2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC)
November 18-21, 2025. Gold Coast, Australia

A Differentiable Constraint-Aware Motion Planning Module for Unsupervised Trajectory Optimization

Zhaoyan Huang[1], Chao Zhang[2], Rui Zhang[3], Yiding Ji[4], Jinlong Hong[1,*], Bingzhao Gao[1,2], and Qingwei Liu[5]

2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC) | 979-8-3315-2418-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/ITSC60802.2025.11423500

***Abstract*—Autonomous driving requires planning trajectories that are not only goal-oriented but also compliant with safety and feasibility constraints such as lane boundaries, inter-agent distances, and motion smoothness. However, in dynamic and uncertain environments, obtaining ground-truth trajectories for supervised learning can be costly or impractical. In this paper, we propose a constraint-aware, supervision-free trajectory planning framework that could learn safe and executable behaviors directly from violation-prone data. Our approach outputs control actions that are rolled out using a kinematic model to generate trajectories, which are then optimized via a differentiable loss incorporating learnable Lagrangian penalties. In particular, we develop a lane boundary constraint based on signed lateral deviations and enforce it through a smooth penalty formulation. Experiments on the nuScenes Open Motion Dataset demonstrate that our method ensures strict compliance with road boundaries while staying close to reference goals, and it can be seamlessly integrated as a plug-in module with End-to-End pipelines.**

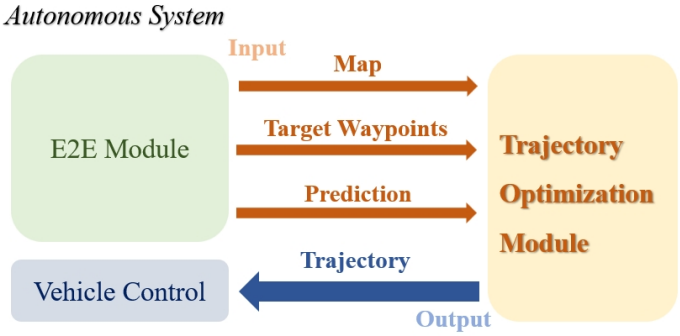

Fig. 1. Architecture of the autonomous driving system. This work focuses on the Trajectory Optimization Module, which receives inputs from the end-to-end module, including perception outputs, map information, predictions of surrounding agents, and the initial target waypoints of the ego vehicle. The module generates trajectories that satisfy safety constraints, vehicle kinematic feasibility, and ride comfort requirements, providing smooth and trackable paths for the downstream control module.

I. Introduction

Planning safe and efficient trajectories is a central task in autonomous driving systems. The planner must produce trajectories that guide the ego vehicle toward its destination while simultaneously satisfying safety-critical constraints such as road boundary adherence, collision avoidance, and comfort. Traditional rule-based planners can handle such constraints explicitly but lack adaptability and scalability in diverse and dynamic scenarios. On the other hand, recent learning-based planners offer better generalization by leveraging large-scale data but often struggle to guarantee hard constraint satisfaction, especially in safety-critical domains.

Most existing methods [1, 2, 3, 4, 5] formulate End-to-end (E2E) systems as imitation learning tasks, typically performing trajectory regression under sparse supervision. These approaches primarily focus on reference trajectory fitting under sparse supervision, while often neglecting robust closed-loop performance and relying heavily on either costly expert demonstrations or dense reward designs that are challenging to define in ambiguous or multi-modal scenarios. In addition, closed-loop oriented methods [6, 7, 8, 9] seek to address system-level feedback but face additional challenges, including non-convex optimization problems [10] and accumulated steering errors [11] during long-horizon execution.To address these limitations, this paper proposes a differentiable, constraint-aware trajectory optimization framework that enhances the safety of end-to-end autonomous driving systems by enforcing hard constraints without relying on ground-truth supervision(as shown in Fig. 1).

A. Related Work

1) Learning-based Planning: E2E driving systems have made significant progress in recent years, offering fully differentiable pipelines that map raw sensor inputs directly to planning or control outputs. Several recent studies have further advanced E2E autonomous driving. For example, TCP [12] enhances decoder expressiveness and unifies control and trajectory prediction through multitask learning from a monocular camera. ST-P3 [13] jointly designs perception and planning from sequential camera inputs, passing shared features through a decoder to enhance semantic scene understanding. UniAD [1] integrates perception, prediction, and planning into a unified network architecture, achieving a full-stack, control-

*This work is supported in part by the National Nature Science Foundation of China (62373289 and 62473291), and the Fundamental Research Funds for the Central Universities. (Corresponding author: Jinlong Hong)

[1]Zhaoyan Huang, Jinlong Hong(corresponding author), and Bingzhao Gao are with the School of Automotive Studies, Tongji University, Shanghai, China 201804. huangzhaoyan@tongji.edu.cn, hongjl@tongji.edu.cn, gaobz@tongji.edu.cn

[2]Chao Zhang is with Shanghai Research Institute for Intelligent Autonomous Systems, Tongji University, Shanghai, China 201804. 2211043@tongji.edu.cn

[3]Rui Zhang is with SAIC Z-ONE Technology Co., Ltd., and School of Computer Science and Technology, Tongji University, Shanghai, China. zhangrui07@saicmotor.com

[4]Yiding Ji is with Robotics and Autonomous Systems Thrust, Systems Hub,The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. jiyiding@hkust-gz.edu.cn

[5]Qingwei Liu is with SAIC Z-ONE Technology Co., Ltd., Shanghai, China. liuqingwei01@saicmotor.com

lable autonomous driving model with improved extensibility, interpretability, safety, and iterative capability.

