



**ARTICLE**

# Planning by Simulation: A Query-Centric Search-Based Framework for Interactive Planning in Autonomous Driving

Tian Niu, Kaizhao Zhang, Zhongxue Gan and Wenchao Ding\*

College of Intelligent Robotics and Advanced Manufacturing, Fudan University, Shanghai, China

\*Corresponding Author: Wenchao Ding. Email: dingwenchao@fudan.edu.cn

Received: 19 January 2026; Accepted: 24 March 2026; Published: 27 April 2026

**ABSTRACT:** Ensuring operational safety for autonomous vehicles is a critical challenge in modern engineering, particularly due to the intricate interactions among diverse traffic participants. Traditional approaches often treat planning and prediction as unidirectional processes, failing to capture the dynamic, game-theoretic nature of real-world traffic. In the context of Digital Twins, there is an urgent need for high-fidelity virtual representations that can model the continuous, bidirectional evolution of the ego vehicle and surrounding agents to support robust decision-making under uncertainty. To address these limitations, a novel framework named Planning by Simulation with mutual influence prediction is proposed, which functions as a high-fidelity simulation-based predictive planner for autonomous driving decision-making. This framework explicitly models the iterative interplay between the ego vehicle's planning and the predicted trajectories of surrounding agents within a virtual environment. By integrating a query-centric trajectory prediction mechanism with Monte Carlo Tree Search, the proposed approach orchestrates intelligent model exploration. It iteratively refines the ego vehicle's actions by simulating future scenarios and adapting to the dynamic behaviors of other agents, thereby tightly coupling data-driven predictions with physics-based planning constraints. Comprehensive evaluations on the Argoverse 1 and Argoverse 2 dataset in Metadrive simulator demonstrate the efficacy of this simulation-based approach. The framework successfully captures complex interaction dynamics that static models overlook. The results indicate that the proposed method generates significantly safer, more rational, and human-like trajectories compared to existing baselines, validating the system's high-fidelity predictive capabilities. The proposed framework illustrates the transformative potential of advanced virtual simulation technologies in autonomous mobility. By enabling the continuous integration of predictive data into the planning loop, this study provides a powerful foundation for interpretable and reliable decision-making in virtual engineering systems. It highlights how coupling generative simulation with interactive planning can resolve critical safety challenges in the lifecycle of intelligent autonomous systems.

**KEYWORDS:** Interactive prediction; simulation-based decision making; digital twins; autonomous vehicles

## 1 Introduction

The operational safety of autonomous vehicles (AVs) in complex traffic environments hinges on the accurate deduction of future traffic dynamics and the formulation of rational decisions based on these deductions. In road scenarios characterized by significant uncertainty and dense interactions, AVs must continuously execute behavioral choices that balance safety, efficiency, and comfort. Achieving this goal requires not only predicting the potential motion trends of surrounding traffic participants but also explicitly considering the reciprocal influence of the ego vehicle's behavior on the overall scene evolution. To address

this, the academic community has conducted systematic research ranging from explainable planning methods [1–3] to data-driven decision models [4,5].

In recent years, the end-to-end learning paradigm has gained widespread attention. By mapping raw sensor observations directly to control commands via a unified network structure, these methods demonstrate strong adaptability in complex environments [6–8]. However, due to their reliance on implicit feature representations and opaque reasoning mechanisms, their reliability and verifiability in safety-critical scenarios remain controversial [9,10]. Particularly in strong interaction, multi-game, and long-tail scenarios, end-to-end methods often lack the capacity for systematic reasoning about multi-branch future evolution, which is a core requirement for robust virtual engineering systems for AVs.

Conversely, modular autonomous driving systems typically decouple perception, prediction, and planning into independent stages, offering engineering controllability [11,12]. While stable in deployment, this paradigm often treats the behavior prediction of surrounding agents as a static external input, ignoring the reverse influence of the ego vehicle's planning on others. In reality, traffic is an evolving multi-agent game where every participant continuously adjusts strategies based on others' actions. A high-fidelity virtual engineering system must capture this dynamic interplay rather than relying on static snapshots.

To bridge the gap between prediction and planning, recent research has explored interactive frameworks, notably those based on Monte Carlo Tree Search (MCTS) [13,14] and game-theoretic modeling [15]. These methods enumerate future scenarios under different action hypotheses, endowing planning with foresight and interaction awareness. However, most existing approaches rely on trajectory-level deduction in explicit physical coordinate spaces. This leads to significant computational redundancy and limited search efficiency, making them difficult to scale in high-dimensional multi-agent environments [16].

Recent advances in latent world modeling suggest that performing future deduction in a compact, semantically structured latent space can reduce computational complexity while retaining the expressive power of multi-agent dynamics [17,18]. Inspired by this, search-based planning is revisited from the perspective of a latent world model. Instead of repeatedly generating full physical trajectories [19], a query-driven mechanism is implemented to iteratively reason about scene evolution within a latent interaction space [20,21]. This approach effectively constructs a lightweight, fast-running simulator that allows the system to identify optimal decisions within a virtualized environment before execution in the physical world.

Building on these insights, an interactive trajectory prediction and planning method based on state-space deduction is proposed. Using MCTS as the reasoning backbone, the evolution of future traffic scenarios is represented as a state expansion process on a tree structure. By introducing a query-centric representation mechanism, the bidirectional influence between the ego vehicle and surrounding agents during the search is dynamically characterized. Unlike traditional methods, every tree node is not explicitly represented as a complete set of physical trajectories. Instead, the scene is encoded using latent semantic states, allowing structural information to be reused across search branches, thereby significantly reducing computational overhead.

In the specific deduction process, each expansion of the tree search corresponds to an ego action hypothesis. Under this hypothesis, the query-driven latent state evolution module infers the behaviors of surrounding agents and updates the latent representation of the overall scene. Consequently, planning is no longer a passive selection after a single prediction but an iterative reasoning simulator that continuously corrects and converges through interactive deduction. By exploring multiple potential futures in parallel, this framework systematically evaluates behavioral decisions in terms of safety, smoothness, and traffic efficiency while maintaining computational viability.

Structuring the method as a deep fusion of prediction and planning, the interactive game relationships among multiple agents are explicitly modeled. Experimental results on the Argoverse 1 and Argoverse 2 datasets demonstrate that this method generates safer, more natural, and human-like prediction and planning outcomes, laying a unified state-space reasoning simulator for future DT development in the intelligent driving field.

In summary, the main contributions of this paper are as follows:

1. An interactive trajectory prediction and planning method based on latent world state deduction is proposed, modeling the autonomous driving decision process as a search problem within a unified state space. Acting as a decision-centric virtual simulator, this method uses MCTS to explicitly characterize the bidirectional influence between the ego vehicle and surrounding agents, effectively mitigating the modeling limitations caused by the traditional decoupling of prediction and planning.
2. A query-driven state representation and a stochastic rollout mechanism are introduced to resolve the issues of computational redundancy and poor scalability associated with repeated trajectory generation in explicit physical space. By performing parallel reasoning of multi-agent interactions in a compact semantic space, the proposed method achieves efficient exploration and state reuse for future traffic scene evolution.
3. A deep integration of prediction and planning is realized within a unified framework, ensuring that planning no longer relies on static prediction results but iterates and converges through interactive deduction. Evaluations on the Argoverse 1 and Argoverse 2 datasets indicate that this method produces safer, smoother, and more human-like decisions in complex scenarios, validating the effectiveness of DT-enabled simulation for autonomous systems.

The remainder of this paper is structured as follows: [Section 2](#) reviews relevant literature, [Section 3](#) outlines the proposed framework, [Section 4](#) presents experimental results and comparisons, and [Section 5](#) concludes the paper and discusses directions for future work.

## 2 Related Work

MCTS is a robust search algorithm that combines classical tree search techniques with reinforcement learning. Its success in complex decision-making tasks, such as defeating human world champions in Go and Chess [22], highlights its versatility and effectiveness. Within autonomous systems, MCTS has been widely applied to multi-robot active perception [23,24] and autonomous vehicle control [25]. By effectively balancing exploration and exploitation, MCTS is well suited to address the complex and uncertain dynamics of real-world traffic scenarios [26,27]. However, conventional MCTS-based planning approaches often rely on simplistic action sets, leading to suboptimal trajectories and inefficient search processes. To mitigate these limitations, the proposed approach integrates MCTS with parallel scenario prediction and Frenét frame-based trajectory generation, significantly improving both planning efficiency and trajectory quality.

### 2.1 Trajectory Simulation in Structured Spaces

Trajectory simulation is a fundamental component of motion planning, providing physically feasible and dynamically consistent paths for autonomous vehicles. Traditional methods operating in the Cartesian coordinate space [28] often suffer from inefficient exploration and irregular motion. To address these shortcomings, structured representations such as the Frenét frame have been widely adopted for lane-aligned motion generation [29–31]. These representations decompose vehicle motion into longitudinal and lateral components, simplifying optimization and ensuring kinematic consistency [32]. Recent studies have further extended these ideas to support high-speed driving and occlusion-aware decision-making [33]. In

this work, the structured Frenét representation is incorporated into a latent-world reasoning simulator, where it provides a physically grounded foundation for decoding latent rollouts into interpretable and smooth trajectories.

