$, $\mathbf{O}_a^0$ and $\mathbf{O}_m$. Following the practice in Liu et al. (2023a), current backward flow $\mathbf{FL}^0$ is computed as extra input. Occupancy prediction branch takes $\mathbf{X}_{occ} = \{\mathbf{O}_e^0, \mathbf{O}_a^0, \mathbf{O}_m, \mathbf{FL}^0\}$ as input and predicts ego vehicle's occupancy $\mathbf{O}_e$, other agents' occupancy $\mathbf{O}_a$ and scene's backward flow $\mathbf{FL}$ over future horizon T_f. In alignment with trajectory planning, occupancy branch outputs multimodal prediction $\mathbf{Y}_{occ} = \{(\mathbf{O}_{e,i}, \mathbf{O}_{a,i}, \mathbf{FL}_i) \mid i = 1...M\}$. Consequently, the formulation of our model is given as:

$$\mathbf{Y}, \mathbf{Y}_{occ} = f(\mathbf{X}, \mathbf{X}_{occ} \,|\theta), \tag{1}$$

where f denotes the neural network of OccDriver, θ is the model parameters.

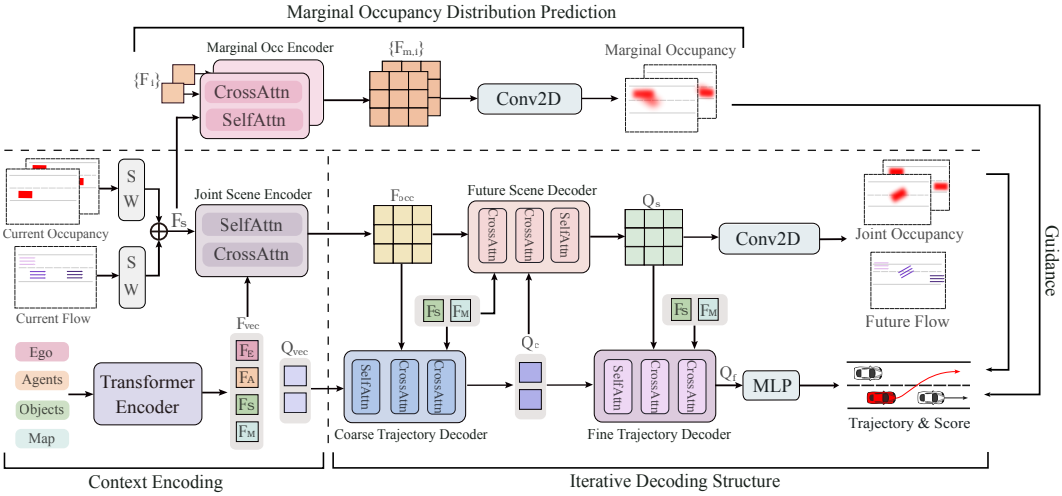


Figure 2: **The architecture of the OccDriver** comprises three fundamentals. Context Encoding first encodes heterogeneous inputs into vectorized individual features F_{vec} and joint scene feature F_{occ} respectively. Initialized by Q_{vec} and F_{occ}, dual-branch iterative decoding structure decodes joint future occupancy and trajectory iteratively. Short-term marginal occupancy is generated via marginal occupancy distribution prediction. Joint and marginal occupancy predictions enforces explicit guidance to trajectory through specialized losses.

3.2 Dual-branch Architecture

Context Encoding. For vectorized branch, heterogeneous inputs $\{\mathbf{E}, \mathbf{A}, \mathbf{S}, \mathbf{M}\}$ are encoded as individual features $\{F_E, F_A, F_S, F_M\}$ with separate encoders. After added by positional embedding, encoded features are concatenated as $F_{vec} \in \mathbb{R}^{(1+N_A+N_S+N_M)\times D}$. We then perform scene-level feature fusion via a transformer encoder, employing self-attention to capturing social interactions between encoded scene semantics.

For the rasterized occupancy branch, $\mathbf{O}^0 = \{\mathbf{O}_e^0, \mathbf{O}_a^0, \mathbf{O}_m\}$ and $\mathbf{FL}^0$ are embedded into occupancy feature F_o and flow feature F_f with separate Swin-Transformer blocks. F_o and F_f are then concatenated and projected into the scene feature map $F_s \in \mathbb{R}^{(H/4)\times(W/4)\times D}$ with an MLP. Then we propose a two-layer attention-based joint scene encoder for better aggregation of scene information. Cross-attention is employed to integrate coarse trajectory feature F_{vec}'s dense semantic information, followed by a self-attention for feature fusion in the occupancy space. The process of joint scene encoder is described as:

$$F_s = \mathrm{CrossAttn}(F_s, F_{vec}, F_{vec}), \tag{2}$$
$$F_{occ} = \mathrm{SelfAttn}(F_s), \tag{3}$$

where $\mathrm{CrossAttn}(Q, K, V)$ and $\mathrm{SelfAttn}(X)$ indicates cross-attention and self-attention mechanism respectively. The encoded occupancy feature is denoted as F_{occ}.

Iterative Decoding Structure. Retaining encoded vectorized feature F_{vec} and occupancy feature F_{occ}, we decode future occupancy evolution and future trajectories iteratively. As depicted in Fig. 2, iterative decoding structure is composed of 3 sequential decoders: coarse trajectory decoder D_c, future scene decoder D_s and fine trajectory decoder D_f. For better consistency in scene evolution, we jointly decode the future trajectories of ego vehicle and agents. A set of M learnable queries $Q_m \in \mathbb{R}^{M\times D}$ is combined with $\{F_E, F_A\} \subset F_{vec}$, forming multi-modal decoding queries $Q_{vec} \in \mathbb{R}^{M\times(N_A+1)\times D}$.

Coarse trajectory decoder D_c takes Q_{vec} as input, deriving coarse trajectory queries Q_c. Each layer of D_c comprises of three types of attention mechanisms. Within each modality, Q_{vec} first performs self-attention to extract social interactions among agents. Then cross attention is employed to integrate static obstacle and map information from $\{F_S, F_M\} \subset F_{vec}$. Lastly, Q_{vec} queries F_{occ} via another cross-attention, establishing a spatial scene understanding.

In occupancy branch, future scene decoder D_s acts as a world model in BEV view, decoding F_{occ} into feature Q_s that represent future scene evolution conditioned on ego coarse trajectories. In each layer of D_s, F_{occ} conducts two cross-attention operations to query Q_c and $\{F_S, F_M\}$ respectively, aggregating instance-level features and map information. F_{occ} further applies self-attention for social interaction modeling. The decoded Q_s encompasses an intuitive prior of the future scene evolution.

Despite aggregating current scene information, Q_c requires further refinement under the guidance of future scene evolution. Thus, utilizing future scene information from Q_s, fine trajectory decoder D_f refines Q_c into future-informed trajectory query Q_f to make scene-consistent planning. D_f and D_c share the same architecture, except that D_f leverages Q_s instead of F_{occ} as Key and Value in the third cross-attention. The iterative decoding process is formulated below:

$$Q_c = D_c(\mathrm{Q} = Q_{vec},\ \mathrm{K,V} = F_{occ}, \{F_S, F_M\}), \tag{4}$$

$$Q_s = D_s(\mathrm{Q} = F_{occ},\ \mathrm{K,V} = Q_c, \{F_S, F_M\}), \tag{5}$$

$$Q_f = D_f(\mathrm{Q} = Q_c,\ \mathrm{K,V} = Q_s, \{F_S, F_M\}). \tag{6}$$

Prediction heads. Given Q_f and Q_s, different prediction heads are implemented. Fine trajectory feature Q_f determines M-modal joint planning trajectories $\mathbf{Y}$ with corresponding confidence scores π via two MLPs. To generate reasonable raw behaviors, coarse trajectory feature Q_c is decoded into coarse trajectory $\mathbf{Y_c}$ via another MLP. Future scene feature Q_s is upsampled back to original shape of the input images and then deployed with two separate 2d-CNNs to output M-modal future occupancy fields and flow fields. Different from earlier works, we decouple the future occupancy prediction into the ego vehicle $\mathbf{O}_e$ and surrounding vehicles $\mathbf{O}_a$ to facilitate explicit loss guidance detailed in Sec. 3.4. We provide further architectural details in Appendix. C.