Despite promising results on public benchmarks such as Waymo [14] and nuScenes [15], as well as simulators like CARLA [16] and industrial deployments such as Tesla, E2E systems still face significant challenges. Compared to traditional modular designs, E2E frameworks operate largely as black boxes, complicating interpretability and system debugging. Moreover, imitation learning-based approaches inherently lack mathematical guarantees of safety and rely heavily on the data distribution of supervised training. As a result, E2E systems often exhibit poor generalization to unseen, complex, or dynamic environments, limiting their robustness under real-world conditions.

2) Physics-Informed and Differentiable Planning: Inspired by physics-informed neural networks (PINNs) [17], differentiable trajectory planners aim to embed physical constraints and system dynamics directly into the optimization process. Early works [18, 19] proposed differentiable layers that incorporate dynamic models and constraint handling into control policies, often relying on quadratic programs or convex approximations. While effective for structured problems, these approaches are limited in expressiveness and scalability due to their reliance on convexity assumptions.

Recent efforts have extended differentiable planning frameworks to autonomous driving. For instance, Zhou et al. [20] proposed FusionAssurance, a novel trajectory-based end-to-end driving fusion framework integrating physics-informed control for safety, capable of handling untrained and neural network-ungeneralizable scenarios. Huang et al. [21] proposed an integrated, differentiable motion prediction and planning approach using a learnable cost function, followed by a general optimization solver, enabling end-to-end optimization across prediction and planning modules. However, such methods do not explicitly guarantee the satisfaction of hard safety constraints such as lane boundary adherence and collision avoidance, which are critical for autonomous driving safety.

In contrast, our method aims to bridge the gap between model-based constraint enforcement and data-driven flexibility by incorporating a constraint-aware trajectory optimization framework within a fully differentiable learning system. Instead of relying on strong convexity assumptions or ground-truth supervision, our approach directly embeds differentiable safety constraints into the training process to improve trajectory feasibility and safety, without the need for an additional numerical optimizer.

B. Contributions

This paper proposes a constraint-aware trajectory planner that learns from violation-prone data without requiring ground-truth supervision. Our method directly predicts control sequences, which are rolled out through a differentiable kinematic model to generate future trajectories. These trajectories are optimized using a loss function that integrates task objectives and Lagrangian penalty terms to softly enforce safety-critical constraints.

Specifically, we design a novel lane boundary constraint based on signed lateral deviation and introduce a learnable Lagrange multiplier to adaptively balance constraint enforcement during training. Unlike black-box imitation learning approaches, our planner supports seamless integration with upstream perception or prediction modules and serves as a safety-aware differentiable optimizer.

We summarize our main contributions as follows:

- We propose a supervision-free trajectory optimization framework based on differentiable kinematic rollout and constraint-driven learning.
- We develop a differentiable Lagrangian penalty formulation that enables learning from violation-prone data while enforcing hard safety constraints.
- We introduce a scene-adaptive constraint weighting mechanism and demonstrate plug-and-play compatibility with E2E pipelines.

II. Preliminary

A. Problem Formulation

We consider a continuous-space, discrete-time autonomous driving scenario involving an ego vehicle (AV) and a varying number of interacting traffic participants. The state of the AV at time t is denoted as s_t^0, and the state of surrounding agents is denoted as s_t^i, where $i = 1, \ldots, N$ indexes other agents. Each agent is semantically labeled (e.g., pedestrian, vehicle), and current states for the are aggregated into $\mathbf{X} = \{s_0^{0:N}\}$. Additionally, we assume the availability of a semantic high-definition map and traffic context, denoted as $\mathbf{M}$.

The planner takes $(\mathbf{X}, \mathbf{M})$ as input and predicts future control sequences for both the AV and surrounding agents. Specifically, it outputs:

- A sequence of ego vehicle controls $\mathbf{u}^0 = [a_t^0, \delta_t^0]_{t=1}^T$;
- Sequences of controls for each surrounding agent $\mathbf{u}^i = [a_t^i, \delta_t^i]_{t=1}^T$, where $i = 1, \ldots, N$.

Using a kinematic model, these control sequences generate corresponding trajectories:

$$\mathbf{x}^0(\mathbf{u}^0) = [x_t^0, y_t^0, \theta_t^0, v_t^0]_{t=1}^T$$

for the AV, and

$$\mathbf{x}^i(\mathbf{u}^i) = [x_t^i, y_t^i, \theta_t^i, v_t^i]_{t=1}^T$$

for each surrounding agent.