## 2.2 Query-Centric Representation Learning for Planning

Recent advances in trajectory prediction have transitioned from agent- or scene-centric coordinates [34–36] toward query-centric representations, where each agent’s future behavior is inferred by querying a shared latent scene embedding [20]. Such representations enable efficient reuse of invariant scene features and facilitate parallel reasoning across multiple agents. Beyond trajectory prediction, query-centric learning has also been employed in end-to-end frameworks [21], allowing planners to operate directly on latent representations of the world state rather than explicit coordinates. Building upon these ideas, the query-centric paradigm is extended to the planning domain itself, introducing a framework in which MCTS expansions are guided by latent trajectories. Each trajectory interacts with each other to simulate the evolution of multi-agent behaviors, forming the foundation of the proposed physical world rollouts.

## 2.3 Interactive and Latent-World Planning

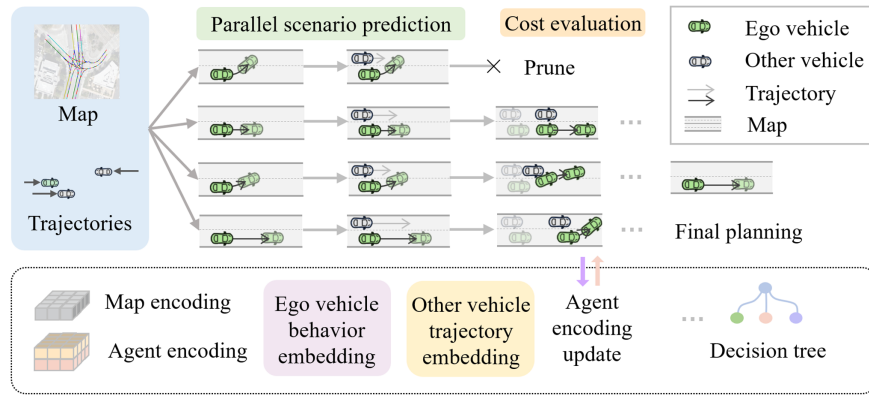
Coupling prediction and planning has long been recognized as critical for achieving robust autonomous driving [37]. Early studies addressed this challenge through game-theoretic formulations [38] or ego-conditioned motion prediction [39]. More recent efforts, such as GameFormer [15], employ transformer-based joint models to learn differentiable prediction–planning couplings. Meanwhile, world-model-driven approaches have emerged [40], conducting planning via imagination or simulation within a latent space rather than relying on explicit environment modeling. Representative works [41,42] highlight the increasing emphasis on latent-world reasoning as a scalable alternative to explicit simulation.

In contrast to these end-to-end generative approaches, the proposed method retains a search-based structure that preserves interpretability while incorporating physical world rollouts for efficient inference. This combination enables query-driven interactive reasoning between the ego vehicle and surrounding agents, unifying simulation and decision-making within a single iterative process.

## 3 Method

The proposed framework, as illustrated in Fig. 1, integrates a learning-based parallel scenario prediction module with an MCTS planner. The key innovation lies in modeling the bidirectional influence between the ego vehicle’s planning and the predicted behaviors of surrounding agents, enabling more realistic and adaptive decision-making. Each traffic scenario is encoded from the Frenét frame into a latent state, which abstracts away explicit geometric coordinates and focuses on interaction semantics, thereby reducing computational overhead while maintaining essential context. By leveraging the Frenét frame for trajectory generation and a query-centric prediction module for dynamic interaction modeling, the framework facilitates safer and more human-like decision-making in complex traffic scenarios.

The planning problem is formulated as a tree-structured search, aiming to approximate the intractable continuous-space policy planning problem by sampling a discrete set of ego trajectories. Given a sequence of past observed states  $\{s_1^t, s_2^t, \dots, s_N^t\}_{t=1}^T$  and high-definition map information  $M$ , the objective is to select the optimal action  $a \in A$  for the ego vehicle at time  $T + 1$  that minimizes the cost function  $C$ . Here,  $N$  represents the number of agents in the scene, and  $\{s_1^t\}_{t=1}^T$  denotes the state history of the ego vehicle.



**Figure 1:** Illustration of the proposed trajectory planning framework. The model uses high-definition maps and historical agent states to extract lane polygons and trajectories, generating the ego vehicle’s planning set. Its prediction model decouples map and agent encoding, producing high-accuracy future trajectories in parallel. MCTS prunes suboptimal branches, with remaining states forming new nodes for rapid re-encoding. This iterative cycle dynamically refines both the ego vehicle’s plans and surrounding agents’ predicted trajectories, enhancing decision-making accuracy and adaptability.

### 3.1 Planning Process

The planning process is outlined in Algorithm 1. The core idea is to iteratively simulate future scenarios and evaluate their costs using MCTS. The algorithm begins by initializing the search tree with the current state of all agents. At each iteration, the simulate function Algorithm 2 is invoked to explore potential future states and update the tree.

---

#### Algorithm 1: Planning process

---

**Input:** sequence of past observed states  $\{s_1^t, s_2^t, \dots, s_N^t\}_{t=1}^T$  and high-definition map information  $M$

**Output:** the most appropriate action  $a$

```

1: function SELECT ACTION( $s_{n=1, \dots, N}^T, d$ )
2:   while  $iter < iterations$  do
3:     SIMULATE( $s_{n=1, \dots, N}^T, d$ )
4:   end while
5:   return  $\operatorname{argmax}_a Q(s_{n=1, \dots, N}^T, a)$ 
6: end function
    
```

---

The planning algorithm, as outlined in Algorithm 1, constructs a trajectory tree by iteratively simulating future scenarios. Each node in the tree represents a state

$$s_n^i = \left( x_n^i, y_n^i, v(x)_n^i, v(y)_n^i, \theta_n^i \right), \quad (1)$$

where  $x$  and  $y$  denote position,  $v(y)$  and  $v(x)$  represent velocities in the longitudinal and lateral directions, respectively, and  $\theta$  is the steering angle. The action set

$$A = \{0.5, 1.5, \dots, 13.5, 14.5\} \quad (2)$$

defines target speeds in the Frenét frame, which are used to generate kinematically feasible trajectories along the road centerline.

**Algorithm 2:** Simulation process

---

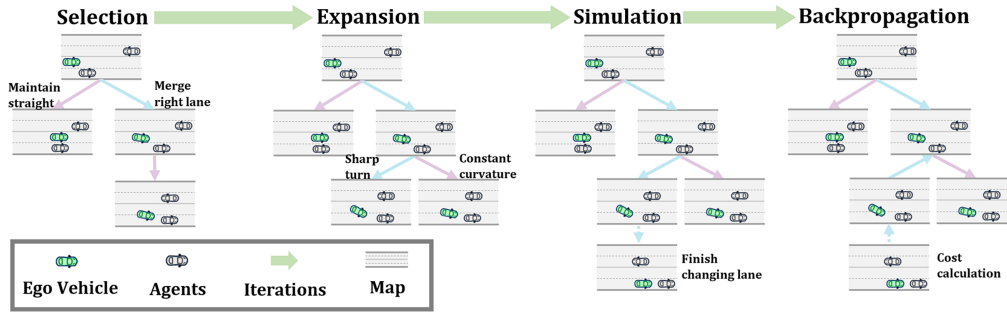
```

1: function SIMULATE( $s, d$ )
2:   if  $d = 0$  then
3:     return 0
4:   end if
5:   if  $s \notin Tree$  then
6:     for  $a \in A$  do
7:        $Q(s, a), N(s, a) \leftarrow 0, 0$ 
8:     end for
9:      $Tree = Tree \cup \{s\}$ 
10:    return ROLLOUT( $s, d$ )
11:  end if
12:   $a \leftarrow \operatorname{argmax}_a Q(s, a) + c \sqrt{\frac{\log N(s)}{N(s, a)}}$ 
13:   $s' \sim \text{TRANSITION}(s, a)$ 
14:   $r \sim -\text{COST}(s, a)$ 
15:   $q \leftarrow r + \lambda \text{SIMULATE}(s', d - 1)$ 
16:   $N(s, a) \leftarrow N(s, a) + 1$ 
17:   $Q(s, a) \leftarrow Q(s, a) + \frac{q - Q(s, a)}{N(s, a)}$ 
18:  return  $q$ 
19: end function
20:
21: function ROLLOUT( $s, d$ )
22:   if  $d = 0$  then
23:     return 0
24:   else
25:      $a \sim \text{RANDOM}(A)$ 
26:      $s' \sim \text{TRANSITION}(s, a)$ 
27:      $r \sim -\text{COST}(s, a)$ 
28:     return  $r + \lambda \text{ROLLOUT}(s', d - 1)$ 
29:   end if
30: end function

```

---

As outlined in Algorithm 2 and shown in Fig. 2, the basic steps of the proposed planner follow the traditional MCTS framework [14]. First, the selection step identifies a node near the current state that maximizes the upper confidence bound (UCB), balancing exploration and exploitation. Upon reaching a state not yet explored, new leaves are expanded by iterating over all possible actions during the expansion step. Next, random simulations are performed to a fixed depth to evaluate the value of the leaves in the rollout step. The final scene is simulated from the leaf node by applying a random action for the ego vehicle. Finally, the statistics of all selected nodes are updated via backpropagation. Through these four steps, the planner iteratively generates a growing, asymmetric tree until a predefined maximum number of simulations is reached and the ego vehicle arrives at the goal. At this point, performance is evaluated, and the action corresponding to the maximum reward is executed.



**Figure 2:** Illustration of planning future scenarios. The reasoning process, which iterates multiple times, is based on MCTS and consists of four steps: selection, expansion, simulation, and backpropagation.

