

Drive in Corridors: Enhancing the Safety of End-to-end Autonomous Driving via Corridor Learning and Planning

Zhiwei Zhang¹, Ruichen Yang¹, Ke Wu¹, Zijun Xu¹,
Jingchu Liu², Lisen Mu², Zhongxue Gan¹ and Wenchao Ding¹

Abstract—Safety remains one of the most critical challenges in autonomous driving systems. In recent years, the end-to-end driving has shown great promise in advancing vehicle autonomy in a scalable manner. However, existing approaches often face safety risks due to the lack of explicit behavior constraints. To address this issue, we uncover a new paradigm by introducing the *corridor* as the intermediate representation. Widely adopted in robotics planning, the corridors represents spatio-temporal obstacle-free zones for the vehicle to traverse. To ensure accurate corridor prediction in diverse traffic scenarios, we develop a comprehensive learning pipeline including data annotation, architecture refinement and loss formulation. The predicted corridor is further integrated as the constraint in a trajectory optimization process. By extending the differentiability of the optimization, we enable the optimized trajectory to be seamlessly trained within the end-to-end learning framework, improving both safety and interpretability. Experimental results on the nuScenes dataset demonstrate state-of-the-art performance of our approach, showing a 66.7% reduction in collisions with agents and a 46.5% reduction with curbs, significantly enhancing the safety of end-to-end driving. Additionally, incorporating the corridor contributes to higher success rates in closed-loop evaluations. Project page: <https://zhiwei-pg.github.io/Drive-in-Corridors>.

Index Terms—Integrated Planning and Learning, Collision Avoidance, Vision-Based Navigation

I. INTRODUCTION

END-TO-END autonomous driving has recently gained attention as a promising alternative to traditional methods, offering better scalability and adaptability to complex real-world scenarios. This approach directly maps raw sensor data to trajectories through a unified neural network trained to imitate human drivers. By integrating modules (e.g. perception, prediction, and planning) into a single trainable architecture, it eliminates the need for handcrafted rules.

Manuscript received: January, 2, 2025; Revised April, 2, 2025; Accepted April, 24, 2025.

This paper was recommended for publication by Editor Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Science and Technology Commission of Shanghai Municipality (No. 24511103100), National Natural Science Foundation of China (NSFC) under Grant 62403142 and Shanghai Pujiang Program (23PJ1400900). (*Corresponding author: Wenchao Ding*).

¹Zhiwei Zhang, Ruichen Yang, Ke Wu, Zijun Xu, Zhongxue Gan and Wenchao Ding are with College of Intelligent Robotics and Advanced Manufacturing, Fudan University, China. zhiweizhang23@m.fudan.edu.cn, dingwenchao@fudan.edu.cn

²Jingchu Liu and Lisen Mu are with Horizon Robotics.

Digital Object Identifier (DOI): see top of this page.

©2026 IEEE

Despite the significant promise of end-to-end methods, ensuring safe actions remains a critical challenge. This challenge arises from the inherent lack of precise mathematical guarantees and interpretability in learning-based approaches [1]. Previous studies [2]–[4] seek to address this issue by introducing safety cost functions over sampled trajectories. However, this strategy relies on a large quantity of high-quality trajectory samples and precise evaluation functions. UniAD [5] incorporates post trajectory optimization on occupancy grids, while VAD [6] adopts constraint losses to enhance trajectory safety. While these approaches have shown some success, there remains significant room for improving the safety of end-to-end driving systems.

To resolve the above challenges and elevate the safety of end-to-end autonomous driving, we undertake two primary investigations. The first involves tackling the absence of an effective representation to constrain the vehicle's actions. To this end, we introduce the core concept of corridor into our work. Safe corridor, a widely used tool in robotics planning, is a geometric representation that defines safe, obstacle-free regions the vehicle is going to traverse. We integrate corridor learning into the multi-task pipeline of end-to-end driving by annotating corridors in the dataset, designing network architecture and devising loss functions. These components empower the model to accurately and flexibly identify safe drivable area across diverse and dynamic driving scenarios.

Second, we adopt a trajectory optimization process to generate the planning trajectory. Compared to neural-based planning heads [4]–[7], our approach offers a distinct advantage in interpretability. Previously overlooked considerations such as vehicle kinematics and control bounds are inherently incorporated into the optimization formulation. Moreover, as a representation of intended driving zones, the corridor integrates naturally with the optimization by enforcing constraints on the vehicle's state. Furthermore, leveraging advances in optimization theories [8], we ensure that the optimization process is differentiable, allowing the gradient of the solution to propagate back through the network. Consequently, the optimized trajectory can be seamlessly trained within the joint learning framework, marking a broadened scope of end-to-end driving while fostering a more cohesive planning process.

Bringing everything together, we present **CorDriver**, an end-to-end driving model with explicit and differentiable safety constraints. Our design is implemented on top of the leading-edge model VAD [6] and rigorously evaluated with