3.3 Marginal Occupancy Distribution Prediction

Accounting for emergency risk, we model potential agent behaviors by extending occupancy prediction beyond agents' joint distribution to individual's short-term marginal distribution. As shown in the top of Fig. 2, the scene feature map F_s integrate individual agent features $F_i \in F_A$ through an additional marginal occupancy encoder. F_s attends only to the vectorized feature of a single agent, rather than all elements in F_{vec}, thereby capturing the agent's marginal behavioral feature $F_{m,i}$. Without modality decomposition, $F_{m,i}$ is directly leveraged to predict i-th agent's short-term marginal occupancy $\mathbf{O}_{m,i}$ of T_s horizon ($T_s < T_f$) through upsampling and a 2D-CNN output head.

We perform agent pruning to reduce the computational cost, retaining only interactive agents for marginal occupancy distribution prediction. We devise a rule-based pruning method with minor inductive bias by selecting agents whose future bounding boxes cross with ego's future path. Please refer to Appendix. D for details about agent pruning. Marginal occupancy prediction is applied only in the training phase, enabling the model to learn individual agent's behavior patterns and short-term uncertainties during the optimization process, which also serves as a foundation for contingency planning. Its entire procedure is presented as follows:

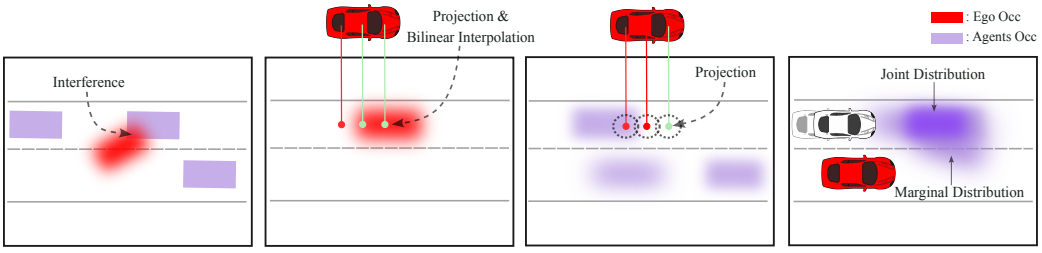
$$\begin{aligned} \{F_i \,|i = 1, 2, ..., N_m\} &= \mathrm{Prune}(F_A), \\ F_{m,i} &= f_m(F_s, F_i), \\ \mathbf{O}_{m,i} &= \mathrm{Conv}(\mathrm{Upsample}(F_{m,i})). \end{aligned} \tag{7}$$

where N_m indicates the number of pruned interactive agents, while f_m denotes the network of marginal occupancy encoder, sharing the same structure as joint scene encoder.

3.4 Training Loss

Apart from basic trajectory and occupancy flow supervision, we devise a suite of specialized losses as shown in (a)-(c) in Fig. 3. Occupancy interference loss is applied to strengthen the interaction awareness between predicted ego and agents' joint occupancy. Occupancy guidance loss is further introduced to guide the trajectories with future occupancy probabilistic distribution. Utilizing predicted marginal and joint occupancy distribution, we apply contingency planning strategy for better driving safety without compromising efficiency. We leave details of trajectory planning loss $\mathcal{L}_{traj}$ and occupancy prediction loss $\mathcal{L}_{occ}$ in Appendix. E.

Occupancy interference loss. The occupied areas of ego vehicle and agents are mutually exclusive, inherently reflecting future interactions. We incorporate this property within occupancy interference loss, which is formulated as Eq. 8. For training stability, we apply teacher-forcing Williams & Zipser (1989) technique. For both ego and agents, the loss is calculated as the average predicted occupancy within the opponents' GT occupied regions, measuring the extent of spatial interference. Optimizing occupancy interference loss effectively enhances the interaction awareness in the occupancy space,



(a) Occupancy Interference Loss (b) Traj-Occ Alignment Loss (c) Traj-Occ Collision Losss (d) Contigency Planning

Figure 3: (a) - (c) illustrates the proposed losses: (a) penalizes the overlapping region between occupancy prediction and its opponent's ground truth; (b) is imposed if ego's trajectory point is mapped onto its low-occupancy region; (c) is applied when the distance between ego's mapped point and agents' high-occupancy region is smaller than the safety margin. (b) and (c) constitute occupancy guidance loss. (d) illustrates our contingency policy in occupancy space. Short-term occupancy takes the maximum value of joint and marginal distribution, considering potential behavior uncertainty.

facilitating subsequent occupancy guidance loss as well.

$$\mathcal{L}_{oe} = \text{sum}(\mathbf{O}_e^* \cdot \mathbf{O}_a^{gt}) \,/\, \text{sum}(\mathbf{O}_a^{gt}), \;\; \mathcal{L}_{oa} = \text{sum}(\mathbf{O}_a^* \cdot \mathbf{O}_e^{gt}) \,/\, \text{sum}(\mathbf{O}_e^{gt}),$$
$$\mathcal{L}_{oi} = \mathcal{L}_{oe} + \mathcal{L}_{oa}, \tag{8}$$

where $\mathbf{O}^*$ denotes occupancy prediction of best modality and $\mathbf{O}^{gt}$ denotes ground truth occupancy.

Occupancy guidance loss. Future occupancy can serve as spatial prior information in the BEV view, explicitly guiding ego's trajectory planning. Utilizing both $\mathbf{O}_a{}^*$ and $\mathbf{O}_e^*$, we devise trajectory-occupancy collision loss to steer the ego trajectory away from agents' high-probability occupied area and trajectory-occupancy alignment loss is introduced to constrain ego trajectory within its own high-probability occupied area.

Considering ego vehicle's shape, at each timestep t, we offset trajectory position (x_t, y_t) into N_v circle centers $\{P_i^t \mid i = 1...N_v\}$ to approximate the ego vehicle. For trajectory-occupancy alignment loss, we obtain P_i^t's occupied probability O_i^t on the predicted ego occupancy grid through coordinate projection and bilinear interpolation. To enforce alignment between trajectory and high-occupancy areas, penalty is applied to points whose occupancy probability is below the predefined threshold $\boldsymbol{\varepsilon}$.

$$\mathcal{L}_{align} = \frac{1}{T_f} \sum_{t=1}^{T_f} \sum_{i=1}^{N_v} \max(0, \boldsymbol{\varepsilon} - O_i^t). \tag{9}$$

For trajectory-occupancy collision loss, after mapping P_i^t onto the predicted agents' occupancy grid through coordinate projection, we compute its minimum Euclidean distance d_i^t to high-occupancy regions (where occupancy probability exceeds threshold $\boldsymbol{\zeta}$). Collision penalty is applied when d_i^t is below the safety margin $\boldsymbol{\eta}$, which indicates high collision risk.

$$\mathcal{L}_{collision} = \frac{1}{T_f} \sum_{t=1}^{T_f} \sum_{i=1}^{N_v} \max(0, d_i^t - \boldsymbol{\eta}). \tag{10}$$

Contingency Planning is incorporated into $\mathcal{L}_{collision}$ to enhance planning safety. Conventional contingency planning divides trajectory into short-term safe maneuver and subsequent branched long-term behavior sets. Compared to trajectory, occupancy probability models behavioral uncertainty more effectively. In our work, the predicted short-term marginal occupancy $\mathbf{O}_{m,i}$ is leveraged to represent the uncertainty of single agent's short-term behavior. As shown in Eq. 11, before computing $\mathcal{L}_{collision}$, the predicted all-agents' occupancy $\mathbf{O}_a^*$ incorporates $\{\mathbf{O}_{m,i} \mid i = 1...N_m\}$ through an element-wise maximum operation over a short period T_s. Thus, $\mathcal{L}_{collision}$ enforces ego trajectory to account for short-term risks caused by agents' behavior uncertainty, while keeping modality-compliant planning in the long term.

$$\tilde{\mathbf{O}}_a^* = \begin{cases} \max(O_a^t{}^*, \max_{i=1}^{N_m}(O_{m,i}^t)), \; t \le T_s \\ O_a^t{}^*, \; t > T_s \end{cases} \tag{11}$$

Occupancy guidance loss $\mathcal{L}_{og}$ is represented as a weighted sum of $\mathcal{L}_{align}$ and $\mathcal{L}_{inter}$, regulating ego's trajectory with the explicit guidance of predicted future scenario:

$$\mathcal{L}_{og} = w_1 \mathcal{L}_{align} + w_2 \mathcal{L}_{collision}. \tag{12}$$

Table 1: Performance comparison of closed-loop planning on nuPlan **Val14** benchmark. All Metrics are higher the better. Among learning-based methods, OccDriver achieves SOTA in both none-reactive score (NR-S) and reactive score (R-S) with top safety performance (Collisions and TTC).