At each selection step, the action  $a \in A$  that maximizes  $Q(s, a) + c\sqrt{\frac{\log N(s)}{N(s, a)}}$  is executed [43], where  $N(s)$  and  $N(s, a)$  represent the number of visits. Here  $c$  is a hyperparameter that controls the exploration-exploitation trade-off.

### 3.2 Ego Vehicle Transition

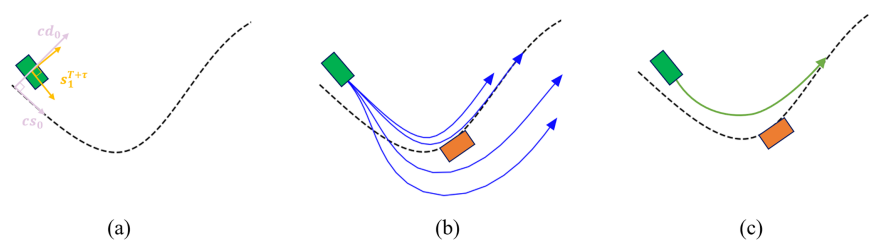
Given the high-definition map information  $M$  and the target global route, the directional vector of the target road centerline in the Cartesian coordinate system is first calculated. The most recent leaf node

$$s_1^{T+\tau+1} = (x_1^{T+\tau+1}, y_1^{T+\tau+1}, v(x)_1^{T+\tau+1}, v(y)_1^{T+\tau+1}, \theta_1^{T+\tau+1}) \quad (3)$$

on the branch is then transformed from Cartesian coordinates to Frenét coordinates. This enables the determination of the longitudinal and lateral positions ( $cs_0, cd_0$ ) of the ego vehicle relative to the centerline. Next, a target extension time  $\Delta t$  is sampled from the range  $[T_{min}, T_{max}]$ , yielding the longitudinal distance set  $cT$  along the center line. To obtain the corresponding lateral position  $cD$ , the lateral displacement  $\Delta d$  is sampled from the range  $[-m_{road,max}, m_{road,max}]$ . Using the chosen target speed  $a \in A$ , a corresponding set of trajectories is generated through quintic and quartic polynomials, denoted as

$$fplist = Polynomial(cs_0, cd_0 \mid \Delta t \in cT, \Delta d \in cD, a) \quad (4)$$

By minimizing the cost function consistent with Bellman's principle of optimality, the best path  $bestpath \in fplist$  is selected, as visualized in Fig. 3. This path represents the optimal trade-off between jerk and time. Finally, by converting  $bestpath$  back into Cartesian coordinates, the child node  $s_1^{i+1}$  is computed from  $s_1^i$ .



**Figure 3:** Illustration of generating planning set for the ego vehicle. (a) Projection of the ego vehicle state onto the reference lane to obtain the Frenét state. (b) Generation of candidate trajectories. (c) Mapping the selected trajectory.

### 3.3 Other Agents Transition

Independently calculating the future trajectories of other agents would neglect the influence of ego vehicle planning, as discussed in [11]. The updated branch is defined as

$$h_N^\tau = [s_{n=1,\dots,N}^T, s_{n=1,\dots,N}^{T+1}, \dots, s_{n=1,\dots,N}^{T+\tau}], \quad (5)$$

where the ego vehicle's state  $s_1^{T+\tau+1}$  has already been obtained as described in Section 3.2. The prediction model begins by encoding both the road polyline information  $M$  and the historical trajectory data  $h_N^\tau$  for all agents, denoted as  $E^\tau$ . This encoded representation  $E^\tau$  is then decoded into multi-modal trajectory predictions along with their associated likelihoods. Normally, when a new state  $s_{n=1,\dots,N}^{T+\tau}$  is added to the branch

$$h_N^{\tau-1} = [s_{n=1,\dots,N}^T, s_{n=1,\dots,N}^{T+1}, \dots, s_{n=1,\dots,N}^{T+\tau-1}], \quad (6)$$

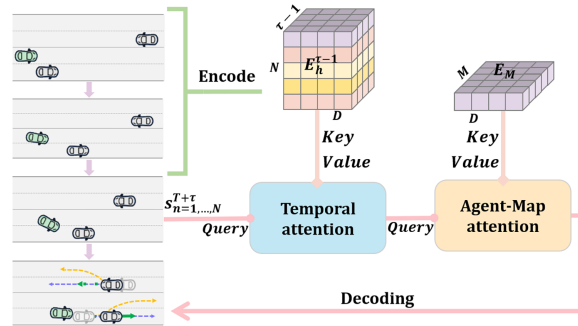
$h_N^\tau$  would be re-encoded as  $h_N^{\tau-1} \cup s_{n=1,\dots,N}^{T+\tau}$ . However, by transforming the Cartesian frame into a query-centric frame through relative spatial-temporal positioning, the historical trajectory  $h_N^{\tau-1}$  and map information  $M$  can be separately encoded into  $E_h^{\tau-1}$  and  $E_M$ , respectively. These components are then combined into a single encoded representation

$$E^{\tau-1} = \text{CrossAttention}(E_h^{\tau-1}, E_M). \quad (7)$$

In this framework, the new state  $s_{n=1,\dots,N}^{T+\tau}$  serves as the query for the attention mechanism, allowing  $E_h^{\tau-1}$  to be encoded temporally based on both  $E_h^{\tau-1}$  and  $s_{n=1,\dots,N}^{T+\tau}$ . Since the map embedding  $E_M$  remains constant throughout the scene, prior encodings can be efficiently reused, updating the representation as

$$E^\tau = E^{\tau-1} \oplus s_{n=1,\dots,N}^{T+\tau}. \quad (8)$$

As illustrated in Fig. 4, this method provides an efficient and accurate way to predict the future positions  $\left\{ \left( {}^k x^{\tau+1}, {}^k y^{\tau+1} \right), \dots, \left( {}^k x^{\tau+1+\tau'}, {}^k y^{\tau+1+\tau'} \right) \right\}_{k=1}^6$  with corresponding probability  $p_1, \dots, p_6$  at time  $\tau + 1$  and beyond for all agents. Importantly,  $\sum_{k=1}^6 p_k = 1$ , ensuring a probabilistic interpretation. Using these interactive trajectories, the future states  $s_{n=2,\dots,N}^{T+\tau+1}$  for the surrounding agents can be easily computed.



**Figure 4:** Illustration of updating other agents' states. The model encodes historical nodes along the branch as an agent embedding of dimension  $[N, \tau - 1, D]$  and a map embedding of dimension  $[M, D]$ , where  $D$  represents hidden layer encoding dimension. When a new node  $s_{n=1,\dots,N}^{T+\tau}$  expands, temporal attention is applied, encoding  $s_{n=1,\dots,N}^{T+\tau}$  as the query while using historical agent embeddings as keys and values. After decoding, the model generates  $k$  future trajectory candidates along with their associated probabilities.

The CrossAttention module is employed to model interactions among agents by encoding their historical trajectories and map information. This module utilizes a multi-head attention mechanism to capture the spatial-temporal dependencies between agents and the environment. Specifically, the module treats the relative state of each agent and the relative map polylines as queries, computing attention weights over the encoded historical data and map embeddings. This design not only facilitates efficient interaction modeling but also allows for real-time adaptation to changing scenarios.

### 3.4 Cost Function

After the expansion and rollout steps, the route's efficiency and safety must be considered comprehensively [44]. The overall cost function is a weighted linear combination of multiple components, denoted as

$$c = \omega_1 c^{(1)} + \omega_2 c^{(2)} + \omega_3 c^{(3)} + \omega_4 c^{(4)} + \omega_5 c^{(5)}, \quad (9)$$

where each term accounts for different aspects of the driving task, such as efficiency, comfort, and safety. To ensure that the ego vehicle moves forward as quickly as possible within the speed limit  $v_{max}$ , the efficiency cost is defined as

$$c^{(1)} = 1 - \left( \frac{v_\tau - v_{max}}{v_\tau} \right)^2, \quad (10)$$

where  $v_\tau$  represents the vehicle's velocity at time step  $\tau$ . To encourage smooth acceleration, abrupt changes in acceleration are penalized by setting the acceleration smoothness cost to

$$c^{(2)} = \sum_{t=T+1}^{\tau} (a_t - a_{t-1})^2, \quad (11)$$

where  $a_t$  is the acceleration at time step  $t$ . Similarly, smooth steering is encouraged by penalizing sharp changes in the steering angle,

$$c^{(3)} = \sum_{t=T+1}^{\tau} (\delta_t - \delta_{t-1})^2, \quad (12)$$

where  $\delta_t$  represents the steering angle at time step  $t$ . To avoid abrupt acceleration and braking, the hard acceleration and braking penalty is defined as

$$c^{(4)} = \sum_{t=T+1}^{\tau} (\ln(1 + \exp \kappa(a_t - \alpha)) + \ln(1 + \exp -\kappa(a_t - \beta))), \quad (13)$$

where  $\alpha = 4 \text{ m/s}^2$ ,  $\beta = -5 \text{ m/s}^2$  and  $\kappa = 15$ . To minimize the risk of collision with nearby vehicles, it is assumed that all agents are represented by rectangular shapes. The collision penalty is computed as

$$c^{(5)} = \sum_{t=T+1}^{\tau} [S(\lambda_x(\Delta x_t + l_x)) + S(\lambda_x(l_x - \Delta x_t))] \cdot [S(\lambda_y(\Delta y_t + l_y)) + S(\lambda_y(l_y - \Delta y_t))], \quad (14)$$

where  $l_x = 10.0 \text{ m}$  and  $l_y = 2.0 \text{ m}$  represent the minimum longitudinal and lateral distances between the ego vehicle and surrounding agents at time step  $t$ . The function  $S(\chi)$  is defined as

$$S(\chi) = \text{Sigmoid}(\chi) - \frac{1}{2} = \frac{1}{1 + \exp(-\chi)} - \frac{1}{2}, \quad (15)$$

where  $\lambda_x = 0.5$  and  $\lambda_y = 9.0$  control the longitudinal and lateral range for collision risk evaluation. The weights for the cost components are set empirically, based on their relative importance:  $\omega_1 = 1.0$ ,  $\omega_2 = -0.01$ ,  $\omega_3 = -1.5$ ,  $\omega_4 = -1.0$  and  $\omega_5 = -14.0$ .