The planner solves the following constrained optimization problem:

$$\begin{aligned} \min_{\{\mathbf{u}^0, \mathbf{u}^1, \ldots, \mathbf{u}^N\}} \quad & \omega_{\text{task}} \mathcal{L}_{\text{task}}(\{\mathbf{x}^i\}) + \omega_{\text{smooth}} \mathcal{L}_{\text{smooth}}(\{\mathbf{u}^i\}) + \\ & \omega_{\text{target}} \mathcal{L}_{\text{target}}(\{\mathbf{x}^i\}) \\ \text{s.t.} \quad & g_j(\{\mathbf{x}^i\}) \le 0, \quad j = 1, \ldots, m \\ & \mathbf{x}_{t+1}^i = f_k(\mathbf{x}_t^i, \mathbf{u}_t^i), \\ & \quad \forall i = 0, \ldots, N, \quad \forall t = 1, \ldots, T-1 \end{aligned} \tag{1}$$

a) Inequality Constraints: explicitly include:

- **Collision avoidance constraint (ego and participants):** For the AV and its nearest 10 surrounding traffic participants, the minimum safety distance r_{safe} must be maintained:

$$\begin{aligned} & \|[x_t, y_t] - [x_t^i, y_t^i]\| - r_{\text{safe}} \ge 0, \\ & \quad i = 1, \ldots, 10, \quad \forall t \end{aligned} \tag{2}$$

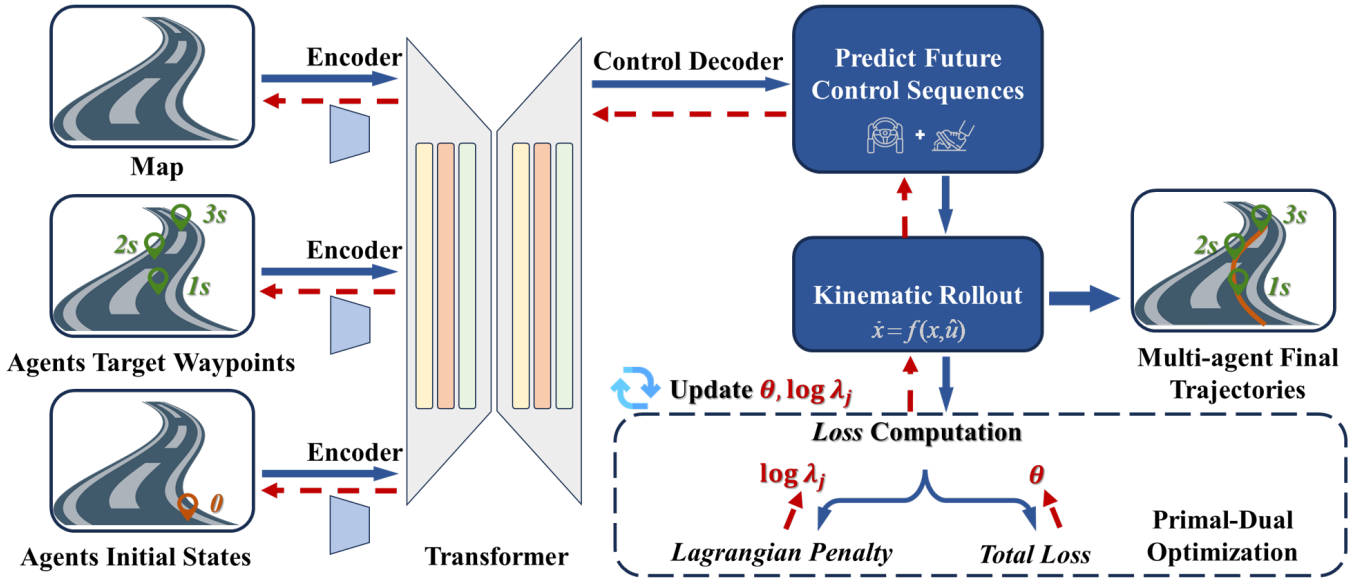

Fig. 2. Overview of the proposed planning architecture. The system takes as input the current states of the ego vehicle and surrounding agents, their assigned future target waypoints, and semantic map features. A transformer-based encoder extracts contextual representations, based on which the network predicts future control sequences for all agents. These are rolled out via a differentiable kinematic model to generate future trajectories, which are then optimized through a constraint-aware objective balancing task performance and safety.