Type	Planner	*Val14*					
		Collisions	TTC	Comfort	Progress	NR-S	R-S
Expert	Log-Replay	0.988	0.944	0.993	0.990	0.937	0.812
Learning	PDM-Open Dauner et al. (2023)	0.745	0.691	0.995	0.699	0.502	0.548
	RasterModel Caesar et al. (2021)	0.870	0.815	0.815	0.806	0.669	0.647
	UrbanDriver Scheel et al. (2022)	0.856	0.803	**1.000**	0.808	0.677	0.649
	PlanTF Cheng et al. (2024b)	0.941	0.907	0.937	**0.898**	0.853	0.771
	PLUTO Cheng et al. (2024a)	0.962	0.933	0.964	0.896	0.890	0.800
	BeTopNet Liu et al. (2024a)	0.966	0.916	0.932	0.866	0.883	0.837
	DiffusionPlanner Zheng et al. (2025a)	-	-	-	-	**0.899**	0.828
	OccDriver (Ours)	**0.971**	**0.938**	0.969	0.885	0.896	**0.838**

Table 2: Performance comparison of closed-loop planning on nuPlan $\mathbf{Test14-Hard}$ benchmark. All Metrics are higher the better. Compared to vectorized-only, topology-guided and diffusion-based methods, OccDriver achieves top driving scores with desirable planning safety and progress.

Planner	Inference Time (ms)	*Test14-Hard*			
		Collisions	Progress	NR-S	R-S
PLUTO Cheng et al. (2024a)	7.39	0.938	0.816	0.787	0.753
BeTopNet Liu et al. (2024a)	70.00	**0.968**	0.747	0.771	0.688
DiffusionPlanner Zheng et al. (2025a)	40.00	-	-	0.760	0.692
OccDriver (Ours)	23.03	0.941	**0.829**	**0.794**	**0.759**

All losses are differentiable, allowing for end-to-end training. The overall training loss comprises trajectory planning loss $\mathcal{L}_{traj}$, occupancy prediction loss $\mathcal{L}_{occ}$, occupancy interference loss $\mathcal{L}_{oi}$ and occupancy guidance loss $\mathcal{L}_{og}$. It is formulated as :

$$\mathcal{L} = \mathcal{L}_{traj} + \mathcal{L}_{occ} + \mathcal{L}_{oi} + \mathcal{L}_{og}. \tag{13}$$

4 Experiments

4.1 Benchmark and Metrics

OccDriver is trained and evaluated on nuPlan dataset Caesar et al. (2021). We use a standardized training set of 1M frames with 2s history and 8s horizon. Our evaluation is conducted on **Val14** split Dauner et al. (2023), and verified using closed-loop evaluation metrics: Non-Reactive Closed-Loop Simulation(CLS-NR) score and Reactive Closed-Loop(CLS-R) provided by nuPlan simulator. To verify performance under challenging cases, we further evaluate OccDriver on $\mathbf{Test14-Hard}$ split Cheng et al. (2024c). We follow the nuPlan challenge framework and report the official Planning Scores. More experiment details and parameter settings are in Appendix. F.

4.2 Main Results

Comparison with State of the Art. We conduct a comparative analysis between OccDriver and existing state-of-the-art learning-based methods on nuPlan **Val14** benchmark. All methods are evaluated without post-processing to compare models' planning performance. Comparative results are presented in Table. 1. In closed-loop simulation, OccDriver gains SOTA planning scores of 0.896 NR-Score and 0.838 R-Score. Notably, it demonstrates best performance on safety metircs, boosting to 0.971 (Collisions) and 0.938 (TTC). This can be attributed to guidance of future occupancy and contingency planning strategy, improving ego's awareness of spatial interactions and behavioral uncertainty. Unlike previous methods, OccDriver enhances safety performance with less degradation on Comfort and Progress. This results from the iterative decoding process, which refines trajectory holistically and generates scenario-compliant behavior.

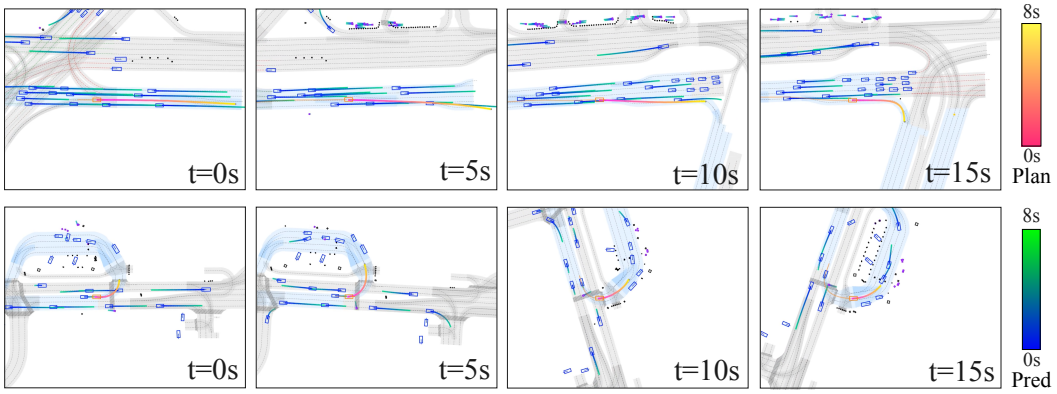
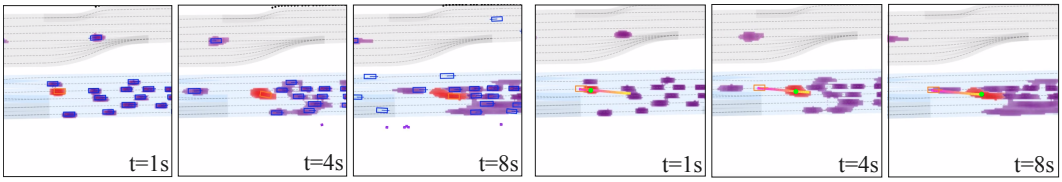


Figure 4: Qualitative results of closed-loop planning. Each scenario lasts 15 seconds. OccDriver performs interaction-compliant planning in changing lane in dense traffic (first row) and turning left after yielding (second row).



(a) Comparison between occupancy prediction and ground truth (b) Guidance from future occupancy to trajectory

Figure 5: Visualization of future occupancy prediction and guidance. (a) ego's (red) and agents' (purple) occupancy predictions coincide with their GT bounding boxes; (b) planning trajectory (green point) aligns with ego's occupancy while keeps away from agents' occupancy.

To further demonstrate the advantages of OccDriver under challenging scenarios, we conduct comparison with vectorized-only method (PLUTO), topology-guided method (BeTopNet) and diffusion-based method (Diffusion Planner) on $\mathbf{Test14-Hard}$ benchmark, as presented in Table. 2. OccDriver reaches the highest driving scores in both non-reactive and reactive simulation. Compared to PLUTO, OccDriver achieves enhancement in both safety and progress metics, demonstrating the effectiveness of incorporating occupancy branch. OccDriver also outperforms BeTopNet with $+3.0\%$ NR-S and $+10.3\%$ R-S, which models multi-agent interactions with topology connections. Despite excelling in safety metrics, BeTopNet suffers from a substantial degradation in progress. In contrast, our method maintains a favorable trade-off between safety and progress, leading to the best driving scores. This suggests that explicit spatial occupancy delivers more intuitive and fine-grained interaction relations than implicit topological structures, avoiding over-conservative planning. Results in Fig. 4 further prove our model's robust performance in tackling dense traffic scenarios and multi-agent interactive scenarios. Notably, compared to denoising paradigm of diffusion and the intricate topological modeling, OccDriver has less inference latency, enabling deployment in real-world applications.

Qualitative results. To visually demonstrate the effectiveness of future occupancy guidance, we render both the predicted future occupancy and ego's planning trajectory simultaneously. As depicted in Fig. 5, predicted occupancy overlaps with ground-truth bounding boxes, demonstrating the accuracy of future occupancy prediction and robustness of scene evolution reasoning. Besides, planning trajectory is positioned within the ego's high-occupancy region and steers away from agents' high-occupancy region. This further validates effective guidance from future occupancy via implicit feature aggregation and explicit losses supervision.

4.3 Ablation Study

To investigate the effectiveness of proposed framework, auxiliary losses and contingency planning strategy in our work, we conduct an ablation study. All ablation experiments are evaluated on $\mathbf{Val14}$ benchmark, and the results is shown in Table. 3. We first evaluate base dual-branch framework without marginal occupancy prediction module, achieving near-SOTA performance with 0.859 NR-S and 0.787 R-S. With the integration of marginal occupancy prediction module, both safety metrics and driving scores exhibit improvements. It is attributed to to the modeling of individual behavior patterns and short-term uncertainty.