## 4 Experiments

### 4.1 Dataset and Simulator

To comprehensively validate the effectiveness and generalization capability of the proposed method in real-world traffic environments, the representative Argoverse dataset series was selected for evaluation. These datasets are widely recognized in both academia and industry for their diverse urban scenarios, complex interactions, and high-definition map constraints, providing a rigorous foundation for trajectory prediction and behavioral modeling.

Collected via a multi-sensor platform, the Argoverse 1 Dataset [45] provides synchronized LiDAR, camera, and pose data, alongside vector maps containing lane centerlines and topology. Crucially, the dataset filters out trivial constant-velocity scenes, focusing instead on “interesting” scenarios—such as lane changes and turns—to challenge prediction models. It serves as a foundational benchmark with standard training, validation, and testing splits. Argoverse 2 Dataset [46] is a significant expansion of its predecessor, which enhances the scale, diversity, and complexity of the data. It spans a broader range of cities with distinct geographic and behavioral characteristics. This version introduces a richer taxonomy of tracked objects and improved map fidelity with detailed geometric and semantic information. With its extensive scene distribution and high-quality annotations, Argoverse 2 provides a more challenging testbed for learning map-constrained motion and interaction dynamics in high-fidelity virtual simulations.

To rigorously evaluate the decision-making and interaction capabilities of the proposed framework, closed-loop experiments were conducted within the MetaDrive simulator [47] utilizing real-world scenarios from the Argoverse 1 and Argoverse 2 datasets. MetaDrive serves as the core platform for constructing high-fidelity, interactive virtual engineering environments. Unlike traditional simulators such as CARLA [48] or AirSim [49] that prioritize visual rendering, MetaDrive focuses on physical precision and algorithmic generalization. Built upon the Panda3D engine and the Bullet physics engine [50], it ensures accurate vehicle dynamics.

### 4.2 Implementation Details

In terms of data configuration, training and evaluation environments were established using both the Argoverse 1 and Argoverse 2 datasets. Specifically, the model training phase utilized 208,272 trajectory prediction scenarios from Argoverse 1, where each scenario consists of a 1-s historical observation sequence and a 4-s future prediction sequence. Additionally, 199,908 scenarios from Argoverse 2 were employed, featuring 5-s historical observations and 6-s future sequences, to fully capture the motion evolution of traffic participants over continuous time scales. For the testing phase, 2000 scenarios were randomly selected from the official test sets of Argoverse 1 and Argoverse 2 together, to ensure the stability and representativeness of the quantitative evaluation. To ensure the robustness of the quantitative evaluation and mitigate the influence of random variance, all closed-loop testing scenarios were executed across 5 distinct random seeds. The final experimental results presented in this study represent the average performance across these 5 independent runs. Furthermore, a paired  $t$ -test was conducted to verify the statistical significance of the performance improvements, confirming a  $p$ -value of  $p < 0.05$ .

Regarding the planning module, a search-based motion planning framework was adopted. The planning horizon was set to 60 discrete time steps with a time interval of 0.1 s, corresponding to a total planning duration of 6 s. The tree search depth was configured to 6 layers, with each layer representing a 1-s decision span; this configuration strikes a balance between long-term planning capabilities and computational efficiency. The planner executes 100 search iterations per scenario to sufficiently explore the feasible solution space.

During the simulation process, dynamic agents within the traffic scene are permitted to interact and influence one another. The planner trained on Argoverse 1 integrates map constraints with motion predictions based on 1-s historical trajectories of surrounding agents and lane polygon information to generate an optimal 4-s future trajectory for the ego vehicle. Conversely, the planner trained on Argoverse 2 leverages 5-s historical trajectories and surrounding environmental information to generate potential 6-s future trajectories. This distinct configuration enables the effective modeling and evaluation of decision-making behaviors within complex traffic environments.

The Adam optimizer is employed for model training, running for 64 epochs with a learning rate of 0.0001. All experiments are conducted on a workstation equipped with an Intel Core i7-8700K CPU and eight NVIDIA RTX 4090 GPUs.

### 4.3 Comparison Results

**Metrics.** The performance of the planning module and the prediction module is evaluated using distinct sets of metrics. The planning-related metrics include the success rate, which indicates the percentage of scenarios in which the ego vehicle successfully completes its task; the collision rate, which represents the proportion of scenarios involving collisions; the traffic violation rate, which measures the proportion of scenarios involving infractions such as driving outside drivable areas; and comfort, which quantifies the smoothness of motion based on longitudinal jerk. For prediction performance, in addition to the minimum average displacement error (minADE) and the minimum final displacement error (minFDE), two probabilistic metrics are incorporated: the Miss Rate (MR) and the Negative Log-Likelihood (NLL). The MR denotes the ratio of scenarios where the endpoint of the best predicted trajectory deviates from the ground truth by more than a specific distance threshold 2.0 m reflecting the model's tendency to produce significant failures. The NLL measures the quality of the predicted probability distribution by quantifying the likelihood assigned to the ground truth trajectory, thereby evaluating the model's uncertainty estimation and confidence calibration. Furthermore, to assess the computational efficiency and real-time applicability of the framework, the inference time is measured, which quantifies the average latency required to complete a full prediction and planning cycle.

**Baselines.** To rigorously evaluate the effectiveness and advancement of the proposed framework in constructing a high-fidelity, interactive simulation environment for autonomous driving, a benchmark is established against a comprehensive set of state-of-the-art methods spanning the last decade. For trajectory prediction, the proposed approach is compared with four distinct paradigms: graph-based methods like VectorNet [51] and LaneGCN [52], which excel in explicit map topology modeling; Transformer-based architectures such as AgentFormer [53], HiVT [54], and Wayformer [55], selected for their superior ability to capture long-range spatiotemporal dependencies; joint probabilistic models like FJMP [56] and CoMTP [57], which emphasize game-theoretic multi-agent interactions; and self-supervised representation learning methods like Forecast-MAE [58], which demonstrate robust generalization via masked autoencoding. This selection allows for the systematic validation of the model's performance against varying strategies of environmental encoding and future inference.

In parallel, to assess the behavioral decision-making capabilities, five representative planning paradigms ranging from classical constraints to data-driven reasoning are referenced. Purely learning-based strategies

are represented by Urban Driver [59] and Learning by Cheating [60], while the emerging field of generative planning is evaluated through diffusion-based models like Diffusion-based Planner [61] and Diffusion-Drive [62], which utilize denoising processes to model multi-modal distributions. Large Language Model (LLM)-driven approaches, such as AsyncDriver [63] and DriveGPT4 [64], are also included to benchmark high-level semantic reasoning capabilities. These advanced methods are contrasted with the industry-standard, rule-based Intelligent Driver Model (IDM) [65] for baseline stability, and hybrid frameworks like GameFormer [15] that integrate game-theoretic priors with deep learning. By comparing against this broad spectrum—from explicit physical rules to implicit generative AI—the objective is to demonstrate the framework’s unique advantages in facilitating safe and intelligent decision-making within complex virtual engineering systems.

**Results and analysis.** Table 1 presents a quantitative comparison of the proposed framework against state-of-the-art trajectory prediction methods on the Argoverse benchmark. First, in terms of prediction accuracy, the proposed method achieves the lowest minADE of 0.734 and minFDE of 1.13 m. Compared to early graph-based baselines like VectorNet [51] and LaneGCN [52], the proposed framework reduces the average displacement error by a significant margin. This indicates that the query-driven latent representation captures the fine-grained topological constraints of the road network more effectively than explicit graph convolutions. Furthermore, the proposed method outperforms advanced Transformer-based architectures, including HiVT [54] and Wayformer [55]. It is observed that while the absolute numerical gains in average displacement metrics are modest—largely because these baselines are already highly optimized for open-loop, passive forecasting—the proposed interactive framework yields substantial and significant improvements in probabilistic safety metrics. Crucially, in terms of reliability and uncertainty estimation, the proposed method achieves the best performance with Miss Rate and Negative Log-Likelihood. The substantial reduction in MR compared to generative methods like Co-MTP [57] suggests that the proposed framework is particularly robust in handling long-tail scenarios. By explicitly simulating the bidirectional interactions between agents rather than treating them as static obstacles, the proposed approach effectively anticipates sudden maneuvers that other models might miss. The lowest NLL score further confirms that the probability distribution generated by the proposed model is highly calibrated, providing a trustworthy foundation for downstream decision-making.

**Table 1:** Quantitative comparison of trajectory prediction performance against state-of-the-art methods on the Argoverse benchmark. Bold values denote the best performance for each metric.