- **Collision avoidance constraint (between participants):** The distance between each pair among the nearest 10 surrounding participants must also satisfy the safety distance constraint:

$$\begin{aligned} &\|[x_t^i, y_t^i] - [x_t^j, y_t^j]\| - r_{\text{safe}} \geq 0, \quad \forall t, \\ &\forall i, j = 1, \ldots, 10, \quad i \neq j \end{aligned} \tag{3}$$

- **Lane boundary constraint:** Ensures the AV remains within lane boundaries:

$$|d_{\text{lat}}(x_t, y_t)| - \left(\frac{w(x_t)}{2} - \epsilon\right) \leq 0, \quad \forall t \tag{4}$$

- **Velocity limit constraint:** Ensures the AV velocity v_t stays within limits:

$$v_t - v_{\max} \leq 0, \quad \forall t \tag{5}$$

b) Equality Constraints: represent general vehicle state dynamics:

$$\mathbf{x}_{t+1} = f_k(\mathbf{x}_t, \mathbf{u}_t), \quad \forall t = 1, \ldots, T-1 \tag{6}$$

where the explicit form of f_k will be detailed in subsequent sections.

c) Soft Constraint Penalties in Objective Function::

- **Control smoothness loss:** Penalizes abrupt changes in control inputs separately for AV ($\mathbf{u}^0$) and participants ($\mathbf{u}^i$), with weights ω^0_{smooth} and ω^i_{smooth}:

$$\begin{aligned} \mathcal{L}_{\text{smooth}}(\mathbf{u}) = \omega^0_{\text{smooth}} \sum_{t=1}^{T-1} \left(\|a^0_{t+1} - a^0_t\|^2 + \|\delta^0_{t+1} - \delta^0_t\|^2\right) \\ + \omega^i_{\text{smooth}} \sum_{i,t} \left(\|a^i_{t+1} - a^i_t\|^2 + \|\delta^i_{t+1} - \delta^i_t\|^2\right) \end{aligned} \tag{7}$$

- **Target deviation loss:** Penalizes deviation from predefined target points for the AV (future timesteps 1, 2, and 3 seconds) and participants (at 3 seconds) with weights ω^0_{target} and ω^i_{target}:

$$\begin{aligned} \mathcal{L}_{\text{target}}(\mathbf{x}(\mathbf{u})) = \omega^0_{\text{target}} \sum_{\tau \in \{1,2,3\}} \|[x_\tau, y_\tau] - [x^{\text{ref}}_\tau, y^{\text{ref}}_\tau]\|^2 \\ + \omega^i_{\text{target}} \sum_i \|[x^i_3, y^i_3] - [x^{i,\text{ref}}_3, y^{i,\text{ref}}_3]\|^2 \end{aligned} \tag{8}$$

This comprehensive formulation incorporates explicit safety and feasibility constraints, smoothness, and precision to predefined targets, effectively balancing multiple aspects critical for autonomous driving scenarios.

III. Methodology

Our proposed planning framework is composed of a modular pipeline that integrates feature encoding, control prediction, differentiable vehicle dynamics, and constraint-aware optimization. In this section, we describe the core components and the training strategy of our approach.

A. Overview of the Architecture

Given the scene input $(\mathbf{X}, \mathbf{G}, \mathbf{M})$, where $\mathbf{X} \in \mathbb{R}^{(N+1)\times d}$ represents the current states of the ego vehicle and surrounding agents, $\mathbf{G} \in \mathbb{R}^{(N+1)\times g}$ denotes the assigned future target waypoints for all agents, and $\mathbf{M} \in \mathbb{R}^{L\times m}$ denotes semantic map features, the system proceeds through four stages. First, a transformer-based encoder extracts contextual representations by fusing agent states, target goals, and map information. Based on the encoded scene features, the network predicts future control sequences $\mathbf{u} \in \mathbb{R}^{(N+1)\times T\times 2}$ for both the ego vehicle and surrounding participants. These control commands are passed through a differentiable kinematic model to roll out corresponding trajectories $\mathbf{x}(\mathbf{u}) \in \mathbb{R}^{(N+1)\times T\times 4}$. Finally, the trajectories are optimized by minimizing a constraint-aware

objective that balances task performance, comfort, and safety constraints. Fig. 2 illustrates the overall architecture of the proposed framework.

B. Multi-Agent Scene Encoder

To effectively model multi-agent interactions and incorporate contextual semantics, we adopt a transformer-based encoder architecture. Each agent, including the ego vehicle and surrounding participants, is embedded as a token enriched with its current kinematic state, assigned target waypoints, and agent type information. Map elements, such as lane boundaries and road topology, are separately encoded and fused with agent embeddings. The transformer applies masked self-attention across all tokens, enabling the network to reason about interactions among agents and between agents and the environment. This process results in contextualized feature representations $\mathbf{h}^i \in \mathbb{R}^d$ for each entity, serving as the foundation for downstream control prediction and constraint-aware optimization.