Table 3: Ablation results of OccDriver's planning performance with different components. All proposed components contribute to improvements in safety metrics and driving scores. $\mathcal{L}_{oi}$, $\mathcal{L}_{collision}$, $\mathcal{L}_{align}$ denotes occ interference loss, traj-occ collision loss and traj-occ alignment loss. $\mathcal{MP}$ and $\mathcal{CP}$ denotes marginal prediction and contingency planning.

$\mathcal{MP}$	$\mathcal{L}_{oi}$	$\mathcal{L}_{collision}$	$\mathcal{L}_{align}$	$\mathcal{CP}$	Collisions	TTC	Comfort	Progress	NR-Score	R-Score
-	-	-	-	-	0.933	0.905	0.966	**0.893**	0.859	0.787
✓	-	-	-	-	0.938	0.913	0.968	0.889	0.863	0.800
✓	✓	-	-	-	0.943	0.914	**0.975**	0.886	0.864	0.807
✓	✓	✓	-	-	0.960	<u>0.931</u>	0.966	0.882	0.879	0.825
✓	✓	✓	✓	-	<u>0.960</u>	0.928	<u>0.971</u>	<u>0.892</u>	<u>0.885</u>	<u>0.830</u>
✓	✓	✓	✓	✓	**0.971**	**0.938**	0.969	0.885	**0.896**	**0.838**

Table 4: Impact of different horizon T of joint occupancy in occupancy guidance loss.

T	Collisions	TTC	NR-score	R-score
2s	0.933	0.908	0.828	0.787
4s	0.943	**0.916**	0.843	0.799
6s	**0.946**	0.914	**0.845**	**0.801**
8s	0.931	0.917	0.828	0.787

Table 5: Impact of different threshold ζ of high-occupancy regions in $\mathcal{L}_{collision}$

ζ	Collisions	TTC	NR-score	R-score
0.5	0.938	0.916	0.837	0.786
0.6	0.941	**0.917**	0.841	0.799
0.7	0.946	0.914	**0.845**	**0.801**
0.8	**0.948**	0.903	0.840	0.799

Based on proposed framework, we first apply occupancy interference loss $\mathcal{L}_{oi}$ to learn exclusivity between ego's and other agents' occupancy. It enhances Collision metric while benefiting Comfort metric due to improved awareness of spatial interaction and feasible safe areas. In the fourth experiment, a substantial boost in safety metrics (Collision from 0.943 to 0.960 and TTC from 0.914 to 0.931) is observed after adding $\mathcal{L}_{collision}$, attributed to its explicit penalty for collision and excessive proximity to other agents. On this basis, we further introduce $\mathcal{L}_{align}$, which yields an improvement in Progress metric due to ego occupancy's positive guidance. Notably, the continuous increases of driving safety do not compromise the Comfort metric. The above experiments demonstrates that the proposed losses effectively distill spatial information from occupancy branch to trajectory branch.

The bottom row of Table. 3 presents the ablation study on contingency planning. Safety metrics and driving scores reach their peaks at a minor trade-off in the Progress metric. We attribute this to the model's consideration of potential marginal behaviors of relevant agents, adopting a relatively cautious strategy to boost contingency safety.

Table. 4 shows the effects of different horizons of joint occupancy prediction in occupancy guidance loss. We observe a consistent growth in Collision metric and driving scores as the horizon T increases, reaching peak at $T = 6$s. Driving performance starts to degrade when the horizon extends to 8s. It suggests that long-horizon future occupancy captures scene dynamics and agents' long-term interactions, facilitating interaction-consistent planning. However, as the uncertainty of occupancy prediction accumulates over time, leveraging highly uncertain future occupancy as guidance introduces ambiguous or incorrect supervision signal, ultimately degrading planning performance.

In Table. 5, we ablate different thresholds of high-occupancy regions in $\mathcal{L}_{collision}$. Collision metric increases as the threshold ζ rises, while TTC starts decreasing once ζ exceeds 0.6. The reason is that higher threshold offers more deterministic collision supervision, while simultaneously losing uncertainty awareness. Overall, ζ = 0.7 reaches the optimal performance.

5 Conclusion

In this paper, we present OccDriver, a future occupancy guided dual-branch trajectory planning framework. Occupancy branch is incorporated to predict future scenes in occupancy space, guiding interaction-aware trajectory planning through implicit iterative decoding process and explicit loss supervision. Contingency planning is applied, leveraging short-term marginal and long-term joint occupancy predictions simultaneously to mitigate uncertainty risks and sustain scene consistency. Experiments on nuPlan benchmark verify OccDriver's state-of-the-art performance in closed-loop planning, leading to significant improvements in driving safety.

REFERENCES

Lina Achaji, Thierno Barry, Thibault Fouqueray, Julien Moreau, Francois Aioun, and Francois Charpillet. Pretr: Spatio-temporal non-autoregressive trajectory prediction transformer. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2457–2464. IEEE, 2022.

Sepideh Afshar, Nachiket Deo, Akshay Bhagat, Titas Chakraborty, Yunming Shao, Balarama Raju Buddharaju, Adwait Deshpande, and Henggang Cui Motional. Pbp: Path-based trajectory prediction for autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12927–12934. IEEE, 2024.

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

Frédéric Bouchard, Sean Sedwards, and Krzysztof Czarnecki. A rule-based behaviour planner for autonomous driving. In *International Joint Conference on Rules and Reasoning*, pp. 263–279. Springer, 2022.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024.

Weihuang Chen, Fangfang Wang, and Hongbin Sun. S2tnet: Spatio-temporal transformer networks for trajectory prediction in autonomous driving. In *Asian conference on machine learning*, pp. 454–469. PMLR, 2021.

Jie Cheng, Yingbing Chen, and Qifeng Chen. Pluto: Pushing the limit of imitation learning-based planning for autonomous driving. *arXiv preprint arXiv:2404.14327*, 2024a.

Jie Cheng, Yingbing Chen, Xiaodong Mei, Bowen Yang, Bo Li, and Ming Liu. Rethinking imitation-based planners for autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14123–14130. IEEE, 2024b.

Jie Cheng, Yingbing Chen, Xiaodong Mei, Bowen Yang, Bo Li, and Ming Liu. Rethinking imitation-based planners for autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14123–14130. IEEE, 2024c.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.

Alexander Cui, Sergio Casas, Abbas Sadat, Renjie Liao, and Raquel Urtasun. Lookout: Diverse multi-future prediction and planning for self-driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16107–16116, 2021.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning*, pp. 1268–1281. PMLR, 2023.

Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Nitin Singh, and Jeff Schneider. Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2095–2104, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9710–9719, 2021.

Hao Gao, Shaoyu Chen, Bo Jiang, Bencheng Liao, Yiang Shi, Xiaoyang Guo, Yuechuan Pu, Haoran Yin, Xiangyu Li, Xinbang Zhang, Ying Zhang, Wenyu Liu, Qian Zhang, and Xinggang Wang. Rad: Training an end-to-end driving policy via large-scale 3dgs-based reinforcement learning, 2025. URL https://arxiv.org/abs/2502.13144.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11525–11533, 2020.

Ke Guo, Wei Jing, Junbo Chen, and Jia Pan. Ccil: Context-conditioned imitation learning for urban driving. *arXiv preprint arXiv:2305.02649*, 2023.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15273–15282, 2021.

Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zachary Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. *Advances in Neural Information Processing Systems*, 35:20703–20716, 2022a.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023a.

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 *European Conference on Computer Vision*, pp. 533–549. Springer, 2022b.

Yihan Hu, Kun Li, Pingyuan Liang, Jingyu Qian, Zhening Yang, Haichao Zhang, Wenxin Shao, Zhuangzhuang Ding, Wei Xu, and Qiang Liu. Imitation with spatial-temporal heatmap: 2nd place solution for nuplan challenge. *arXiv preprint arXiv:2306.15700*, 2023b.

Zhiyu Huang, Xiaoyu Mo, and Chen Lv. Multi-modal motion prediction with transformer-based neural network for autonomous driving. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2605–2611. IEEE, 2022.

Zhiyu Huang, Haochen Liu, and Chen Lv. Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3903–3913, 2023.

Zhiyu Huang, Peter Karkus, Boris Ivanovic, Yuxiao Chen, Marco Pavone, and Chen Lv. Dtpp: Differentiable joint conditional prediction and cost evaluation for tree policy planning in autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6806–6812. IEEE, 2024.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8350, 2023.

Shivesh Khaitan, Qin Lin, and John M Dolan. Safe planning and control under uncertainty for self-driving. *IEEE Transactions on Vehicular Technology*, 70(10):9826–9837, 2021.