Method	minADE (m) ↓	minFDE (m) ↓	MR (%) ↓	NLL ↓
VectorNet [51]	1.87	3.33	32.15	5.60
LaneGCN [52]	0.89	1.38	17.11	2.96
AgentFormer [53]	1.46	2.88	24.00	5.01
HiVT [54]	0.75	1.16	10.96	2.55
Wayformer [55]	0.79	1.17	12.36	2.67
Forecast-MAE [58]	0.75	1.81	17.50	2.80
Fjmp [56]	0.80	1.93	18.71	2.88
Co-MTP [57]	0.76	1.15	17.65	2.74
PS	<b>0.73</b>	<b>1.13</b>	<b>8.35</b>	<b>2.27</b>

Table 2 evaluates the closed-loop planning capabilities of the proposed framework in complex dynamic environments. The results highlight the system’s ability to balance safety, efficiency, and comfort, surpassing

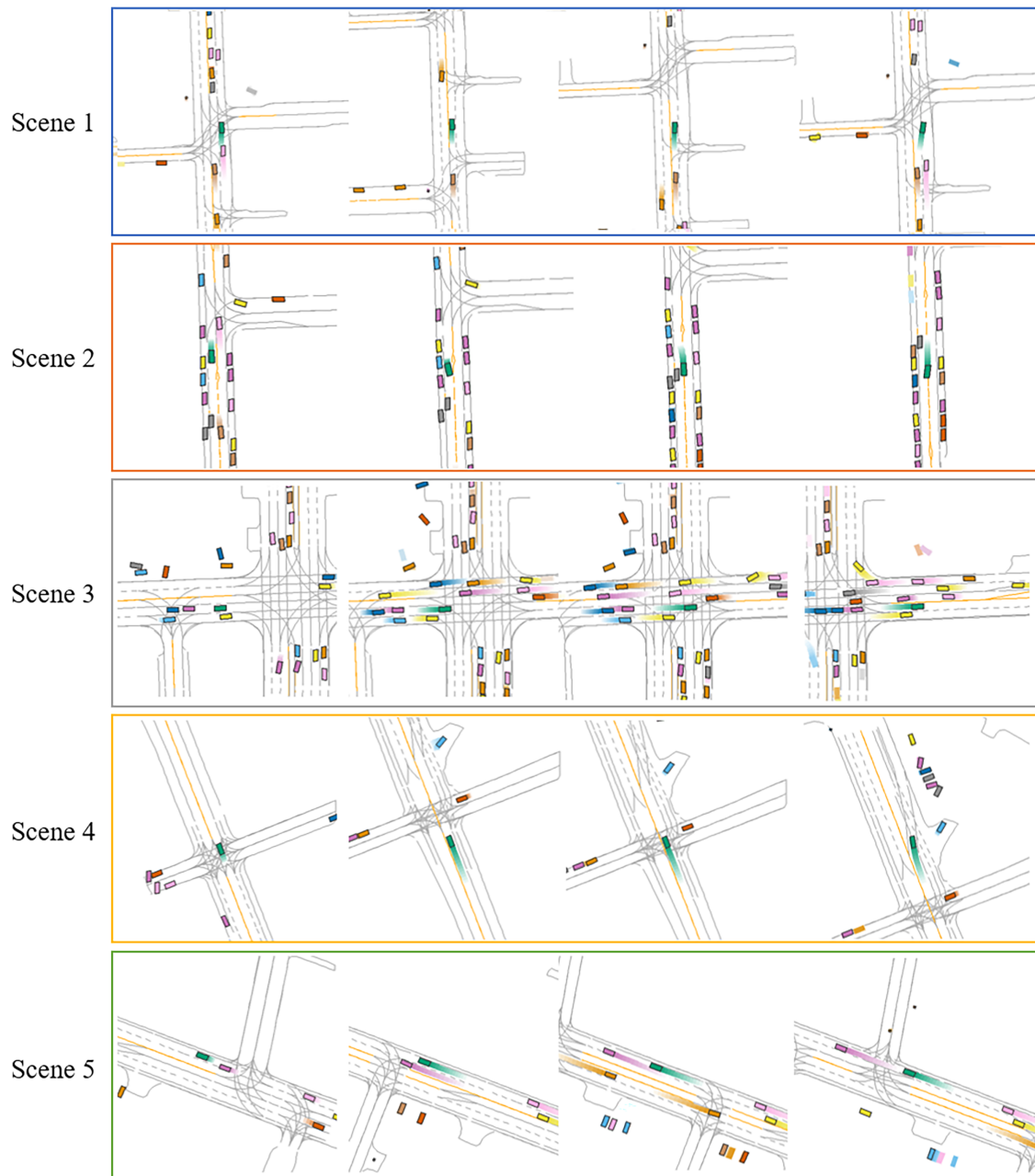
both learning-based and rule-based baselines. The proposed framework achieves the lowest Collision Rate and Violation Rate among all compared methods. In contrast, pure learning-based methods like Urban Driver [59] exhibit higher collision rates, often due to covariate shift issues where the model fails to generalize to unseen interaction states. Similarly, while LLM-driven methods like AsyncDriver [63] show a high success rate, the proposed method matches this efficiency while offering superior safety precision. This validates that the planning by simulation strategy acts as a safety shield, allowing the ego vehicle to foresee potential conflicts in the virtual world and adjust its trajectory before they occur in reality. Notably, the proposed method achieves the highest Comfort score. While diffusion-based planners are known for generating smooth, multi-modal trajectories, the proposed framework outperforms them by leveraging MCTS to explicitly optimize for smoothness constraints within the search tree. Compared to the rule-based IDM, which often produces jerky braking in dense traffic, the proposed approach generates human-like speed profiles. When compared to GameFormer, which also models interactions, the proposed framework shows a slight but consistent improvement across all metrics. This suggests that performing rollout simulations in a latent physical state space is more effective than explicit game-theoretic constraint modeling for handling high-dimensional traffic scenarios. While the absolute numerical margins in terms of inference time may appear relatively large, PS consistently outperforms a strong joint-planning baseline in critical areas such as success rate and collision rate. This consistent performance, especially in safety-critical metrics, is significant. Even small fractional improvements in success rate and collision avoidance across large-scale evaluations represent substantial enhancements in overall system reliability. The higher inference time of PS is a trade-off for these improvements in safety and decision-making robustness.

**Table 2:** Quantitative evaluation of closed-loop behavioral decision-making performance in the MetaDrive simulator. Bold values denote the best performance for each metric.

Method	Inference Time (ms) ↓	Success Rate (%) ↑	Collision Rate (%) ↓	Violation Rate (%) ↓	Comfort ↑
Urban Driver [59]	152.42	95.90	2.15	1.93	86.77
Learning by Cheating [60]	194.76	95.11	2.28	2.68	85.78
Diffusion-Based Planner [61]	52.74	96.22	1.33	2.17	88.99
DiffusionDrive [62]	<b>15.20</b>	96.58	2.03	2.20	90.27
AsyncDriver [63]	264.96	97.10	1.65	1.25	83.64
DriveGPT4 [64]	216.00	96.70	1.34	1.97	89.83
IDM [65]	129.73	94.87	1.39	3.74	83.64
GameFormer [15]	169.81	97.01	1.34	2.15	84.50
PS	403.23	<b>97.15</b>	<b>1.16</b>	<b>1.22</b>	<b>90.92</b>

#### 4.4 Qualitative Analysis

To intuitively demonstrate the effectiveness of the proposed decision-making model within a high-fidelity closed-loop environment, and to validate the dynamic planning capabilities of the behavioral decision module under strong interaction traffic flows, this section selects five representative dynamic scenarios for qualitative analysis. As shown in Fig. 5, the green box in the center represents the ego vehicle, while boxes of other colors represent background vehicles in the virtual environment. The trailing lines behind the vehicles illustrate their trajectories during the closed-loop simulation. A detailed review of these five typical scenarios clearly reveals the intelligent performance of the proposed algorithm in handling lane-change games, obstacle avoidance, congested traffic navigation, and intersection interactions.



**Figure 5:** Qualitative analysis of the ego vehicle's behavior in five representative dynamic scenarios within the closed-loop simulation. The green box indicates the ego vehicle, while other colors represent background agents.

Scenario 1 illustrates a lane-changing decision process involving a rear vehicle in the target lane. In this closed-loop test, the ego vehicle is required not only to plan a smooth lane-change trajectory but also to calculate the relative speed and safety distance with the approaching vehicle in real-time. The results indicate that the proposed model possesses a keen gap-capturing capability. Instead of adopting a conservative waiting strategy due to the presence of the rear vehicle, the ego vehicle accurately assesses that the rear vehicle's speed is within a controllable range and decisively yet smoothly cuts into the right lane. Throughout the process, the ego vehicle maintains a safe time headway via refined speed adjustments, completing the task without forcing

the rear vehicle to brake urgently. This demonstrates the model's profound understanding of right-of-way negotiation and precise control over safety boundaries.

Scenario 2 depicts the response mechanism when encountering congestion or stationary vehicles ahead. In this scenario, the perception and prediction modules successfully identify that the leading vehicle is static and determine that continued car-following would degrade traffic efficiency. Based on this judgment, the decision module generates a proactive lane-change command. As observed in the trajectory plot, the ego vehicle does not wait until it is critically close to the obstacle to make a hasty turn. Instead, it pre-plans a comfortable trajectory with a large curvature radius to merge smoothly into the adjacent free-flow lane. This proves that the proposed closed-loop framework possesses excellent predictive foresight, capable of translating semantic understanding of the virtual environment into efficient avoidance maneuvers, thereby preventing unnecessary stops.