C. Control Decoder and Kinematic Rollout

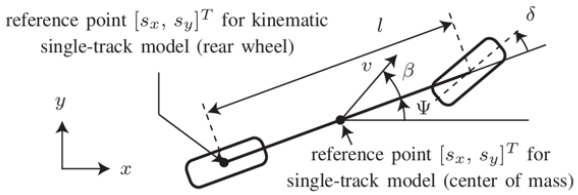

Fig. 3. Kinematic bicycle model [22]

The control decoder predicts the ego vehicle's control sequence $\mathbf{u} = [a_t, \delta_t]_{t=1}^T$, where a_t and δ_t are the acceleration and steering angle at time step t. The future trajectory $\mathbf{x}(t) = [x_t, y_t, \theta_t, v_t]$ is generated via a differentiable kinematic bicycle model(as show in Fig. 3):

$$\begin{aligned} x_{t+1} &= x_t + v_t \cos(\theta_t)\Delta t, \\ y_{t+1} &= y_t + v_t \sin(\theta_t)\Delta t, \\ \theta_{t+1} &= \theta_t + \frac{v_t}{L}\tan(\delta_t)\Delta t, \\ v_{t+1} &= v_t + a_t \Delta t \end{aligned} \tag{9}$$

where L is the vehicle's wheelbase. This rollout mechanism is fully differentiable and enables backpropagation of the loss gradients through control predictions. In addition, the same rollout process is applied to other traffic participants.

D. Lagrangian Penalty-Based Loss Construction

To enforce hard constraints without access to ground-truth trajectories, we design a total loss function that combines task-oriented objectives and differentiable constraint penalties using a Lagrangian formulation (as illustrated in Fig. 4):

$$\begin{aligned} \mathcal{L}_{\text{total}} = \omega_{\text{task}}\mathcal{L}_{\text{task}} + \omega_{\text{smooth}}\mathcal{L}_{\text{smooth}} + \omega_{\text{target}}\mathcal{L}_{\text{target}} \\ + \sum_j \exp(\log \lambda_j) \cdot \mathcal{L}_{\text{constraint}}^{(j)} \end{aligned} \tag{10}$$

1) Lane Boundary Constraint: To ensure the ego trajectory stays within drivable boundaries, we model each lane by its left and right boundaries and define a constraint penalty based on lateral deviation.

For each trajectory point (x_t, y_t), we compute the closest centerline point from the averaged boundaries and calculate the signed lateral deviation using the cross product with the local tangent vector:

$$d_{\text{lat}}(x_t, y_t) = \text{sign}(\vec{v}_{\text{dir}} \times \vec{v}_{\text{offset}}) \cdot \|\vec{v}_{\text{offset}}\| \tag{11}$$

The constraint violation is then defined as:

$$g(x_t) = |d_{\text{lat}}(x_t, y_t)| - \left(\frac{w(x_t)}{2} - \epsilon\right) \tag{12}$$

where $w(x_t)$ is the local lane width and ϵ is a safety margin.

This violation is softly penalized using the smooth $\text{softplus}(z) = \log(1 + e^z)$ function:

$$\mathcal{L}_{\text{lane}} = \lambda_{\text{lane}} \cdot \sum_t \text{softplus}(\beta \cdot g(x_t)) \tag{13}$$

The multiplier λ_{lane} is represented as a learnable parameter in log-space to ensure positivity:

$$\lambda_{\text{lane}} = \exp(\log \lambda) \tag{14}$$

This formulation allows E2E training from constraint-violating samples and enables the model to learn safe behavior from imperfect demonstrations.

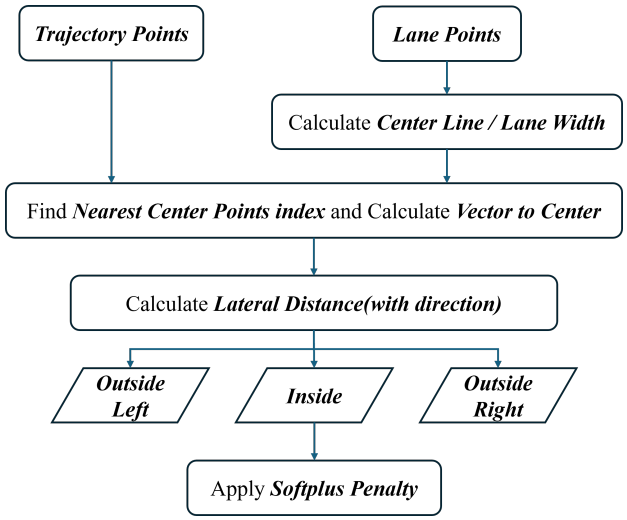

Fig. 4. Calculate Lane Boundary Constraint.

2) Other Constraints: Additional differentiable constraints include collision avoidance:

$$\mathcal{L}_{\text{coll}} = \sum_{i,t} \max\left(0, r_{\text{safe}} - \left\|[x_t, y_t] - [x_t^{(i)}, y_t^{(i)}]\right\|\right)^2 \tag{15}$$

and velocity limits:

$$\mathcal{L}_{\text{speed}} = \sum_t \max\left(0, v_t - v_{\max}\right)^2 \tag{16}$$

Each λ_j term can be fixed or adaptively updated using the dynamic loss weighting module described next.