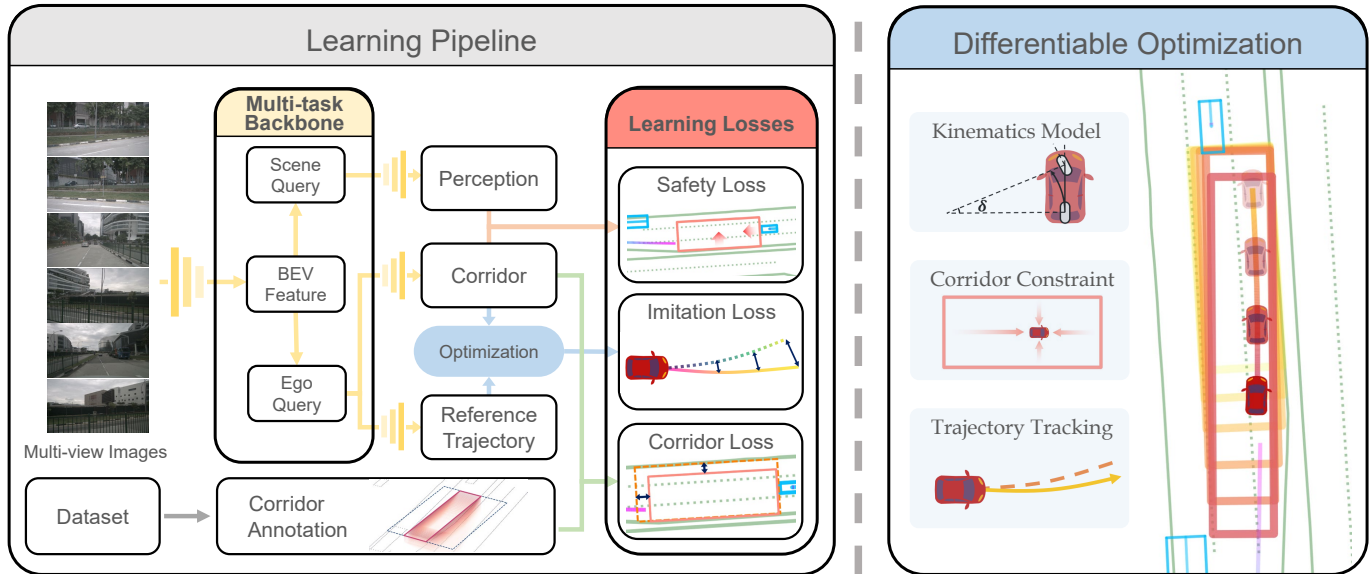


Fig. 1: Architecture of our method. The multi-task backbone processes multi-view images as input and outputs perception results, the reference trajectory, and the corridor. The predicted corridor is supervised using the annotation from the dataset and further refined through the safety loss to minimize overlap with agents and curbs. Built on the kinematic bicycle model, the differentiable optimization module utilizes the corridor as the constraint and is aimed to track the reference trajectory. Finally, the optimized trajectory is trained to imitate human driving actions.

respect to collisions with agents and curbs. On the nuScenes dataset [9], our approach achieves a 66.7% reduction in agent collisions and a 46.5% reduction in curb collisions. Similar improvements are observed in the Bench2Drive [10] closed-loop tests, where integrating corridor into the planning process leads to higher success rate. These advancements represent a significant step toward making end-to-end autonomous driving both safer and more interpretable.

We summarize the contributions of this paper as follows:

- 1) We propose an explicit and interpretable approach to enhance the safety of autonomous vehicles within the end-to-end framework.
- 2) To the best of our knowledge, we are the first to introduce the safe corridor into learning based autonomous driving. We develop a complete pipeline for corridor learning and demonstrate its effectiveness in improving driving safety.
- 3) A differentiable optimization process incorporating the corridor as the constraint enables the generation of safe trajectories while considering vehicle kinematics, thereby enhancing the interpretability of end-to-end driving.
- 4) Through sufficient and comprehensive validations, our approach demonstrates a significant improvement in the safety of end-to-end planning.

II. RELATED WORK

A. Safe End-to-end Driving

A common approach to ensure safety in end-to-end autonomous driving is to assign cost functions to a set of sampled trajectories [2] [3] [4]. Earlier methods [2] [3] predict occupancy grids and count the number of overlapping cells with the ego vehicle as a safety evaluation metric. VADv2 [4] focuses

on modeling trajectory distributions from demonstrations, penalizing colliding samples with lower probabilities. Although effective, these sampling-based methods are highly dependent on the quality and diversity of the sampled trajectories, and a larger trajectory library inevitably increases computational overhead.

Other end-to-end approaches predict trajectories directly through regression, such as UniAD [5] and VAD [6]. UniAD refines trajectories through post-optimization to repel them from occupancy grids, while VAD introduces additional loss functions based on vectorized perception outputs to penalize collisions. More recent works [11] [12] highlight the connection between trajectory quality and ego status, while others [7] [13] [14] propose improved architectures and decoders, advancing trajectory planning to unprecedented levels of precision. Despite these advancements, ensuring safety remains a major challenge in end-to-end trajectory planning.

B. Safe Corridor

Safe corridor is a powerful representation for collision avoidance and trajectory optimization in robotics planning. It is first introduced in robotics planning through IRIS [15], where convex obstacle-free regions are computed through iterative optimizations to plan the footsteps of bipedal robots. Liu et al. [16] enhance the efficiency of this approach and successfully apply the corridor to a trajectory optimization problem by formulating the constraint as linear inequalities, limiting trajectories inside polyhedrons. The concept of safe corridor has been adapted to autonomous driving in the subsequent works [17] [18] [19]. In this work, we seek to extend the use of this powerful representation to the domain of end-to-end driving. The architecture is demonstrated in Fig. 1.

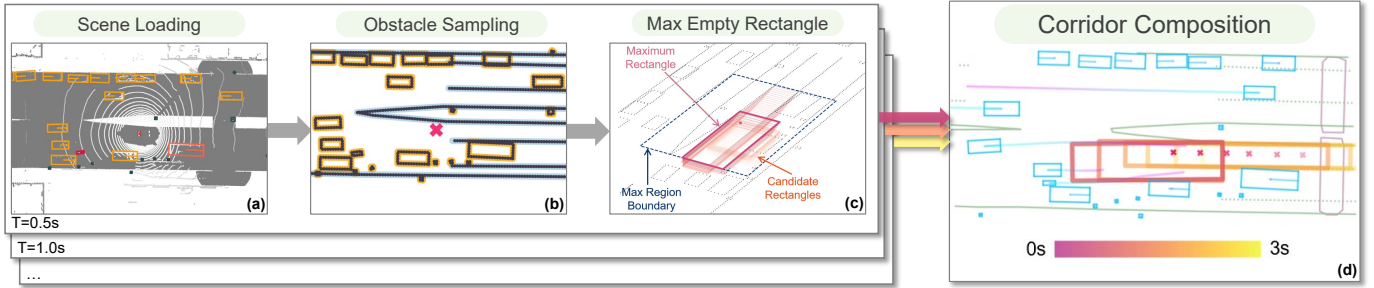


Fig. 2: Corridor Annotation. (a) Agent boxes and map data are loaded from the dataset. (b) Coordinates are sampled. (c) The largest rectangle is identified and highlighted on top and other candidate rectangles are displayed below ground level with high opacity. (d) Iterating over timestamps and combining the results forms the corridor, where rectangles are stacked on the BEV, and colors represent different timestamps.