Scenario 3 simulates an extremely complex urban congested intersection, characterized by irregular vehicle movements and potential intruders from all directions. In this high-dimensional state space, the ego vehicle must perform a straight-line crossing. The results show that despite the chaotic environment and minimal inter-vehicle spacing, the ego vehicle maintains a stable driving posture. By jointly predicting the intentions of surrounding multi-agents, the model successfully anticipates cut-in intentions from lateral vehicles and the stop-and-go rhythm of leading vehicles. The ego vehicle exhibits human-like car-following skills in closed-loop control, closely following the preceding trajectory to prevent frequent cut-ins by surrounding vehicles while ensuring zero collisions, thus validating the algorithm's robustness in unstructured, high-density scenarios.

Scenario 4 examines the interaction capability with crossing traffic flows at an intersection, a typical conflict scenario during unprotected turns or signal phase transitions. In the figure, the ego vehicle encounters a blockade by lateral traffic upon entering the intersection. Rather than mechanically executing an emergency stop, the model performs a dynamic risk assessment based on the velocity vectors of the lateral vehicles. Upon confirming that the lateral vehicles have passed, the ego vehicle swiftly utilizes the time gap to traverse the conflict zone. This ability to find feasible solutions during dynamic interaction indicates that the behavioral decision module is not a mere compilation of rules but possesses dynamic planning capabilities to handle complex spatiotemporal constraints, effectively balancing efficiency and safety.

Scenario 5 demonstrates decision stability when the ego vehicle is the subject of an overtaking maneuver. In this scenario, a rear vehicle initiates a rapid overtaking action. For a closed-loop control system, the close-range cut-in of an external agent imposes perception pressure and potential path interference. However, the visualization results show that the model exhibits a high level of "social" driving capability. Upon recognizing the overtaking intention of the rear vehicle, the ego vehicle maintains stability in speed and heading within its current lane, avoiding panic-induced evasive maneuvers or acceleration that might block the other driver, thus leaving ample space for a safe overtake. This stability during passive interaction is crucial for constructing a safe and harmonious mixed traffic flow and further corroborates the model's adaptability to dynamic environmental changes.

In summary, the qualitative analysis of these five typical scenarios provides a comprehensive and multi-dimensional validation of the proposed framework in closed-loop dynamic environments. Whether handling right-of-way games and lane keeping at high speeds, or executing proactive avoidance and interaction at complex urban intersections, the model demonstrates flexibility and environmental adaptability that surpass traditional rule-based methods. Notably, throughout the entire simulation process, the ego vehicle not only strictly adheres to static environmental rules but also exhibits human-like driving characteristics that prioritize safety while accounting for efficiency during continuous interactions with dynamic traffic participants. This ability to maintain decision consistency and trajectory smoothness in long-horizon,

strong-interaction scenarios strongly proves that the proposed method effectively resolves the decision-making rigidity often found in unstructured environments, providing solid algorithmic support for realizing truly safe, comfortable, and efficient autonomous driving.

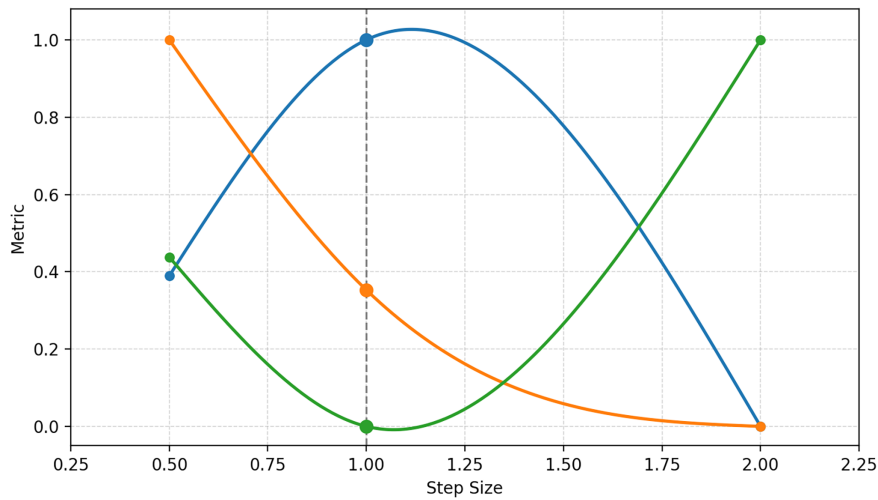
#### 4.5 Ablation Study

To highlight the concept of latent trajectory modeling, a rule-based prediction approach utilizing a constant-velocity, lane-following strategy is employed, referred to as Rule in the Table 3. To demonstrate the importance of cyclic interaction between prediction and planning, an experiment is conducted in which prediction is inferred only once for each action, denoted as Non-iter. In the third ablation study, the adaptive action generation is replaced with a fixed action set, labeled as Fix in the table. The quantitative comparison reveals that the full PS framework significantly outperforms all ablated variants, confirming that each component is indispensable for high-fidelity decision-making. The Rule variant exhibits the worst performance, with a drastically high Collision Rate of 12.10%. This indicates that simple kinematic extrapolation fails to capture complex social interactions and multi-modal intentions in dynamic traffic, proving that a learned high-fidelity predictive representation is a prerequisite for safety. The Non-iter approach shows a notable degradation in prediction accuracy and a lower Success Rate. This underscores that separating prediction from planning ignores the ego vehicle's influence on the scene. The superior performance of the proposed method confirms that modeling the “game-theoretic” recurrence is vital for accurate future deduction. The Fixed variant yields a sub-optimal Success Rate. This suggests that a static action space struggles to find optimal solutions in high-dimensional scenarios. The adaptive sampling strategy allows the planner to fine-tune trajectories within the continuous space, thereby achieving smoother control and higher task completion.

**Table 3:** Ablation study analyzing the impact of iterative interaction and adaptive action sampling on planning performance. Bold values denote the best performance for each metric.

Method	Success Rate (%) $\uparrow$	Collision Rate (%) $\downarrow$	Violation Rate (%) $\downarrow$	minADE ( $m$ ) $\downarrow$
Rule	89.33	12.1	11.29	1.33
Non-iter	90.27	2.04	9.06	1.69
Fixed	90.62	1.78	9.93	1.44
PS	<b>97.41</b>	<b>1.20</b>	<b>1.22</b>	<b>0.73</b>

To investigate the impact of action granularity, three target speed step sizes, 0.5, 1.0, and 2.0 m/s, were evaluated under a fixed MCTS iteration budget. As depicted in Fig. 6 the discretization step size critically dictates the balance between control resolution and computational complexity. At a fine granularity of 0.5 m/s, the inference time peaks because the reduced step size exponentially inflates the search tree's branching factor. Under a limited iteration budget, this combinatorial explosion traps the planner in shallow local optima, consequently decreasing the success rate and increasing collision risks. Conversely, a coarse granularity of 2.0 m/s minimizes inference time but causes the collision rate to surge and the success rate to plummet, as it deprives the ego vehicle of crucial micro-level speed adjustments needed for dense interactive traffic. The interpolated trends confirm that the 1.0 m/s step size strikes the optimal equilibrium—maximizing success and minimizing collisions while maintaining a tractable inference time, and is therefore adopted for the final framework.



**Figure 6:** Sensitivity analysis of action granularity. The graph illustrates the normalized performance metrics—Success Rate (blue), Inference Time (orange), and Collision Rate (green), across different target speed step sizes. Data points are normalized to a [0, 1] scale for comparative visualization and connected using cubic interpolation to demonstrate the continuous trend of the algorithmic trade-offs.

Table 4 summarizes the sensitivity analysis of the MCTS iteration budget, revealing a clear trade-off characterized by diminishing returns. At a restricted budget of 50 iterations, insufficient tree exploration leads to a high Collision Rate and a suboptimal Success Rate. Increasing the budget to the baseline of 100 iterations yields a substantial performance leap, significantly reducing collisions while maintaining a tractable inference time. Notably, further expanding to 150 or 200 iterations provides negligible safety benefits. Because the search largely converges around 100 iterations, additional rollouts merely inflate the inference time linearly through redundant exploration without meaningful performance gains. Consequently, 100 iterations serve as the optimal configuration, balancing decision robustness with computational efficiency.

**Table 4:** Sensitivity analysis of the MCTS iteration budget on closed-loop planning performance and computational inference time. Bold values denote the best performance for each metric.

Method	Success Rate (%)↑	Collision Rate (%)↓	Violation Rate (%)↓	Inference Time (ms) ↓
50	93.58	3.25	2.84	<b>298.54</b>
100	97.15	1.16	1.22	403.23
150	97.32	1.11	1.18	498.73
200	<b>97.38</b>	<b>1.09</b>	<b>1.16</b>	662.00

In conclusion, the ablation study validates that the integration of physical world simulation, iterative interactive reasoning, and adaptive search is essential for achieving the robustness and human-like driving capability demonstrated by our framework.