Algorithm 1 Primal–Dual Constraint-Aware Trajectory Optimization

Require: Initial network parameters θ and Lagrange multipliers $\log \lambda_j$

Require: Learning rates η_{p} (for Primal) and η_{d} (for Dual), training dataset $\mathcal{D}$

for each training epoch **do**

  for each mini-batch $(X, M) \sim \mathcal{D}$ **do**

    Encode scene features using Transformer encoder

    Predict control sequence $\mathbf{u} = [a_t, \delta_t]_{t=1}^{T}$

    Roll out trajectory $\mathbf{x}(t)$ via differentiable dynamics

    Compute the following losses:

$$\mathcal{L}_{\text{task}}, \quad \mathcal{L}_{\text{smooth}}, \quad \mathcal{L}_{\text{target}}, \quad g_j(\mathbf{x}) \quad (j = 1, \ldots, C)$$

    Compute soft constraint penalties:

$$s_j = \text{softplus}(\beta \cdot g_j(\mathbf{x}))$$

    Compute $\lambda_j = \exp(\log \lambda_j)$

    Compute total primal loss:

$$\mathcal{L}_{\text{primal}} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{smooth}} + \mathcal{L}_{\text{target}} + \sum_{j=1}^{C} \lambda_j \, s_j$$

    Primal Update:

    Backpropagate $\nabla_\theta \mathcal{L}_{\text{primal}}$

    Update network parameters: $\theta \leftarrow \theta - \eta_{\text{p}} \nabla_\theta \mathcal{L}_{\text{primal}}$

    Dual Update:

    Detach $\mathbf{x}$ to freeze gradients with respect to θ

    Compute dual loss:

$$\mathcal{L}_{\text{dual}} = -\sum_{j=1}^{C} \lambda_j \, s_j$$

    Backpropagate and Update Lagrange multipliers: $\nabla_{\log \lambda} \mathcal{L}_{\text{dual}}$

  end for

end for

E. Training Procedure

As summarized in Algorithm 1, the model is trained through an unsupervised primal-dual optimization [23] framework without relying on ground-truth supervision. At each iteration, given the initial state and environmental context, the model predicts a sequence of control inputs, and trajectory rollout is conducted using differentiable vehicle dynamics. The loss function integrates task performance, control smoothness, target tracking, and soft penalties for constraint violations. Gradients are propagated throughout the encoder, decoder, dynamics model, and constraint modules, and the network parameters are optimized using Adam with a cosine learning rate schedule.

Concurrently, the Lagrange multipliers associated with each constraint are updated by evaluating constraint violations on the detached primal trajectories and applying gradient ascent. The multipliers are initialized from a uniform distribution and adaptively adjusted during training to softly enforce feasibility without introducing instability. This approach enables the model to progressively converge towards safe and feasible behaviors even when learning from suboptimal or noisy training data.

IV. Experimental Results

A. Dataset

We evaluate our planning framework using the nuScenes dataset [15], a large-scale benchmark for autonomous driving in urban environments. The dataset contains 1,000 scenes, each 20 seconds long, collected in Boston and Singapore with high-resolution sensor data including LiDAR, radar, camera, GPS, and semantic HD maps.

In total, we obtained 28,130 training samples and 6,019 test samples. Notably, our method does not utilize ground-truth future trajectories during training. Instead, we directly employ the predictions generated by an upstream E2E model, combined with map information, as references for optimization. All methods are evaluated on the same set of validation scenarios to ensure consistency and fairness.

B. Evaluation

We evaluate our method based on two key metrics: the number of collision events and the number of lane boundary violations across all validation samples. A collision is recorded if the ego vehicle's trajectory intersects with another dynamic agent within a predefined safety distance. A boundary violation occurs if any trajectory point crosses outside the designated lane region.

We qualitatively illustrate the effectiveness of our approach through two representative scenarios:

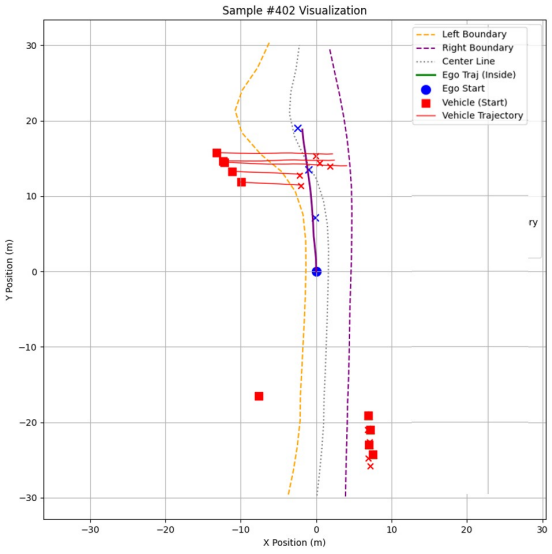

Fig. 5. Scenario 1: The red crosses represent the target waypoints of surrounding traffic participants, while the blue crosses represent the target waypoints of the AV. The planner adjusts the trajectory to proactively avoid surrounding agents while staying close to the reference points, demonstrating safe and feasible motion planning under dynamic traffic interactions.