III. CORRIDOR LEARNING

A. Corridor Representation

Corridors are commonly represented using connected polyhedrons or spheres. However, given the highly structured nature of driving environments, such as lanes and agent bounding boxes, a simpler yet effective choice is to represent the corridor using rectangles. These rectangles can be adequately described on the 2D Bird’s Eye View (BEV) using properties such as position, orientation, and size. To incorporate temporal information in driving scenes, we assign one rectangle to each planning timestamp, resulting in the corridor formulation \mathcal{C} :

$$\mathcal{C} = \{c_t = [c_x, c_y, \theta, l, w]_t \mid t = 1, \dots, N\}, \quad (1)$$

where c_x and c_y denote the rectangle’s center position, θ represents its heading, and l and w correspond to its length and width. The variable N indicates the total number of future timestamps.

In practice, we find this representation effective at capturing safe regions, even in highly dynamic scenarios and on curved roads. Compared to alternatives like 4D occupancy, it is more lightweight and better suited for the subsequent convex optimization. In the rest of the paper, the term *rectangle* refers to the extracted empty space at one timestamp, and the union of rectangles over the planning horizon makes up the *corridor*.

B. Corridor Annotation

To enable the model to identify safe corridors, we need to generate corridor annotations as supervisions. The first step to label corridors in the dataset is to identify both physical and semantic obstacles in the traffic scene. We take the following concerns for obstacle selection. **Agents.** Traffic participants such as vehicles and pedestrians are critical for safe driving, and their bounding boxes are maintained as obstacles. **Curbs.** Road boundaries act as essential constraints, ensuring the vehicle remains within drivable areas. **Lanes.** Lanes also serve as implicit restrictions to guide driving behavior. To this end, we first extract the ego trajectories in a recent period T_{ego} , and keep lanes that do not overlap with the trajectory as flexible semantic obstacles. Based on the geometry of these elements, we sample 2D coordinates along their contours using a specified threshold δ_{obs} , forming the obstacle point set:

$$\mathcal{O} = \left\{ [p_x^o, p_y^o]_i \mid i = 1, \dots, N_o \right\}, \quad (2)$$

where N_o is the total number of points. Notably, this approach is adaptable and can be extended to additional semantics, such as traffic lights and pedestrian crossings.

With the essential obstacles identified, we can now generate the safe corridor by finding the maximum rectangles. At each timestamp t , the 2D obstacle points \mathcal{O}_t are extracted, and their coordinates are transformed into the local frame based on the ego state $(p_x^{ego}, p_y^{ego}, \theta^{ego})$. To enhance computational efficiency, we define a maximum region with dimensions (l_{max}, w_{max}) and disregard points outside the region. We fix the rectangle’s orientation to match the ego’s heading $\theta = \theta^{ego}$, and focus on determining the optimal center position (c_x, c_y) and size (l, w) of the rectangle. The problem is formulated as the well-studied Maximum Empty Rectangle (MER) problem [20] in computational geometry—that is, finding the largest axis-aligned rectangle within a given boundary that contains an origin while avoiding obstacle points. To solve the MER problem, we traverse combinations of obstacle points, where each point precisely defines a candidate rectangle’s edge. The area of each valid rectangle is computed and compared, and the largest rectangle is selected. Repeating this process at each timestamp and composing the rectangles together produces the complete annotation for the corridor. An illustration of the process is shown in Fig. 2.

C. Corridor Learning

Our end-to-end learning model is based on VAD [6], which utilizes vectorized scene representations and employs transformer-based interactions to produce detections, predictions, and maps from the image inputs. Its planning head takes the interacted ego query Q_{ego} , along with the ego status and driving commands, and decodes the ego trajectory $\hat{\xi}$ using an MLP. In the following sections, we refer to this trajectory as the *reference trajectory*. Inspired by the observation that corridors are constructed around an initial path indicating ego intent [15] [16], we introduce an additional head parallel to the original planning head, directly decoding corridor predictions from Q_{ego} .

We supervise the predicted corridor using the annotations from Section III-B. Following prior works [21] [22], the corridor orientations are represented using their cosine and sine components. The corridor loss is computed as

$$\mathcal{L}_{cor} = \mathcal{L}_1(\hat{\mathcal{C}}, \mathcal{C}^*), \quad (3)$$

where \mathcal{L}_1 is the L1 loss, $\hat{\mathcal{C}}$ and \mathcal{C}^* are predicted and ground truth corridors respectively.

To further capture the geometrical properties, we introduce three auxiliary losses for corridor learning, including two safety loss terms that repel the corridor from obstacles. The perception outputs provide the locations of curbs and agents. Curbs are represented as a set of points, \mathcal{M} , marking the road boundaries, while agents are simplified to the vertices of their bounding boxes, \mathcal{A} , obtained from the detection and prediction head. Note that the safety losses do not penalize overlaps with lane dividers. As a result, the corridor predictions may intersect with some lane markers in favor of a larger safe area, imitating the human tendency to utilize adjacent lane space when driving. Subsequently, the edges of each rectangle of the predicted corridor are computed, and the safety losses are determined by the minimum distance between an obstacle point and the edges.