### 5 Conclusions

In this study, the Planning by Simulation framework was introduced, functioning as a high-fidelity simulation-based predictive planner for autonomous driving. It utilizes Monte Carlo Tree Search and a query-centric world model to bridge the gap between trajectory prediction and motion planning. By

iteratively reasoning about bidirectional interactions within a compact latent space, the proposed approach effectively addresses the limitations of decoupled architectures and the computational burdens of explicit trajectory generation. Comprehensive closed-loop evaluations on the Argoverse 1 and Argoverse 2 datasets within the MetaDrive simulator demonstrate that the proposed method significantly outperforms state-of-the-art baselines, achieving superior safety, efficiency, and human-like driving behaviors in complex dynamic scenarios. While these findings validate the potential of decision-centric virtual simulations, several limitations remain to be addressed for practical real-world deployment. First, the current unoptimized inference time falls short of the strict low-latency requirements demanded by physical vehicles, necessitating future industrial-level engineering optimizations such as TensorRT deployment and C++ acceleration. Second, the reliance on the Frenét coordinate system restricts the framework's applicability in completely unstructured environments devoid of high-definition map reference lines, suggesting the future need for hybrid Cartesian-based fallback planners. Third, the manually tuned empirical cost function is highly sensitive and lacks the adaptive robustness of dynamic industrial reward models. Finally, since the framework currently operates entirely within a virtual simulator, future research will focus on integrating large language models for enhanced semantic reasoning and deploying the algorithm onto physical platforms for Hardware-in-the-Loop validation. This critical step will ultimately bridge the gap between virtual testing and a fully synchronized real-world Digital Twin.

**Acknowledgement:** The authors would like to acknowledge Fudan University for providing the computational resources and environment required for this research. We also thank the technical staff of the Fudan Magic Lab for their professional assistance and technical support during the simulation and data analysis process.

**Funding Statement:** The authors received no specific funding for this study.

**Author Contributions:** The authors confirm contribution to the paper as follows: Conceptualization, Zhongxue Gan and Wenchao Ding; methodology, Tian Niu and Kaizhao Zhang; software, Zhongxue Gan; validation, Tian Niu and Kaizhao Zhang; formal analysis, Tian Niu and Kaizhao Zhang; investigation, Tian Niu, Kaizhao Zhang, Zhongxue Gan and Wenchao Ding; resources, Wenchao Ding; data curation, Tian Niu and Kaizhao Zhang; writing—original draft preparation, Tian Niu and Kaizhao Zhang; writing—review and editing, Tian Niu and Kaizhao Zhang; visualization, Tian Niu and Kaizhao Zhang; supervision, Tian Niu, Kaizhao Zhang, Zhongxue Gan and Wenchao Ding; project administration, Tian Niu, Kaizhao Zhang, Zhongxue Gan and Wenchao Ding; funding acquisition, Zhongxue Gan and Wenchao Ding. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** The data that support the findings of this study are openly available in <https://www.argoverse.org/>.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Guan Y, Ren Y, Sun Q, Li SE, Ma H, Duan J, et al. Integrated decision and control: toward interpretable and computationally efficient driving intelligence. *IEEE Trans Cybern.* 2022;53(2):859–73. doi:10.1109/tcyb.2022.3163816.
2. Brewitt C, Gyevar B, Garcin S, Albrecht SV. GRIT: fast, interpretable, and verifiable goal recognition with learned decision trees for autonomous driving. In: *Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; 2021 Sep 27–Oct 1; Prague, Czech Republic. p. 1023–30. doi:10.1109/iros51168.2021.9636279.

3. Shu Y, Zhou J, Zhang F. Safety-critical decision-making and control for autonomous vehicles with highest priority. In: Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV); 2023 Jun 4–7; Anchorage, AK, USA. p. 1–8. doi:10.1109/iv55152.2023.10186772.
4. Bachute MR, Subhedar JM. Autonomous driving architectures: insights of machine learning and deep learning algorithms. *Mache Learn Appl.* 2021;6(8):100164. doi:10.1016/j.mlwa.2021.100164.
5. Van Brummelen J, O'brien M, Gruyer D, Najjaran H. Autonomous vehicle perception: the technology of today and tomorrow. *Transp Res Part C Emerg Technol.* 2018;89(4):384–406. doi:10.1016/j.trc.2018.02.012.
6. Chen L, Wu P, Chitta K, Jaeger B, Geiger A, Li H. End-to-end autonomous driving: challenges and frontiers. *IEEE Trans Pattern Anal Mach Intell.* 2024;46(12):10164–83. doi:10.1109/tpami.2024.3435937.
7. Gu J, Hu C, Zhang T, Chen X, Wang Y, Wang Y, et al. Vip3D: end-to-end visual trajectory prediction via 3D agent queries. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 17–24; Vancouver, BC, Canada. p. 5496–506. doi:10.1109/cvpr52729.2023.00532.
8. Sun L, Jia X, Dragan AD. On complementing end-to-end human behavior predictors with planning. arXiv:2103.05661. 2021. doi:10.15607/rss.2021.xvii.037.
9. González D, Pérez J, Milanés V, Nashashibi F. A review of motion planning techniques for automated vehicles. *IEEE Trans Intell Transp Syst.* 2015;17(4):1135–45. doi:10.1109/tits.2015.2498841.
10. Hang P, Lv C, Huang C, Cai J, Hu Z, Xing Y. An integrated framework of decision making and motion planning for autonomous vehicles considering social behaviors. *IEEE Trans Veh Technol.* 2020;69(12):14458–69. doi:10.1109/tvt.2020.3040398.
11. Zhang K, Zhao L, Dong C, Wu L, Zheng L. AI-TP: attention-based interaction-aware trajectory prediction for autonomous driving. *IEEE Trans Intell Veh.* 2022;8(1):73–83. doi:10.1109/tiv.2022.3155236.
12. Liu K, Li N, Tseng HE, Kolmanovsky I, Girard A. Interaction-aware trajectory prediction and planning for autonomous vehicles in forced merge scenarios. *IEEE Trans Intell Transp Syst.* 2022;24(1):474–88. doi:10.1109/tits.2022.3216792.
13. Lei L, Luo R, Zheng R, Wang J, Zhang J, Qiu C, et al. Kb-tree: learnable and continuous monte-carlo tree search for autonomous driving planning. In: Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2021 Sep 27–Oct 1; Prague, Czech Republic. p. 4493–500. doi:10.1109/iros51168.2021.9636442.
14. Weingertner P, Ho M, Timofeev A, Aubert S, Pita-Gil G. Monte Carlo Tree search with reinforcement learning for motion planning. In: Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC); 2020 Sep 20–23; Rhodes, Greece. p. 1–7. doi:10.1109/itsc45102.2020.9294697.
15. Huang Z, Liu H, Lv C. Gameformer: game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In: Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision; 2023 Oct 1–6; Paris, France. p. 3903–13. doi:10.1109/iccv51070.2023.00361.
16. Shu Y, Zhou J, Zhang F. Agile decision-making and safety-critical motion planning for emergency autonomous vehicles. arXiv:2409.08665. 2024. doi:10.1109/tits.2025.3569416.
17. Xiao L, Liu JJ, Yang S, Li X, Ye X, Yang W, et al. Learning multiple probabilistic decisions from latent world model in autonomous driving. In: Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA); 2025 May 19–23; Atlanta, GA, USA. p. 1279–85. doi:10.1109/icra55743.2025.11127996.
18. Zheng Y, Yang P, Xing Z, Zhang Q, Zheng Y, Gao Y, et al. World4Drive: end-to-end autonomous driving via intention-aware physical latent world model. In: Proceedings of the 2025 IEEE/CVF International Conference on Computer Vision (ICCV); 2025 Oct 19–25; Honolulu, HI, USA. p. 28632–42.
19. Jia X, Sun L, Zhao H, Tomizuka M, Zhan W. Multi-agent trajectory prediction by combining egocentric and allocentric views. In: Proceedings of the 5th Conference on Robot Learning; 2021 Nov 8–11; London, UK. p. 1434–43.
20. Zhou Z, Wang J, Li YH, Huang YK. Query-centric trajectory prediction. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 17–24; Vancouver, BC, Canada. p. 17863–73. doi:10.1109/cvpr52729.2023.01713.