Jinkyu Kim, Reza Mahjourian, Scott Ettinger, Mayank Bansal, Brandyn White, Ben Sapp, and Dragomir Anguelov. Stopnet: Scalable trajectory and occupancy prediction for urban autonomous driving. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 8957–8963. IEEE, 2022.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6):4909–4926, 2021.

Ding Li, Qichao Zhang, Zhongpu Xia, Kuan Zhang, Menglong Yi, Wenda Jin, and Dongbin Zhao. Planning-inspired hierarchical trajectory prediction for autonomous driving. *arXiv preprint arXiv:2304.11295*, 2023a.

Qifeng Li, Xiaosong Jia, Shaobo Wang, and Junchi Yan. Think2drive: Efficient reinforcement learning by thinking with latent world model for autonomous driving (in carla-v2). In *European Conference on Computer Vision*, pp. 142–158. Springer, 2024a.

Tong Li, Lu Zhang, Sikang Liu, and Shaojie Shen. Marc: Multipolicy and risk-aware contingency planning for autonomous driving. *IEEE Robotics and Automation Letters*, 2023b.

Tong Li, Lu Zhang, Sikang Liu, and Shaojie Shen. Marc: Multipolicy and risk-aware contingency planning for autonomous driving, 2023c. URL https://arxiv.org/abs/2308.12021.

Xiang Li, Pengfei Li, Yupeng Zheng, Wei Sun, Yan Wang, and Yilun Chen. Semi-supervised vision-centric 3d occupancy world model for autonomous driving. *arXiv preprint arXiv:2502.07309*, 2025.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. *arXiv preprint arXiv:2406.08481*, 2024b.

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 lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024c.

Haochen Liu, Zhiyu Huang, Jingda Wu, and Chen Lv. Improved deep reinforcement learning with expert demonstrations for urban autonomous driving. In *2022 IEEE intelligent vehicles symposium (IV)*, pp. 921–928. IEEE, 2022.

Haochen Liu, Zhiyu Huang, and Chen Lv. Multi-modal hierarchical transformer for occupancy flow field prediction in autonomous driving. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1449–1455. IEEE, 2023a.

Haochen Liu, Zhiyu Huang, and Chen Lv. Occupancy prediction-guided neural planner for autonomous driving. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 4859–4865. IEEE, 2023b.

Haochen Liu, Li Chen, Yu Qiao, Chen Lv, and Hongyang Li. Reasoning multi-agent behavioral topology for interactive autonomous driving. *arXiv preprint arXiv:2409.18031*, 2024a.

Haochen Liu, Zhiyu Huang, Wenhui Huang, Haohan Yang, Xiaoyu Mo, and Chen Lv. Hybrid-prediction integrated planning for autonomous driving, 2024b. URL https://arxiv.org/abs/2402.02426.

Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7577–7586, 2021a.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021b.

Reza Mahjourian, Jinkyu Kim, Yuning Chai, Mingxing Tan, Ben Sapp, and Dragomir Anguelov. Occupancy flow fields for motion forecasting in autonomous driving. *IEEE Robotics and Automation Letters*, 7(2):5639–5646, 2022.

Chen Min, Dawei Zhao, Liang Xiao, Jian Zhao, Xinli Xu, Zheng Zhu, Lei Jin, Jianshu Li, Yulan Guo, Junliang Xing, et al. Driveworld: 4d pre-trained scene understanding via world models for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15522–15533, 2024.

Xiaoyu Mo, Zhiyu Huang, Yang Xing, and Chen Lv. Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):9554–9567, 2022.

Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, A Koepke, Zeynep Akata, and Andreas Geiger. Plant: Explainable planning transformers via object-level representations. *arXiv preprint arXiv:2210.14222*, 2022.

T-YLPG Ross and GKHP Dollár. Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2980–2988, 2017.

Oliver Scheel, Luca Bergamini, Maciej Wolczyk, Błażej Osiński, and Peter Ondruska. Urban driver: Learning to drive from real-world demonstrations using policy gradients. In *Conference on Robot Learning*, pp. 718–728. PMLR, 2022.

Zihao Sheng, Yunwen Xu, Shibei Xue, and Dewei Li. Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):17654–17665, 2022.

Haoran Song, Wenchao Ding, Yuxuan Chen, Shaojie Shen, Michael Yu Wang, and Qifeng Chen. Pip: Planning-informed trajectory prediction for autonomous driving. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pp. 598–614. Springer, 2020.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.

Mingyu Wang, Zijian Wang, John Talbot, J Christian Gerdes, and Mac Schwager. Game-theoretic planning for self-driving cars in multivehicle competitive scenarios. *IEEE Transactions on Robotics*, 37(4):1313–1325, 2021.

Xiao Wang, Ke Tang, Xingyuan Dai, Jintao Xu, Jinhao Xi, Rui Ai, Yuxiao Wang, Weihao Gu, and Changyin Sun. Safety-balanced driving-style aware trajectory planning in intersection scenarios with uncertain environment. *IEEE Transactions on Intelligent Vehicles*, 8(4):2888–2898, 2023.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European Conference on Computer Vision*, pp. 55–72. Springer, 2024a.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14749–14759, 2024b.

Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Lingyu Xiao, Jiang-Jiang Liu, Sen Yang, Xiaofan Li, Xiaoqing Ye, Wankou Yang, and Jingdong Wang. Learning multiple probabilistic decisions from latent world model in autonomous driving. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1279–1285. IEEE, 2025.

Wenda Xu, Jia Pan, Junqing Wei, and John M Dolan. Motion planning under uncertainty for on-road autonomous driving. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2507–2512. IEEE, 2014.

Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, et al. Generalized predictive model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14662–14672, 2024.

Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9327–9335, 2025.

Zhen Yao, Xin Li, Bo Lang, and Mooi Choo Chuah. Goal-lbp: Goal-based local behavior guided trajectory prediction for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 25(7):6770–6779, 2023.

Boliang Yi, Philipp Bender, Frank Bonarens, and Christoph Stiller. Model predictive trajectory planning for automated driving. *IEEE Transactions on Intelligent Vehicles*, 4(1):24–38, 2018.

Kunpeng Zhang, Liang Zhao, Chengxiang Dong, Lan Wu, and Liang Zheng. Ai-tp: Attention-based interaction-aware trajectory prediction for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 8(1):73–83, 2022.

Lunjun Zhang, Yuwen Xiong, Ze Yang, Sergio Casas, Rui Hu, and Raquel Urtasun. Copilot4d: Learning unsupervised world models for autonomous driving via discrete diffusion. *arXiv preprint arXiv:2311.01017*, 2023.

Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. Drivedreamer-2: Llm-enhanced world models for diverse driving video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10412–10420, 2025.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024.

Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, et al. Diffusion-based planning for autonomous driving with flexible guidance. *arXiv preprint arXiv:2501.15564*, 2025a.

Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, et al. World4drive: End-to-end autonomous driving via intention-aware physical latent world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28632–28642, 2025b.

Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8823–8833, 2022.

Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17863–17873, 2023.

APPENDIX

A DISCUSSIONS

Towards a better understanding of this work, we supplement intuitive questions that may arise and provide empirical answers.

A.1 What is the key of OccDriver as a world model guided planning paradigm?

Most world models in autonomous driving predict future scene evolution by generating videosHu et al. (2023a); Wang et al. (2024a); Yang et al. (2024); Zhao et al. (2025) or 3D representationsZhang et al. (2023); Zheng et al. (2024); Li et al. (2025), serving primarily as data generators and RL simulation environmentsLi et al. (2024a); Gao et al. (2025). DriveWorldMin et al. (2024) and LAWLi et al. (2024b) leverage world models to extract latent scene representations, enhancing spatiotemporal scene understanding of autonomous driving systems. Recent research integrates world models into end-to-end planning. MILEHu et al. (2022a) and DriveDreamerWang et al. (2024a) jointly decode driving actions and the corresponding future scene dynamics. Drive-WMWang et al. (2024b)and Drive-OccWorldYang et al. (2025) utilize explicit future scene predictions to formulate cost functions for trajectory selection or optimization. Latent feature-based methods implicitly transfer future world knowledge to the planner. LatentDriverXiao et al. (2025) unifies the environment's next states and ego's next action as a mixture distribution. World4DriveZheng et al. (2025b) constructs an intention-aware latent world model to rank trajectories.

Compared with prior work, OccDriver extends beyond predicting future scenario evolution by further exploiting interaction modeling in occupancy world and how this can benefit planning safety. OccDriver extracts spatial exclusivity and behavior uncertainty from ego and agents' occupancy distribution, distilling this interaction knowledge to trajectory planning via both feature transfer and occupancy guidance losses.

A.2 What are the key differences between OccDriver and other occupancy-assisted planning methods?