Specifically, the map safety loss \mathcal{L}_{map} and the agent safety loss \mathcal{L}_{agent} are formulated as,

$$\mathcal{L}_{map} = \sum_{t=1}^N \max_i \mathbb{D}(\mathbf{p}_t^i, \mathbf{c}_t), \mathbf{p}_t^i \in \mathcal{M}, \quad (4)$$

$$\mathcal{L}_{agent} = \sum_{t=1}^N \max_i \mathbb{D}(\mathbf{p}_t^i, \mathbf{c}_t), \mathbf{p}_t^i \in \mathcal{A}, \quad (5)$$

where \mathbb{D} represents the function that calculates the distance between a point and the closest edge of the rectangle when the point lies inside the rectangle. It is defined as

$$\mathbb{D}(\mathbf{p}, \mathbf{c}) = \begin{cases} \min_{i \in \{bl, br, tl, tr\}} d_i, & \text{if } \mathbf{p} \in \mathbf{C}, \\ 0, & \text{if } \mathbf{p} \notin \mathbf{C}. \end{cases} \quad (6)$$

where d_i denotes the distance between the point and the four edges (bottom left, bottom right, top left, top right) of the rectangle \mathbf{c} , and \mathbf{C} represents the area enclosed by the rectangle.

Additionally, to prevent the corridor from shrinking excessively, we introduce an area loss to encourage larger corridor sizes. The area loss is defined as

$$\mathcal{L}_{area} = \sum_{t=1}^N e^{-\alpha w_t l_t}, \quad (7)$$

where α is a scaling parameter that controls the magnitude of the penalty, and w_t and l_t represent the width and length of the rectangle at time t , respectively.

D. Corridor Refinement

Leveraging the prediction and mapping results, we further refine the initially predicted corridor to mitigate potential conflicts with perception results. The refinement process is also formulated as an MER problem, as described in Section III-B, but with different data settings. Specifically, the predicted agent boxes and the curbs are treated as obstacle points that the refined rectangle must exclude. The refined rectangle retains the origin of the predicted rectangle $[\hat{c}_x, \hat{c}_y]_t$, while its boundary is constrained by the predicted rectangle size $[\hat{c}_l, \hat{c}_w]_t$. This

ensures that the refinement does not introduce large positional offsets or excessive shape distortion. By solving the MER problem at each timestep, the refined corridor \mathcal{C}^+ is obtained through this post-processing step, effectively “shrinks” the corridor to align with perceived obstacles.

IV. DIFFERENTIABLE OPTIMIZATION WITH CORRIDOR

Tracking the reference trajectory using a vehicle model has been extensively studied in control theory, often formulated as an optimization problem. With the compact representation of our corridor, the safety constraint can be seamlessly integrated, framing the problem as a quadratic programming (QP). Recent advancements [8] [23] have made such optimization processes differentiable, allowing gradients to be backpropagated to the cost functions and constraints. This breakthrough inspires us to embed the optimization process into the network as a differentiable head. To the best of our knowledge, this is the first approach to integrate trajectory optimization into an end-to-end driving framework, thereby expanding the learnable components. The forward and backward processes are described in the following sections.

A. Forward Optimization

The forward pass involves solving an optimization problem to compute the optimal control sequence that minimizes a defined cost function while respecting system dynamics and constraints. The optimization is formulated as follows:

$$\begin{aligned} \min_{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{N-1}} & \sum_{t=0}^{N-1} [\hat{\mathbf{x}}_{t+1}^\top \mathbf{Q} \hat{\mathbf{x}}_{t+1} + \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t] \\ \text{s.t. } & \mathbf{x}_{t+1} = \mathbf{A} \mathbf{x}_t + \mathbf{B} \mathbf{u}_t, \\ & \mathbf{x}_t \in \mathbf{C}_t, \quad \forall t \in [1, N], \\ & \mathbf{u}_t \in [\mathbf{u}_{min}, \mathbf{u}_{max}], \quad \forall t \in [0, N-1], \\ & \mathbf{x}_0 = \mathbf{x}_{init}. \end{aligned} \quad (8)$$

Here, $\mathbf{x} = [p_x, p_y, \theta, v]^\top \in \mathbb{R}^4$ includes the position, heading, and speed of the vehicle, and the control vector $\mathbf{u} = [a, \delta] \in \mathbb{R}^2$ represents the acceleration and steering angle. The bounds \mathbf{u}_{min} and \mathbf{u}_{max} define the feasible range for the control inputs, and \mathbf{x}_{init} specifies the initial state of the vehicle.

The cost function comprises two component, the tracking cost $\hat{\mathbf{x}}_{t+1}^\top \mathbf{Q} \hat{\mathbf{x}}_{t+1}$ and the control effort $\mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t$, weighted by the positive diagonal matrices \mathbf{Q} and \mathbf{R} . Specifically, $\hat{\mathbf{x}}_t = \mathbf{x}_t - \hat{\xi}_t$ denotes the deviation from the reference trajectory $\hat{\xi}_t$.

The dynamics of the vehicle are modeled using a linearized bicycle kinematic model. The discrete-time dynamics are given by $\mathbf{x}_{t+1} = \mathbf{A} \mathbf{x}_t + \mathbf{B} \mathbf{u}_t$ where the matrices \mathbf{A} and \mathbf{B} are defined as

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & -v \sin \theta \Delta t & \cos \theta \Delta t \\ 0 & 1 & v \cos \theta \Delta t & \sin \theta \Delta t \\ 0 & 0 & 1 & \frac{\tan \theta}{L} \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{v}{L \cos^2 \theta} \Delta t \\ \Delta t & 0 \end{bmatrix},$$

ID	Method	L2 (m) ↓				ACR (%) ↓				CCR (%) ↓				Closed-loop Metric ↑			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.	1s	2s	3s	Avg.	DS*	SR* (%)	DS*	SR* (%)
0	ST-P3	1.59	2.64	3.73	2.65	0.69	3.62	8.39	4.23	2.53	8.17	14.4	8.37	-	-	-	-
1	UniAD	0.20	0.42	0.75	0.46	0.02	0.25	0.84	0.37	0.20	1.33	3.24	1.59	37.72	9.54	47.01	20.00
2	VAD-Base	0.17	0.34	0.60	0.37	0.04	0.27	0.67	0.33	0.21	2.13	5.06	2.47	39.42	10.00	50.92	30.00
3	AD-MLP	0.15	0.32	0.59	0.35	0.00	0.27	0.85	0.37	0.27	2.52	6.60	2.93	9.14	0.00	18.54	0.00
4	BEV-Planner	0.16	0.32	0.57	0.35	0.00	0.29	0.73	0.34	0.35	2.62	6.51	3.16	-	-	-	-
5	CorDriver	0.18	0.34	0.59	0.37	0.02	0.06	0.31	0.13	0.16	0.61	2.01	0.92	37.53	12.72	67.85	37.50
6	CorDriver ⁺	0.18	0.35	0.60	0.38	0.00	0.04	0.29	0.11	0.14	0.57	1.86	0.85	-	-	-	-