21. Zhang D, Wang G, Zhu R, Zhao J, Chen X, Zhang S, et al. SparseAD: sparse query-centric paradigm for efficient end-to-end autonomous driving. arXiv:2404.06892. 2024. doi:10.1109/tai.2025.3639457.
22. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, et al. Mastering the game of go without human knowledge. *Nature*. 2017;550(7676):354–9. doi:10.1038/nature24270.
23. Best G, Cliff OM, Patten T, Mettu RR, Fitch R. Dec-MCTS: decentralized planning for multi-robot active perception. *Int J Robot Res*. 2019;38(2–3):316–37.
24. Sukkar F, Best G, Yoo C, Fitch R. Multi-robot region-of-interest reconstruction with Dec-MCTS. In: 2019 International Conference on Robotics and Automation (ICRA); 2019 May 20–24; Montreal, QC, Canada. p. 9101–7. doi:10.1109/icra.2019.8793560.
25. Tian F, Wang FY, Li L. Enhancing feedback steering controllers for autonomous vehicles with deep Monte Carlo tree search. *IEEE Robot Autom Lett*. 2022;7(4):10438–45. doi:10.1109/lra.2022.3193452.
26. Albrecht SV, Brewitt C, Wilhelm J, Gjevvar B, Eiras F, Dobre M, et al. Interpretable goal-based prediction and planning for autonomous driving. In: 2021 IEEE International Conference on Robotics and Automation (ICRA); 2021 May 30–Jun 5; Xi'an, China. p. 1043–9. doi:10.1109/icra48506.2021.9560849.
27. Wen Q, Gong Z, Zhou L, Zhang Z. Monte-carlo tree search for behavior planning in autonomous driving. arXiv:2310.12075. 2023. doi:10.1109/ssrr62954.2024.10770028.
28. Zhang K, Yin D. Research on automatic driving path optimization algorithm based on Frenet coordinate system and Cartesian coordinate system. In: 2025 5th International Conference on Electronics, Circuits and Information Engineering (ECIE); 2025 May 23–25; Guangzhou, China. p. 835–8. doi:10.1109/ecie65947.2025.11087011.
29. Sarcinelli R, Guidolini R, Cardoso VB, Paixão TM, Berriel RF, Azevedo P, et al. Handling pedestrians in self-driving cars using image tracking and alternative path generation with Frenét frames. *Comput Graph*. 2019;84(4):173–84. doi:10.1016/j.cag.2019.08.004.
30. Ding W, Zhang L, Chen J, Shen S. Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor. *IEEE Robot Autom Lett*. 2019;4(3):2997–3004. doi:10.1109/lra.2019.2923954.
31. Ding W, Zhang L, Chen J, Shen S. EPSILON: an efficient planning system for automated vehicles in highly interactive environments. *IEEE Trans Robot*. 2021;38(2):1118–38. doi:10.1109/tro.2021.3104254.
32. Werling M, Ziegler J, Kammel S, Thrun S. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In: 2010 IEEE International Conference on Robotics and Automation; 2010 May 3–7; Anchorage, AK, USA. p. 987–93.
33. Raji A, Liniger A, Giove A, Toschi A, Musiu N, Morra D, et al. Motion planning and control for multi vehicle autonomous racing at high speeds. In: 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC); 2022 Oct 8–12; Macau, China. p. 2775–82. doi:10.1109/itsc55140.2022.9922239.
34. Ngiam J, Caine B, Vasudevan V, Zhang Z, Chiang HTL, Ling J, et al. Scene transformer: a unified architecture for predicting multiple agent trajectories. arXiv:2106.08417. 2021.
35. Cui H, Radosavljevic V, Chou FC, Lin TH, Nguyen T, Huang TK, et al. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In: 2019 International Conference on Robotics and Automation (ICRA); 2019 May 20–24; Montreal, QC, Canada. p. 2090–6. doi:10.1109/icra.2019.8793868.
36. Mercat J, Gilles T, El Zoghby N, Sandou G, Beauvois D, Gil GP. Multi-head attention for multi-modal joint vehicle motion forecasting. In: 2020 IEEE International Conference on Robotics and Automation (ICRA); 2020 May 31–Aug 31; Paris, France. p. 9638–44. doi:10.1109/icra40945.2020.9197340.
37. Hagedorn S, Hallgarten M, Stoll M, Condurache AP. The integration of prediction and planning in deep learning automated driving systems: a review. *IEEE Trans Intell Veh*. 2025;10(5):3626–43. doi:10.1109/tiv.2024.3459071.
38. Bahram M, Lawitzky A, Friedrichs J, Aeberhard M, Wollherr D. A game-theoretic approach to replanning-aware interactive scene prediction and planning. *IEEE Trans Veh Technol*. 2015;65(6):3981–92. doi:10.1109/tvt.2015.2508009.
39. Huang Z, Karkus P, Ivanovic B, Chen Y, Pavone M, Lv C. 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); 2024 May 13–17; Yokohama, Japan. p. 6806–12. doi:10.1109/icra57147.2024.10610550.

40. Wang X, Zhu Z, Huang G, Chen X, Zhu J, Lu J. Drivedreamer: towards real-world-drive world models for autonomous driving. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2024. p. 55–72. doi:10.1007/978-3-031-73195-2\_4.
41. Li Y, Fan L, He J, Wang Y, Chen Y, Zhang Z, et al. Enhancing end-to-end autonomous driving with latent world model. arXiv:2406.08481. 2024.
42. Yao J, Yang B, Wang X. Reconstruction vs. generation: taming optimization dilemma in latent diffusion models. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2025 Jun 10–17; Nashville, TN, USA. p. 15703–12. doi:10.1109/cvpr52734.2025.01464.
43. Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem. *Mach Learn.* 2002;47:235–56. doi:10.1023/a:1013689704352.
44. Dai Q, Xu X, Guo W, Huang S, Filev D. Towards a systematic computational framework for modeling multi-agent decision-making at micro level for smart vehicles in a smart world. *Robot Auton Syst.* 2021;144(4):103859. doi:10.1016/j.robot.2021.103859.
45. Chang MF, Lambert J, Sangkloy P, Singh J, Bak S, Hartnett A, et al. Argoverse: 3D tracking and forecasting with rich maps. In: Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019 Jun 15–20; Long Beach, CA, USA. p. 8748–57. doi:10.1109/cvpr.2019.00895.
46. Wilson B, Qi W, Agarwal T, Lambert J, Singh J, Khandelwal S, et al. Argoverse 2: next generation datasets for self-driving perception and forecasting. arXiv:2301.00493. 2023.
47. Li Q, Peng Z, Feng L, Zhang Q, Xue Z, Zhou B. Metadrive: composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Trans Pattern Anal Mach Intell.* 2022;45(3):3461–75. doi:10.1109/tpami.2022.3190471.
48. Dosovitskiy A, Ros G, Codevilla F, Lopez A, Koltun V. CARLA: an open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning; 2017 Nov 13–15; Mountain View, CA, USA. p. 1–16.
49. Shah S, Dey D, Lovett C, Kapoor A. Airsim: high-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics: Results of the 11th International Conference. Cham, Switzerland: Springer; 2017. p. 621–35. doi:10.1007/978-3-319-67361-5\_40.
50. He H, Zheng J, Sun Q, Li Z. Simulation of realistic particles with bullet physics engine. *E3S Web Conf.* 2019;92:14004. doi:10.1051/e3sconf/20199214004.
51. Gao J, Sun C, Zhao H, Shen Y, Anguelov D, Li C, et al. Vectornet: encoding HD maps and agent dynamics from vectorized representation. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 11525–33. doi:10.1109/cvpr42600.2020.01154.
52. Liang M, Yang B, Hu R, Chen Y, Liao R, Feng S, et al. Learning lane graph representations for motion forecasting. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2020. p. 541–56. doi:10.1007/978-3-030-58536-5\_32.
53. Yuan Y, Weng X, Ou Y, Kitani KM. Agentformer: agent-aware transformers for socio-temporal multi-agent forecasting. In: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision; 2021 Oct 10–17; Montreal, QC, Canada. p. 9813–23. doi:10.1109/iccv48922.2021.00967.
54. Zhou Z, Ye L, Wang J, Wu K, Lu K. HIVT: hierarchical vector transformer for multi-agent motion prediction. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. p. 8823–33. doi:10.1109/cvpr52688.2022.00862.
55. Nayakanti N, Al-Rfou R, Zhou A, Goel K, Refaat KS, Sapp B. Wayformer: motion forecasting via simple & efficient attention networks. In: 2023 IEEE International Conference on Robotics and Automation (ICRA); 2023 May 29–Jun 2; London, UK. p. 2980–7. doi:10.1109/icra48891.2023.10160609.
56. Rowe L, Ethier M, Dykhne EH, Czarnecki K. FJMP: factorized joint multi-agent motion prediction over learned directed acyclic interaction graphs. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 18–22; Vancouver, BC, Canada. p. 13745–55. doi:10.1109/cvpr52729.2023.01321.
57. Zhang X, Zhou Z, Wang Z, Ji Y, Huang Y, Chen H. Co-MTP: a cooperative trajectory prediction framework with multi-temporal fusion for autonomous driving. arXiv:2502.16589. 2025. doi:10.1109/icra55743.2025.11127303.

58. Cheng J, Mei X, Liu M. Forecast-MAE: self-supervised pre-training for motion forecasting with masked autoencoders. In: Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV); 2023 Oct 1–6; Paris, France. p. 8679–89. doi:10.1109/iccv51070.2023.00797.
59. Scheel O, Bergamini L, Wolczyk M, Osiński B, Ondruska P. Urban driver: learning to drive from real-world demonstrations using policy gradients. In: Proceedings of the 5th Conference on Robot Learning; 2022 Nov 8–11; London, UK. p. 718–28.
60. Chen D, Zhou B, Koltun V, Krähenbühl P. Learning by cheating. In: Proceedings of the 2020 Conference on Robot Learning; 2020 Nov 16–18; Virtual. p. 66–75.
61. Zheng Y, Liang R, Zheng K, Zheng J, Mao L, Li J, et al. Diffusion-based planning for autonomous driving with flexible guidance. In: ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy; 2025 Apr 24–28; Singapore.
62. Liao B, Chen S, Yin H, Jiang B, Wang C, Yan S, et al. Diffusiondrive: truncated diffusion model for end-to-end autonomous driving. In: Proceedings of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2025 Jun 10–17; Nashville, TN, USA. p. 12037–47. doi:10.1109/cvpr52734.2025.01124.
63. Chen Y, Ding ZH, Wang Z, Wang Y, Zhang L, Liu S. Asynchronous large language model enhanced planner for autonomous driving. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2024. p. 22–38. doi:10.1007/978-3-031-72764-1\_2.
64. Xu Z, Zhang Y, Xie E, Zhao Z, Guo Y, Wong KYK, et al. Drivegpt4: interpretable end-to-end autonomous driving via large language model. IEEE Robot Autom Lett. 2024;9(10):8186–93. doi:10.1109/lra.2024.3440097.
65. Treiber M, Kesting A. The intelligent driver model with stochasticity-new insights into traffic flow oscillations. Transp Res Procedia. 2017;23(4):174–87. doi:10.1016/j.trpro.2017.05.011.