Yang et al. (2025) proposed a 3D occupancy-based world model to evaluate the trajectories, which decoupled from the planner as a separate model. In contrast, the occupancy branch in OccDriver is integrated as part of the planner, where the planning branch incorporates occupancy feature during both training and inference, and the predicted occupancy is leveraged to design training losses that improve the planner's trajectory quality. Compared to Liu et al. (2023b), which only incorporates historical occupancy features to provide supplementary information, our approach further exploits future occupancy predictions to explicitly model multi-agent interactions, and integrate this interaction knowledge into the planning branch through both feature-level fusion and specialized loss supervision.

A.3 Why is contingency planning suitable for learning-based interaction modeling, making occupancy-guided planning superior in safety?

Previous works enhance multi-agent prediction using game theoretic approaches and integrated prediction and planningLiu et al. (2024b) and conditional prediction. Although capable of generating scene compliant scene evolution, these methods suffer from over-optimistic when uncooperative agent behaviour occurs. Contingency planning is one of the strongest optimization-based tools for safety-critical motion planning. Yet, it suffers from discretized artifacts and the difficulty of enjoying the power of data-driven planning. OccDriver proposes a natural paradigm to integrate contingency objective with learning-based algorithm, and mitigate the discretized artifacts by representing contingency objective within occupancy space.

A.4 What would be the broader impact of our paradigm?

The paradigm that combines occupancy-based world model and contingency planning can be extended to a varies of sota methods. For example, we use regression method for trajectory learning to better compare with several strong baselines, but diffusion-based planner has shown superior performance by avoiding learning local average behavior. Our pipeline can be adapted to diffusion-based planner by introducing occupancy as classifier guidance and including occupancy feature as DiT condition. Previous works also exploit RL fine-tuning by collecting rollouts and rewards within a world model. Occupancy-based world models are often used as reward models and should be adaptable to RL paradigm. However, the actual gain and design of such architectures need further investigation.

B LIMITATION AND FUTURE WORK.

Our current approach discretizes the occupancy distribution along the time dimension, lacking supervision across time intervals. Future work will extend this to a unified spatiotemporal occupancy distribution, ensuring smoother temporal consistency. Additionally, we aim to develop an end-to-end framework that seamlessly integrates information between branches. Furthermore, we will explore safe reinforcement learning for contingency planning, leveraging risk-aware optimization objectives such as Conditioned Value at Risk (CVaR).

C ARCHITECTURAL DETAILS

Vectorized Encoder. Ego's and agents' inputs involves kinematic states for past T_h = 2 seconds at 10 Hz and attributes. We only leverage the current state for ego vehicle to prevent shortcut bias. All scene inputs $\{\mathbf{E}, \mathbf{A}, \mathbf{S}, \mathbf{M}\}$ are firstly embedded to hidden dimension D = 128 with separate encoders. Ego state $\mathbf{E}$ is embedded through attention based State Dropout Encoder (SDE). Agents' states $\mathbf{A}$ are embedded through attention-based Feature Pyramid Network (FPN). Static objects $\mathbf{S}$ and map polylines $\mathbf{M}$ are embedded through 2-layer Multi Layer Perceptron (MLP) and PointNet-based vector encoder respectively. All embedded features are concatenated together and added respective positional embedding, as $F_{vec} \in \mathbb{R}^{(1+N_A+N_S+N_M)\times D}$. As depicted in Fig, a 4-layer transformer encoder is employed for feature fusion between scene elments. Each layer consists of a multi-head self-attention and a feedforward neural network (FFN).

Rasterized Encoder. We build a $H \times W$ = 128 × 128 spatial grid, covering the ego vehicle's current position with a range of [-20m, 60m] in the x-direction and [-40m, 40m] in the y-direction (resolution rate = 0.625m / pixel). Follow the practice of Ettinger et al. (2021), we seperately rasterize ego state, agent states and map into occupancy grids $\{\mathbf{O}_e, \mathbf{O}_a, \mathbf{O}_m\}$. The backward flow $\mathbf{FL}$ is constructed by calculating the displacement of the occupancy pixels between two consecutive timesteps for the same agent. We only use current occupancy grids and backward flow as input which are embedded and down-sampled to $F_o, F_f \in \mathbb{R}^{(32\times32\times128)}$ with 2 separate Swin-transformer encodersLiu et al. (2021b). Each swin-transformer encoder is a 2-layer transformer with window self-attention (W-SA) and shifted window self-attention (SW-SA). F_o and F_f are concatenated and fed into a MLP to form spatial scene feature $F_s \in \mathbb{R}^{(32\times32\times128)}$.

Joint Scene Encoder is a two-layer transformer module, which takes scene feature map F_s as input query, deriving encoded occupancy feature F_{occ}. Each layer consists of a self-attention, a cross-attention (F_{vec} as Key and Value), and a feed-forward network (FFN). **Marginal Occupancy Encoder** shares the same structure with Joint scene Encoder. The only difference is that the cross-attention in marginal occupancy encoder queries individual agent feature $F_i \in F_A$ instead of F_{vec}.

Iterative Decoding Structure is composed of 3 iterative decoders: coarse trajectory decoder D_c, future scene decoder D_s and fine trajectory decoder D_f. In trajectory branch, we initialize M = 6 learnable embeddings $Q_m \in \mathbb{R}^{M\times D}$ to model ego's M longitudinal modalities. Q_m is concatenated with $F_E \subset F_{vec}$ and projected as $Q_m^{'} \in \mathbb{R}^{M\times1\times D}$ via a MLP. For joint prediction, $Q_m^{'}$ is further concatenated with $F_A \subset F_{vec}$, forming $Q_{vec} \in \mathbb{R}^{M\times(N_A+1)\times D}$. In occupancy branch, D_s directly utilizes F_{occ} as decoding query. D_c and D_f share the same 4-layer Transformer decoder structure, with each layer consisting of a self-attention, two cross-attentions, and a feed-forward network (FFN). We implement D_s as a 2-layer Transformer decoder. Each layer of D_s contains a self-attention, a cross-attention and an FFN.

Prediction Heads. Deocoded fine trajectory query Q_f is passed through 2 separate MLPs to generate joint prediction trajectories $\mathbf{Y} \in \mathbb{R}^{M\times(N_A+1)\times(T_f/\Delta_{traj})\times6}$ for future T_f = 8 seconds and corresponding confidence scores $\pi \in \mathbb{R}^{M\times1}$. Trajectory states contains $(x, y, cos(\theta), sin(\theta), v_x, v_y)$. Coarse trajectory query Q_c is processed with another MLP to generate coarse trajectories $\mathbf{Y}_c$. Future scene feature Q_s is up-sampled and passes through two separate 2d-CNNs to output future occupancy grids $\mathbf{O} = \{\mathbf{O}_e, \mathbf{O}_a\} \in \mathbb{R}^{M\times(T_f/\Delta_{occ})\times128\times128\times2}$ and flow grids $\mathbf{FL} \in \mathbb{R}^{M\times(T_f/\Delta_{occ})\times128\times128\times2}$. Agents' marginal occupancy features $\{f_{m,i}\}$ is decoded into short-term marginal occupancy $\{\mathbf{O}_{m,i} \mid i = 1, 2, ..., N_m\} \in \mathbb{R}^{N_m\times(T_s/\Delta_{occ})\times128\times128\times1}$ for T_s = 2s. Δ_{traj} and Δ_{occ} indicate the prediction frame rates of trajectory and occupancy, respectively.

Table 6: Loss weight parameters.

Loss	$\mathcal{L}_{reg}$	$\mathcal{L}_{cls}$	$\mathcal{L}_{coarse}$	$\mathcal{L}_{e}$	$\mathcal{L}_{a}$	$\mathcal{L}_{m}$	$\mathcal{L}_{flow}$	$\mathcal{L}_{align}$	$\mathcal{L}_{collision}$
Weight	4	1	0.3	3000	800	2000	1	1	9

D AGENT PRUNING

Although OccDriver reduces the computational complexity of contingency planning from $O(n^2)$ to $O(n)$, calculating marginal occupancy distribution of each agent is unacceptable. Therefore, we propose a rule-based agent pruning method that identifies non-directly-interactive agents according to their ground truth trajectories. During training, the annotated non-interactive agents are directly ignored in marginal occupancy computation, and an agent role prediction head takes in agent features and predicts the confidence score of the role of each agent. During inference, the agents that have above threshold confidence of being non-interactive are pruned in marginal occupancy computation.