TABLE I: Planning results on nuScenes. CorDriver utilizes the raw predicted corridor as optimization constraints, while CorDriver⁺ further refines the corridor using perception results. Both models demonstrate exceptional performance in reducing collision rates. In the closed-loop metrics, DS and SR represent Driving Score and Success Rate, respectively. The superscript * indicates tests on bench2drive220, while * denotes tests on dev10.

where Δt is the time step, L the wheelbase of the vehicle.

The constraint $\mathbf{x}_t \in \mathcal{C}_t$ ensures that the ego vehicle remains within the designated corridor. Specifically, the predicted corridor $\hat{\mathcal{C}}$ can be converted into its \mathcal{H} -representation, which characterizes convex polytopes (rectangles in our case) as intersections of half-spaces, each defined by a linear inequality. Consequently, constraining a point \mathbf{p} to lie inside a rectangle is expressed as

$$\mathbf{A}_c \mathbf{p} \leq \mathbf{b}_c, \quad (10)$$

where \mathbf{A}_c and \mathbf{b}_c are the inequality coefficient converted from the rectangle \mathcal{c}_t at each timestamp.

Additionally, the size of the ego vehicle should be considered in the constraint. By projecting the vertices of the ego vehicle into the world frame, the corridor constraint is further detailed as

$$\mathbf{A}_c \left(\bar{\mathbf{R}}_\theta \begin{bmatrix} l_{ego} & -w_{ego} \\ l_{ego} & w_{ego} \\ -l_{ego} & w_{ego} \\ -l_{ego} & -w_{ego} \end{bmatrix}^\top + \mathbf{P}_{xy} \right) \leq \mathbf{b}_c, \quad (11)$$

where $\bar{\mathbf{R}}_\theta$ is the linearized rotation matrix from the current ego heading θ , and \mathbf{P}_{xy} is the stacked position matrix of the ego vehicle. The parameters l_{ego} and w_{ego} represent the half length and width of the ego vehicle, respectively.

Finally by solving the QP, we obtain the optimal control sequence \mathbf{u} . Forward propagating \mathbf{u} from the initial state \mathbf{x}_0 leads to the final trajectory ξ . The imitation loss computes the deviation from the human driving trajectory ξ^* .

$$\mathcal{L}_{imi} = \mathcal{L}_1(\tilde{\xi}, \xi^*). \quad (12)$$

By combining all the weighted losses, namely, the corridor loss (3), map safety loss (4), agent safety loss (5), area loss (7) and imitation loss (12) together with the losses from VAD, we formulate the overall learning loss. Notably, the planning constraints in VAD are no longer required, as our approach introduces a more concrete and compact representation for the trajectory constraint.

B. Backward Gradient Propagation

The backward pass propagates gradients from the QP solution to the problem parameters by leveraging implicit differentiation of the Karush-Kuhn-Tucker (KKT) conditions [8]. The KKT conditions relate the optimal solution to the problem's parameters, enabling gradient flow through the optimization layer. By solving the linearized KKT system, gradients are

efficiently computed, allowing the optimization layer to adjust control policies, planning trajectories and corridor constraints. A detailed discussion of these settings is provided in Section V-D.

V. EXPERIMENTS

A. Implementation Details

Our training process is divided into two stages. In the first stage, the differentiable optimization is excluded, and the remaining tasks — detection, prediction, mapping, planning, and corridor prediction — are trained for 48 epochs. This step ensures a reasonably accurate trajectory and corridor prediction, which is necessary to enable the optimization process in the next stage. In the second stage, spanning 12 epochs, the imitation loss is incorporated while the perception heads are frozen. If the optimization fails during this stage, the reference trajectory $\hat{\xi}$ is used to compute the imitation loss \mathcal{L}_{imi} . The model is trained on 4 A100 GPUs with a batch size of 2 per GPU, requiring approximately 4 days for the first stage and less than 1 day for the second stage.

The hyperparameters are set as follows. Following the convention in most prior works, we set $N = 6$ with a gap of $\Delta t = 0.5s$ between each timestamp, resulting in a planning horizon of 3 seconds. The BEV range is configured to $60m \times 30m$, with a resolution of $0.15m$. For corridor generation, we set ego trajectory during $T_{ego} = [-5s, +5s]$ when selecting lanes as obstacles. The maximum length l_{max} and width w_{max} of the boundary are set to $30m$ and $15m$, respectively, while the obstacle threshold is set to $\delta_{obs} = 0.5m$. For the area loss in equation (7), α is set to 0.01 . The weights for all introduced losses are set to 1.0 .

During inference, in cases where the optimization fails, we employ a soft-constrained variant that relaxes the corridor inequality constraints using slack variables, while all other settings remain as described in Section IV. This fallback mechanism has proven effective, as no failures have been observed in our test cases, ensuring the robustness of our method.