Scenario 1(as show in Fig. 5): Target Following with Dynamic Obstacle Avoidance. In this scene, the E2E-predicted reference points lie within the drivable area but are located

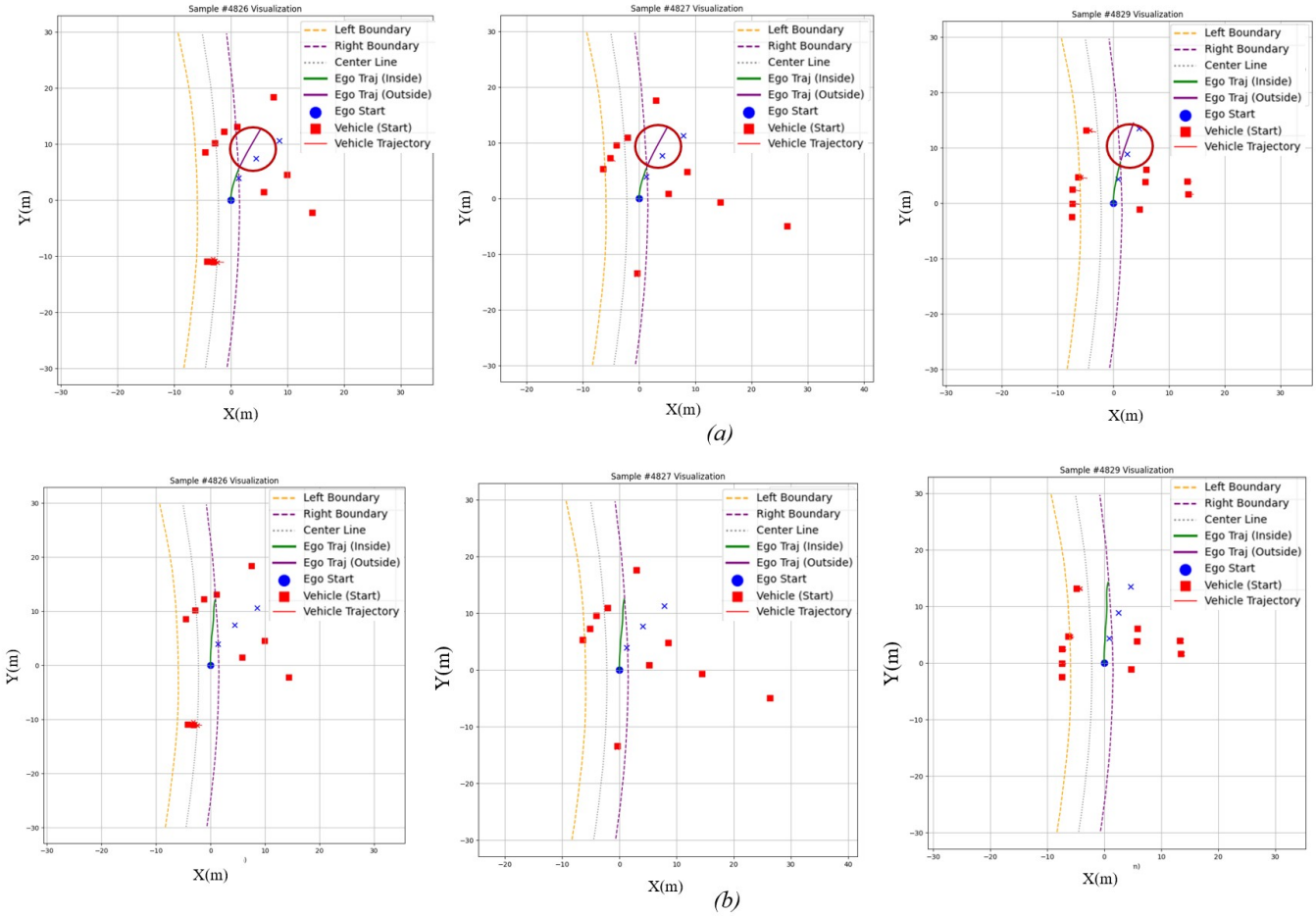

Fig. 6. Scenario 2: Comparison of trajectory optimization results without (a) and with (b) Lagrangian dual-based training. The top row shows that without constraint penalties, optimized trajectories may violate lane boundaries when target points fall outside the drivable area. The bottom row demonstrates that with dual-based training, the trajectories strictly adhere to lane boundaries while remaining close to the target points.

near multiple moving agents. Our planner successfully refines the trajectory to closely follow the targets while proactively avoiding surrounding traffic participants, maintaining safety without significant deviation from the intended path.

Scenario 2(as illustrated in Fig. 6): Handling Target Points Beyond Lane Boundaries. In this scene, some E2E-predicted targets fall outside the valid lane boundary, potentially leading to unsafe behavior. Our optimization framework generates a trajectory that strictly adheres to the lane constraints while staying as close as possible to the provided reference points, demonstrating robust constraint enforcement under adverse conditions.

Table I quantitatively compares the raw E2E outputs and the optimized trajectories over the entire validation set.