We identify 5 regular agent roles: in-lane leader, in-lane follower, lateral intruder(cut-in, cut-out, crossing agents), overtaking target, and non-interactive agents. The interactive relationship is determined by examining whether the ground truth bboxes of ego and agent intersect within a time difference of 4s. If ego reaches the intersection position prior to the agent, the agent is either an in-lane follower or an overtaking target, which is then determined by their heading difference at the intersection position. Similarly, if ego reaches the intersection position later than the agent, the agent is either an in-lane leader or a lateral intruder, which is also further determined by their heading difference at the intersection position. All other agents are considered non-interactive agents as they have no direct impact on the ego vehicle in the short term, thus can be ignored for marginal occupancy computation. Note that predicting detailed agent roles instead of only interactive and non-interactive agents alleviates the causal confusion problem, e.g., in-lane follower.

E TRAINING LOSS

Trajectory planning loss. To avoid mode collapse Liu et al. (2021a), we employ teacher-forcing Williams & Zipser (1989) technique during the training process. We compute the length of the ground-truth trajectory and assign it to the corresponding longitudinal modality based on 20 m segments. The trajectory y^* of the target modality is used to compute the regression loss. For regression loss $\mathcal{L}_{traj}$, we employ the smooth L1 loss, and for classification loss $\mathcal{L}_{cls}$, we utilize the cross-entropy loss.

To facilitate future scene prediction with more reliable initial behaviors and accelerate training convergence, we apply smooth L1 loss $\mathcal{L}_{coarse}$ to coarse trajectory y_c^*. The overall trajectory planning loss $\mathcal{L}_{traj}$ is formulated as a weighted sum of its components:

$$\mathcal{L}_{traj} = w_3\mathcal{L}_{reg} + w_4\mathcal{L}_{cls} + w_5\mathcal{L}_{coarse}. \tag{14}$$

Occupancy prediction loss. We supervise both joint occupancy prediction of target modality $\mathbf{O}_e^*, \mathbf{O}_a^*$ and marginal occupancy predictions $\{\mathbf{O}_{m,i} \,|i = 1, 2, ..., N_v\}$. Due to the significantly larger proportion of unoccupied regions, we utilize focal loss Ross & Dollár (2017) as loss function $\mathcal{L}_e$, $\mathcal{L}_a$, $\mathcal{L}_m$. Additionally, L1 loss $\mathcal{L}_{flow}$ is applied to supervise flow prediction $\mathbf{FL}^*$, facilitating better understanding of traffic scene's dynamics. The overall occupancy prediction loss $\mathcal{L}_{occ}$ is defined as:

$$\mathcal{L}_{occ} = w_6\mathcal{L}_e + w_7\mathcal{L}_a + w_8\mathcal{L}_m + w_9\mathcal{L}_{flow}. \tag{15}$$

Together with w_1 and w_2 for $\mathcal{L}_{align}$ and $\mathcal{L}_{collision}$, detailed loss weight parameters are shown in Table. 6.

F EXPERIMENTAL DETAILS

We perform training on 32 NVIDIA RTX 4090 GPUs with a batch size of 16 for 30 epochs. It takes 20 hours for finishing training. The total number of parameters is 7.9 M and the model size is 31

Table 7: Hyperparameters and configuration settings.

Notation	Parameters	Values
M	Number of modality	6
T_f	Horizon of joint prediction	8s
T_s	Horizon of marginal prediction	2s
Δ_{traj}	frame rate of trajectory	0.1
Δ_{occ}	frame rate of occupancy	1
N_v	Number of covering circles	3
$\boldsymbol{\eta}$	safety margin in $\mathcal{L}_{collision}$	2m
$\boldsymbol{\epsilon}$	occupancy threshold in $\mathcal{L}_{align}$	0.8
$\boldsymbol{\zeta}$	occupancy threshold in $\mathcal{L}_{collision}$	0.7

MB. Learning rate is set to 1e-3, with 3 warm-up epochs and Cosine scheduler. We adopt AdamW optimizer, applying a weight decay of 0.0001. For proper guidance and training stability, we add occupancy guidance loss after 15 epochs. In occupancy guidance loss, the gradient of occupancy predictions is detached to guarantee unidirectional guidance. Details on parameter settings can be found in Table. 7.

Table 8: Impact of different prediction horizons T_{occ} of joint occupancy.

T_{occ}	Collisions	TTC	NR-score	R-score
2s	0.925	0.886	0.836	0.774
4s	0.928	0.886	0.839	0.778
6s	0.933	**0.901**	**0.848**	0.779
8s	**0.935**	0.900	0.846	**0.782**

Table 9: Impact of different weights w_2 of trajectory-occupancy collision loss $\mathcal{L}_{collision}$

w_2	Collisions	TTC	Progress	NR-score
3	0.932	0.903	0.876	0.840
6	0.943	0.908	0.869	0.843
9	**0.946**	**0.914**	0.873	**0.845**
12	0.922	0.879	**0.902**	0.826

Table 10: Impact of different numbers M of decoding modalities.

M	Collisions	TTC	Comfort	Progress	NR-S	R-S
1	0.915	0.887	0.975	0.860	0.823	0.755
3	0.931	0.904	**0.978**	0.869	0.835	0.784
6	**0.946**	**0.914**	0.976	**0.873**	**0.845**	**0.801**

G Additional Ablation Studies

For efficiency, we sample 20% of **nuPlan** training set for hyperparameter ablation experiments and conduct evaluation on the **Val14** split.

Different prediction horizons of joint occupancy. To further investigate the influence of occupancy prediction on trajectory planning, we ablate different joint occupancy prediction horizons T_{occ} without incorporating auxiliary losses. As shown in Table. 8, we observe that increasing T_{occ} leads to simultaneous improvements in both safety metrics and closed-loop driving scores. It suggests that extended occupancy predictions captures long-term dynamics of the traffic, enabling more consistent spatial guidance for trajectory planning. Further performance improvement ceases when T_{occ} extends to 6s, presumably due to the compounding uncertainty inherent in long-term occupancy predictions. In practice, T_{occ} is set as $T_f = 8s$ to ensure consistency with trajectory planning.

Different weights of trajectory-occupancy collision loss. Table. 9 presents the effects of varying w_2 aligned to $\mathcal{L}_{collision}$ on driving performance. As w_2 increases from 3 to 9, OccDriver consistently improves driving safety ($+1.4\%$ Collision and $+1.2\%$ TTC) with minor degradation in Progress,

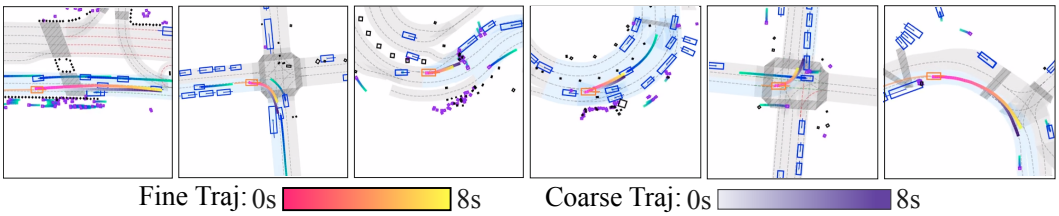


Figure 6: Visualization of coarse trajectory and fine trajectory in nuPlan $\mathbf{Test14-Hard}$ benchmark.

validating the effectiveness of $\mathcal{L}_{collision}$ in collision avoidance. However, further increasing w_2 to 12 results in significant performance degradation, due to over-penalization of collisions, which leads to undesirable shortcut that ego moves out of the occupancy field quickly to avoid collision supervision.

Different numbers of modalities. As presented in Table. 10, a higher number M of modalities contributes to better overall driving performance. A substantial boost in driving scores (NR-S from 0.823 to 0.845 and R-S from 0.755 to 0.801) is observed as M increases from 1 to 6. Finer-grained modality division enables the decomposition of ego's behavioral patterns, mitigating trajectory averaging. Due to OccDriver's joint prediction paradigm, multimodal decoding leads to diverse scene evolution patterns, which facilitate more scene-consistent planning.

H Additional Qualitative Results

Fig. 6 exhibits qualitative comparison between coarse and fine trajectories on Test14-Hard set. In interactive scenarios, compared with collision-prone coarse trajectories, fine trajectories are evidently altered to avoid collisions and handle interactions effectively, validating the effectiveness of the coarse-to-fine decoding architecture and occupancy guidance.

Fig. 7 illustrates the overlapping results of marginal and joint occupancy predictions to validate the effectiveness of contingency planning in modeling behavioral uncertainty. It showcases that marginal occupancy distribution not only captures the waiting behavior but also reflects the aggressive right turn possibility. Interacting with short-term marginal occupancy distribution avoids running into unexpected sudden situations, thus improving safety and comfort. Relevant quantitative results are presented in the ablation study.