B. Main Results

Previous works [12] [14] have revealed limitations in existing evaluations, such as coarse spatial resolution and lack of ego orientation considerations. To overcome these shortcomings, we adopt the evaluation pipeline from BEV-Planner¹ to

¹<https://github.com/NVlabs/BEV-Planner>

Model	Description	L2 (m) ↓				ACR (%) ↓				CCR (%) ↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.	1s	2s	3s	Avg.
\mathcal{M}_0	Base	0.17	0.37	0.70	0.41	0.0	0.18	1.00	0.39	0.17	1.06	2.87	1.37
\mathcal{M}_1	$\mathcal{M}_0 + \mathcal{L}_{cor}$	0.17	0.37	0.69	0.41 (↓0.0%)	0.04	0.25	0.78	0.36 (↓7.7%)	0.10	1.00	2.87	1.32 (↓3.6%)
\mathcal{M}_2	$\mathcal{M}_1 + \text{Aux Loss}$	0.16	0.33	0.59	0.36 (↓12.2%)	0.0	0.14	0.66	0.27 (↓30.8%)	0.12	0.88	2.85	1.28 (↓6.6%)
\mathcal{M}_3	$\mathcal{M}_2 + \text{Opt}$	0.18	0.36	0.62	0.39 (↓4.9%)	0.0	0.12	0.55	0.22 (↓43.6%)	0.14	0.78	2.38	1.10 (↓19.7%)

TABLE II: Ablation study on each component. The base model \mathcal{M}_0 only trains the planning trajectory. \mathcal{M}_1 introduces corridor learning using only \mathcal{L}_{cor} . \mathcal{M}_2 extends this by incorporating auxiliary losses \mathcal{L}_{map} , \mathcal{L}_{agent} and \mathcal{L}_{area} . \mathcal{M}_3 utilizes the reference trajectory and corridor from \mathcal{M}_2 to generate the optimized final trajectory.

assess planning performance. To be detailed, the ego vehicle, agents, and curbs are projected onto a BEV image with a finer grid size of 0.1m. The Agent Collision Rate (ACR) and Curb Collision Rate (CCR) are calculated by counting intersected pixels. A trajectory is considered in collision if an intersection occurs at any timestamp, and the L2 metric is averaged over time, consistent with VAD [6].

We evaluate our algorithm on the public nuScenes dataset [9], a large-scale multimodal benchmark widely used to advance autonomous driving research. The results of our evaluation are presented in Table I. For a fair comparison, UniAD, VAD-Base, and AD-MLP are incorporated with the ego status in the planning stage. To evaluate our method in closed-loop settings, we integrate it into Bench2Drive [10], a comprehensive benchmark for assessing closed-loop end-to-end driving. Bench2Drive provides a standardized framework for fair comparisons within CARLA v2, encompassing diverse driving scenarios under various weather conditions and locations. Evaluations are conducted in two settings: the bench2drive220 split under protocol v0.0.1 and the dev10 split under protocol v0.03. The driving score and success rate are recorded.

In the open-loop setting on nuScenes, our proposed method demonstrates outstanding performance in reducing collision rates. Specifically, the base model, CorDriver, achieves an average reduction in object collisions by 60.6% compared to the leading model, VAD, and a 42.1% reduction in curb collisions compared to UniAD. With the additional refinement introduced in Section III-D, the enhanced version, CorDriver⁺, further achieves an impressive averaged 0.11% ACR and 0.85% CCR, corresponding to a 66.7% and 46.5% decrease in agent and curb collisions, respectively. In the closed-loop experiments, our method achieves higher success rates, demonstrating its ability to keep the vehicle within lanes and successfully complete the route. With driving scores competitive with existing methods, CorDriver maintains safe and efficient driving behavior without increasing infractions. These results highlight the advantages of corridor learning and planning in reducing collisions and enhancing driving robustness.

C. Ablation on Modular Designs

We compare the effects of different modules, as summarized in Table II. For efficient evaluation, the models are trained for 12 epochs, building on a pre-trained perception model. The validation results show that simply incorporating the corridor learning task yields comparable trajectory precision with a modest reduction in collision rates. However, the introduction of auxiliary losses significantly improves the accuracy of

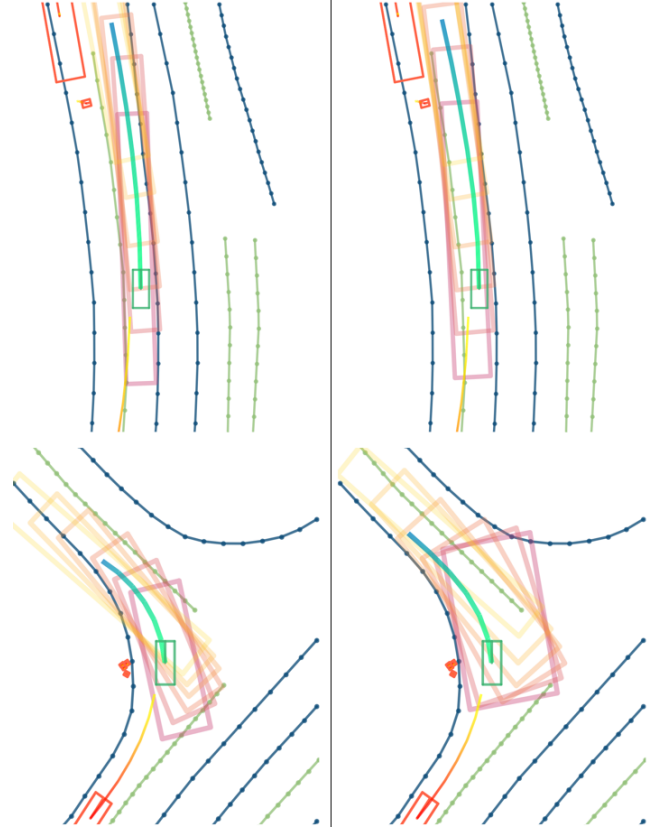


Fig. 3: Visualization comparison of the learned corridors with auxiliary losses. Left: Corridors learned using only \mathcal{L}_{cor} . Right: Auxiliary losses are incorporated, which reduces overlap with curbs. Ground-truth maps and agents are displayed for clearer comparison. Minor intersection between the predicted corridor and ground-truth map may occur due to imperfections in perception outputs, also seen in Fig 4. Note that the bottom case follow the left-hand traffic rules.

corridor predictions, which in turn enhances both the quality and safety of the predicted trajectory. This implies a potential benefit of recognizing driving areas when planning the ego trajectory. Differences in corridor predictions with and without auxiliary losses are visualized in Fig. 3. Additionally, while optimization introduces some precision loss due to modeling limitations, it enforces safety constraints, leading to a substantial reduction in collisions with agents and curbs. We consider this trade-off worthwhile, as minimizing collisions is more critical in practice. The optimization has a more pronounced effect on reducing curb collisions, demonstrating that the corridor effectively captures road geometries and constrains the reference trajectory from deviating off-road. These findings highlight the importance of enforcing explicit constraints in trajectory optimization, showcasing the effectiveness of our interpretable planning process in improving overall safety.