TABLE I
COLLISION AND BOUNDARY VIOLATION COMPARISON

Method	Collision	Boundary Violation
E2E Model (SparseDrive [24])	49	17
E2E + Our Optimizer	11	3

The results indicate that our planner significantly reduces both collision and boundary violation occurrences, validating its ability to enhance the safety and feasibility of raw trajectory predictions. Moreover, due to its modular design, the proposed method can be readily integrated into various upstream prediction or planning systems to further improve real-world driving reliability.

V. CONCLUSION

This paper presents a differentiable, constraint-aware trajectory optimization framework for autonomous driving, capable of operating without ground-truth supervision. By integrating a Transformer-based scene encoder, a differentiable kinematic rollout, and a Lagrangian penalty formulation, our method enforces task objectives and critical safety constraints, enabling feasible and safe trajectory generation from imperfect upstream predictions. Experiments on the nuScenes dataset demonstrate that our approach significantly improves constraint satisfaction while maintaining task performance. Although the current method assumes accurate agent state estimation and focuses on short-to-medium horizon planning, future work will explore uncertainty-aware optimization and closed-loop validation. The modular and plug-and-play design further ensures flexibility for integration with diverse prediction pipelines in real-world systems.

REFERENCES

[1] Y. Hu et al. “Planning-oriented Autonomous Driving”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 17853–17862.

[2] X. Jia et al. *DriveTransformer: Unified Transformer for Scalable End-to-End Autonomous Driving*. arXiv preprint arXiv:2405.19620. 2025.

[3] B. Jiang et al. “VAD: Vectorized Scene Representation for Efficient Autonomous Driving”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023.

[4] X. Weng et al. “Para-Drive: Parallelized Architecture for Realtime Autonomous Driving”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 15449–15458.

[5] D. Zhang et al. *SparseAD: Sparse Query-Centric Paradigm for Efficient End-to-End Autonomous Driving*. arXiv preprint arXiv:2404.06892. 2024.

[6] F. Codevilla et al. “Exploring the Limitations of Behavior Cloning for Autonomous Driving”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 9329–9338.

[7] B. Jaeger, K. Chitta, and A. Geiger. “Hidden Biases of End-to-End Driving Models”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 8206–8215.

[8] A. Prakash, K. Chitta, and A. Geiger. “Multimodal Fusion Transformer for End-to-End Autonomous Driving”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 7077–7087.

[9] H. Shao et al. “Safety-Enhanced Autonomous Driving Using Interpretable Sensor Fusion Transformer”. In: *Proceedings of The 6th Conference on Robot Learning (CoRL)*. 2023, 205:726–737.

[10] S. Chen et al. *VADv2: End-to-End Vectorized Autonomous Driving via Probabilistic Planning*. arXiv preprint arXiv:2402.13243. 2024.

[11] K. Renz et al. *CarLLM: Vision-Language Models for Camera-Only Closed-Loop Driving*. arXiv preprint arXiv:2406.10165. 2024.

[12] P. Wu et al. *Trajectory-guided Control Prediction for End-to-End Autonomous Driving: A Simple yet Strong Baseline*. Advances in Neural Information Processing Systems. 2022.

[13] S. Hu et al. “ST-P3: End-to-End Vision-Based Autonomous Driving via Spatiotemporal Feature Learning”. In: *European Conference on Computer Vision (ECCV)*. 2022, pp. 533–549.

[14] P. Sun et al. “Scalability in Perception for Autonomous Driving: Waymo Open Dataset”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 2446–2454.

[15] H. Caesar et al. “nuScenes: A Multimodal Dataset for Autonomous Driving”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 11621–11631.

[16] CARLA Autonomous Driving Leaderboard. *CARLA Autonomous Driving Leaderboard (CARLA, 2022)*. https://leaderboard.carla.org/leaderboard/. 2022.

[17] M. Raissi, P. Perdikaris, and G. E. Karniadakis. *Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations*. Journal of Computational Physics. 2019.

[18] B. Amos and J. Z. Kolter. “OptNet: Differentiable Optimization as a Layer in Neural Networks”. In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. 2017, pp. 136–145.

[19] Y. Qiao et al. “Scalable Differentiable Physics for Learning and Control”. In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. Vol. 119. PMLR, 2020.

[20] H. Zhou et al. “Enhance Planning with Physics-informed Safety Controller for End-to-end Autonomous Driving”. In: *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2024, pp. 1775–1782.

[21] Z. Huang et al. *Differentiable Integrated Motion Prediction and Planning With Learnable Cost Function for Autonomous Driving*. IEEE Transactions on Neural Networks and Learning Systems. 2024.

[22] R. Rajamani. *Lateral Vehicle Dynamics*. Vehicle Dynamics and Control. 2012.

[23] J. Lei, H. F. Chen, and H. T. Fang. *Primal–Dual Algorithm for Distributed Constrained Optimization*. Systems Control Letters. 2016.

[24] W. Sun et al. *SparseDrive: End-to-End Autonomous Driving via Sparse Scene Representation*. arXiv preprint arXiv:2405.19620. 2024.