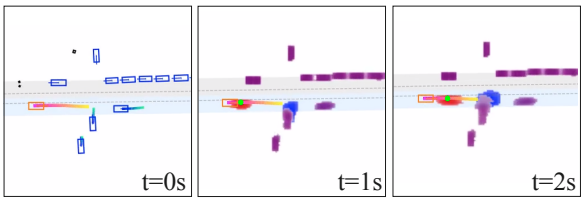


Figure 7: Visualization of contingency planning. Compared to joint occupancy (purple), marginal occupancu (blue) models potential behaviors.

Qualitative closed-loop Planning results on nuPlan **Val14** benchmark is provided in Fig. 8. Qualitative results of occupancy guidance is provided in Fig. 9.

I Video Results

We include several 15-second closed-loop planning videos on $\mathbf{Test14-Hard}$ split in the supplementary material, which intuitively demonstrates OccDriver's closed-loop planning performance.

J Code

We provide the core code of the iterative decoding structure, specialized loss functions and agent pruning mechanism in the supplementary material to facilitate reproducibility.

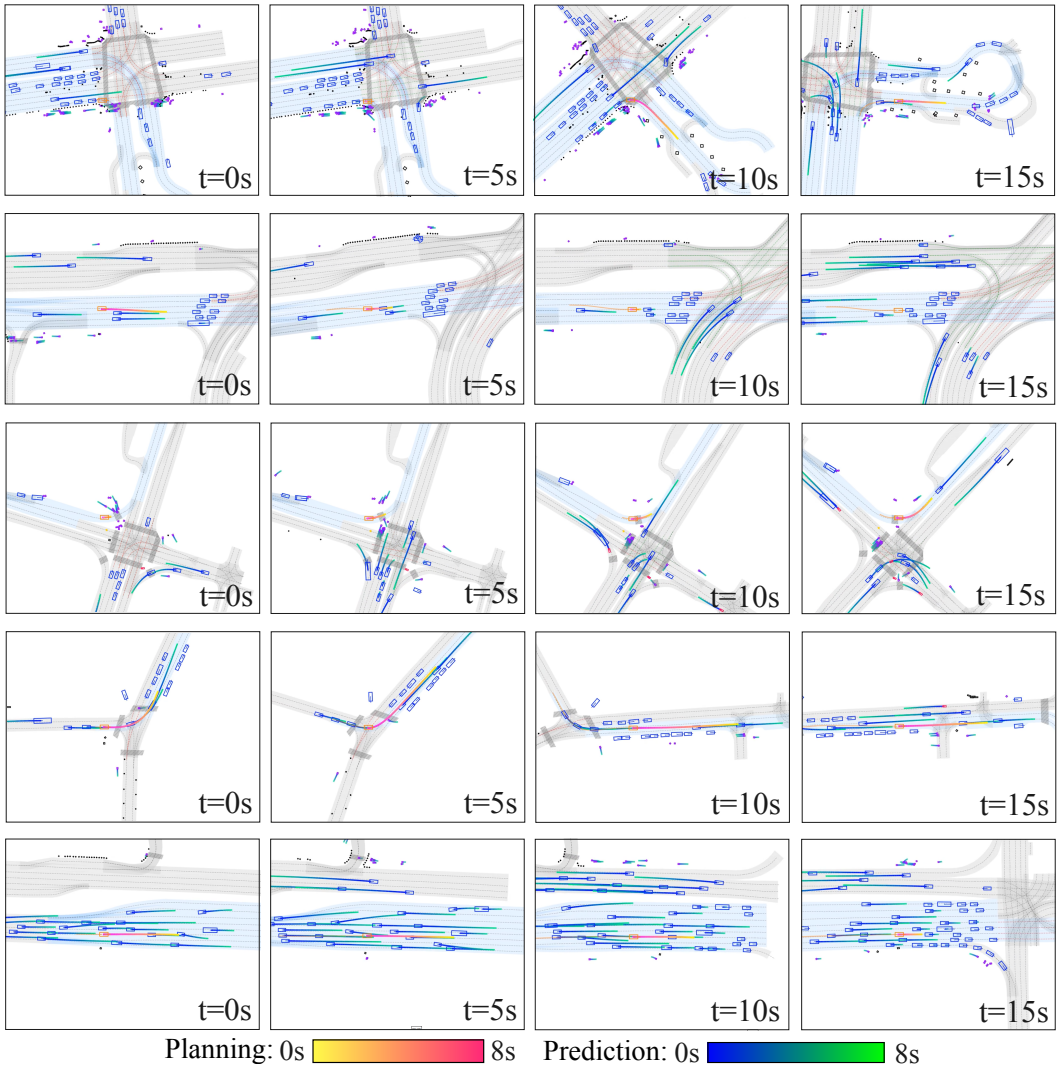


Figure 8: Visualization of closed-loop planning in nuPlan **Val14** benchmark.

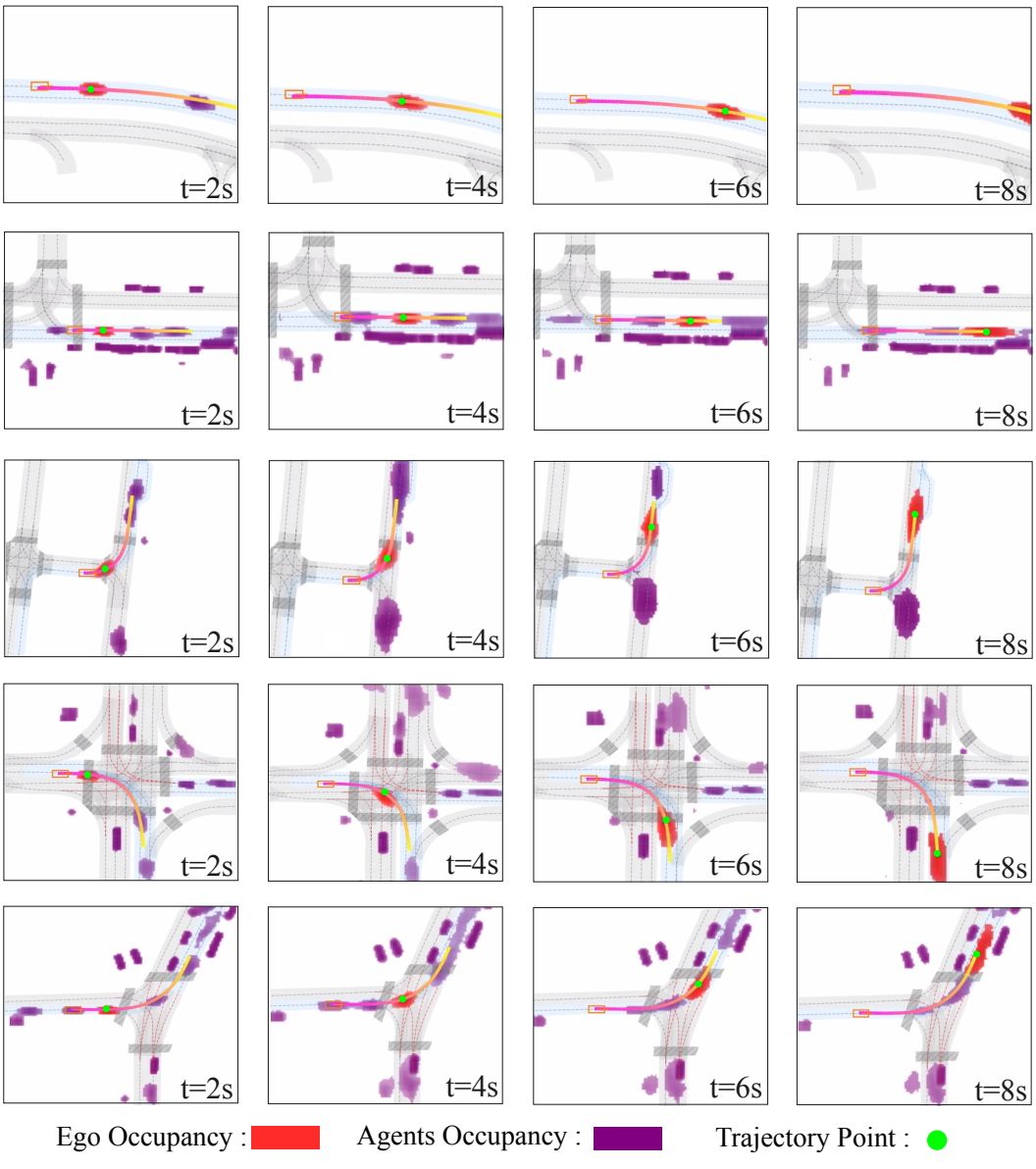


Figure 9: Visualization of closed-loop planning under guidance of future occupancy prediction.