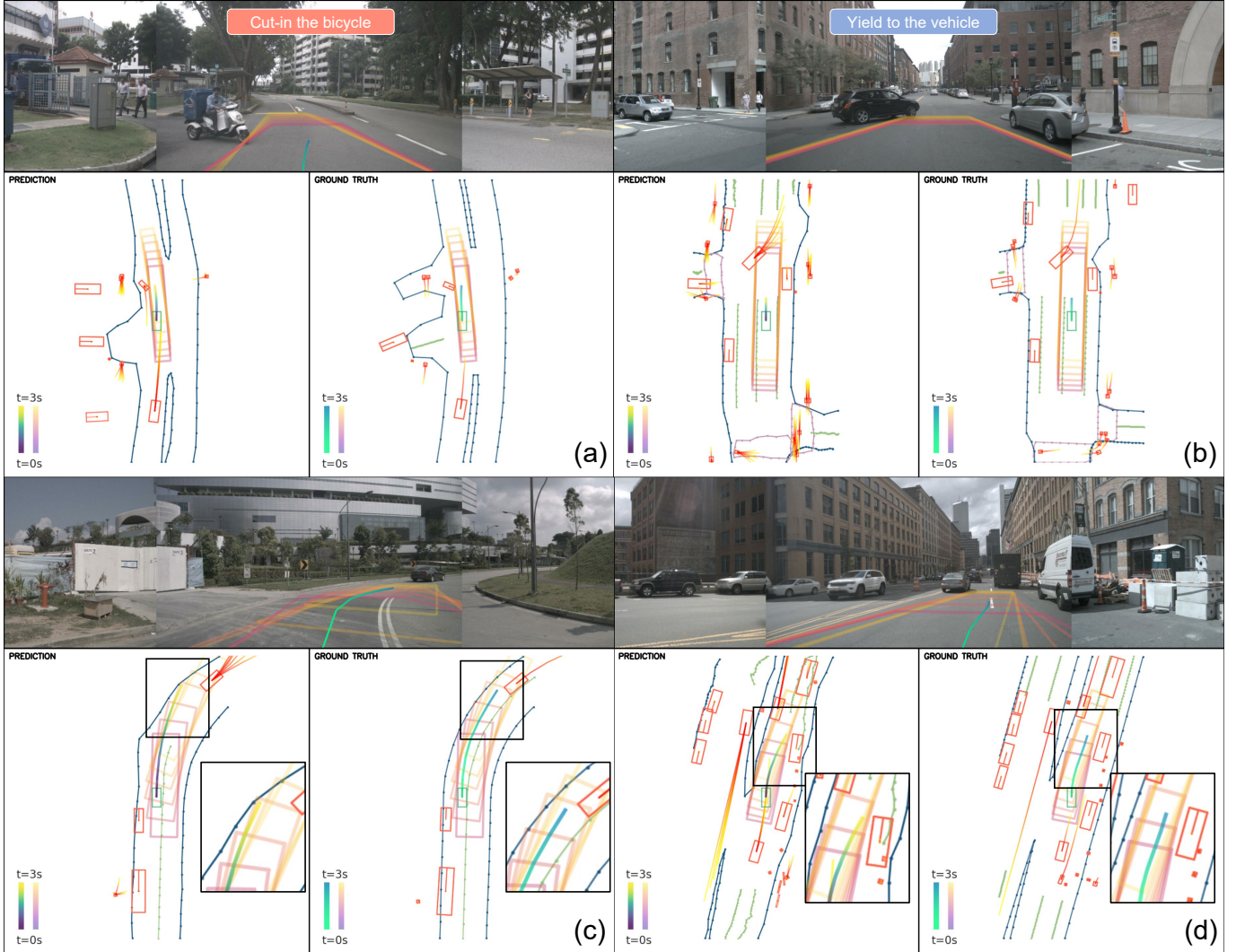


Fig. 4: Qualitative Results. Each subfigure presents the perspective view with projected trajectory and corridor, and the 'PREDICTION' results (perception outputs with the **reference** trajectory) alongside the 'GROUND TRUTH' (ground-truth maps and agents and the **optimized** trajectory). Subfigures (c) and (d) highlight cases where the reference trajectory collides, but the corridor constraints successfully guide the optimized trajectory to remain safe. Slight discrepancies between the perspective and BEV view may occur due to the estimated corridor height.

D. Ablation on Optimization Learning

This section investigates the impact of gradient propagation in the optimization process. Starting with the trained model after the first stage, which is capable of predicting both the corridor and the trajectory, we incorporate \mathcal{L}_{imi} into training to refine these predictions. Specifically, the weight matrices \mathbf{Q} and \mathbf{R} of the QP formulation are treated as learnable parameters. The gradient of \mathcal{L}_{imi} is selectively detached to isolate its effects on the weight matrices, the reference trajectory $\hat{\xi}$ and corridor constraints $\hat{\mathcal{C}}$.

The results shown in Table III indicate that training only the weight matrices yields the best overall performance, while introducing learnable trajectory and corridor components leads to comparable or even degraded performance. Our analysis provides the following insights. During training, we observe larger gradient oscillations for the trajectory and corridor components compared to the weight matrices, which primarily contribute to poorer training outcomes. These fluctuations

arise because the influence of trajectory and corridor varies significantly across different scenarios. For example, when the corridor is spacious, its impact on the optimized trajectory is minimal, whereas tighter constraints force larger control adjustments, leading to more pronounced changes in the optimization result. In contrast, the weight matrices regulate cost term importance, providing more stable influence on the optimization across samples. This smoother gradient behavior is helpful for convergence and performance.

E. Other Results

We present visualization results in several challenging scenarios, such as interacting intersections, high-curvature turns and merging into traffic, as shown in Fig. 4. More cases are available on the project page. Besides, we measure the inference time of our model on an NVIDIA A100 GPU. The multi-task backbone requires an average of 159.6 ms per frame, nearly identical to the base VAD model. In contrast,

ID	weight	corridor	trajectory	L2 (m) ↓				ACR (%) ↓				CCR (%) ↓			
				1s	2s	3s	Avg.	1s	2s	3s	Avg.	1s	2s	3s	Avg.
0				0.178	0.342	0.595	0.372	0.0195	0.0781	0.391	0.163	0.156	0.684	2.012	0.951
1	✓			0.179	0.342	0.597	0.373	0.0195	0.0586	0.313	0.130	0.156	0.606	2.012	0.925
2		✓		0.179	0.341	0.596	0.372	0	0.0391	0.410	0.150	0.195	0.703	2.168	1.022
3			✓	0.182	0.348	0.607	0.379	0.0195	0.0977	0.352	0.156	0.176	0.957	2.442	1.192
4	✓	✓		0.179	0.343	0.602	0.375	0	0.0586	0.430	0.163	0.195	0.606	2.208	1.003
5	✓		✓	0.181	0.346	0.600	0.376	0.0195	0.0977	0.332	0.150	0.195	0.821	2.305	1.107
6		✓	✓	0.175	0.344	0.610	0.376	0	0.0391	0.313	0.117	0.156	0.684	2.872	1.237
7	✓	✓	✓	0.175	0.337	0.594	0.369	0	0.0195	0.391	0.137	0.156	0.664	2.266	1.029

TABLE III: Ablation results of learnable parameters in the optimization. The top 3 averaged metrics are marked with **red**, **orange** and **yellow**. Setting ID 1 achieves best overall performance.

the optimization module adds an extra 44.7 ms per frame, increasing the total computation time by 28% to approximately 200 ms, well above the typical real-time threshold of 30 ms per frame. Further acceleration would be required for deployment on real-world vehicles, particularly in high-speed scenarios.

VI. CONCLUSION

This work enhances the safety of end-to-end autonomous driving by using corridors as a planning representation. Incorporating corridor prediction as a constraint in trajectory optimization increases both safety and interpretability. We also explore the role of differentiable optimization within the end-to-end framework, demonstrating improvements through training certain components. Additionally, our findings reveal that unstable gradients from the optimization process pose a challenge to effective learning. Addressing this issue might involve techniques such as smoothing gradients through penalty functions instead of using hard constraints. How to effectively include the model-based optimizations into scalable end-to-end approaches remains a promising direction for future research.

REFERENCES

- [1] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [2] A. Sadat, S. Casas, M. Ren, X. Wu, P. Dhawan, and R. Urtasun, "Perceive, predict, and plan: Safe motion planning through interpretable semantic representations," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*. Springer, 2020, pp. 414–430.
- [3] S. Casas, A. Sadat, and R. Urtasun, "Mp3: A unified model to map, perceive, predict and plan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 403–14 412.
- [4] S. Chen, B. Jiang, H. Gao, B. Liao, Q. Xu, Q. Zhang, C. Huang, W. Liu, and X. Wang, "Vadv2: End-to-end vectorized autonomous driving via probabilistic planning," *arXiv preprint arXiv:2402.13243*, 2024.
- [5] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 853–17 862.
- [6] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8340–8350.
- [7] B. Liao, S. Chen, H. Yin, B. Jiang, C. Wang, S. Yan, X. Zhang, X. Li, Y. Zhang, Q. Zhang, *et al.*, "Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving," *arXiv preprint arXiv:2411.15139*, 2024.
- [8] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 136–145.
- [9] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nusenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.
- [10] X. Jia, Z. Yang, Q. Li, Z. Zhang, and J. Yan, "Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 819–844.
- [11] J.-T. Zhai, Z. Feng, J. Du, Y. Mao, J.-J. Liu, Z. Tan, Y. Zhang, X. Ye, and J. Wang, "Rethinking the open-loop evaluation of end-to-end autonomous driving in nusenes," *arXiv preprint arXiv:2305.10430*, 2023.
- [12] Z. Li, Z. Yu, S. Lan, J. Li, J. Kautz, T. Lu, and J. M. Alvarez, "Is ego status all you need for open-loop end-to-end autonomous driving?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 14 864–14 873.
- [13] W. Sun, X. Lin, Y. Shi, C. Zhang, H. Wu, and S. Zheng, "Sparsedrive: End-to-end autonomous driving via sparse scene representation," *arXiv preprint arXiv:2405.19620*, 2024.
- [14] X. Weng, B. Ivanovic, Y. Wang, Y. Wang, and M. Pavone, "Paradrive: Parallelized architecture for real-time autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 449–15 458.
- [15] R. Deits and R. Tedrake, "Computing large convex regions of obstacle-free space through semidefinite programming," in *Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2015, pp. 109–124.
- [16] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.
- [17] B. Li, T. Acarman, Y. Zhang, Y. Ouyang, C. Yaman, Q. Kong, X. Zhong, and X. Peng, "Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: A lightweight iterative framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 970–11 981, 2021.
- [18] Z. Han, Y. Wu, T. Li, L. Zhang, L. Pei, L. Xu, C. Li, C. Ma, C. Xu, S. Shen, *et al.*, "An efficient spatial-temporal trajectory planner for autonomous vehicles in unstructured environments," *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [19] W. Ding, L. Zhang, J. Chen, and S. Shen, "Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2997–3004, 2019.
- [20] A. Naamad, D. Lee, and W.-L. Hsu, "On the maximum empty rectangle problem," *Discrete Applied Mathematics*, vol. 8, no. 3, pp. 267–277, 1984.
- [21] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.
- [22] B. Liao, S. Chen, Y. Zhang, B. Jiang, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Maptrv2: An end-to-end framework for online vectorized hd map construction," *International Journal of Computer Vision*, pp. 1–23, 2024.
- [23] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," *Advances in neural information processing systems*, vol. 32, 2019.