A General Autonomous Driving Planner Adaptive to Scenario Characteristics

Xinyu Jiao 1, Zhong cao 2, Junjie Chen 2, Kun Jiang 2, and Diange Yang 2

1Tsinghua University
2Affiliation not available

October 30, 2023

Abstract

Autonomous vehicle requires a general planner for all possible scenarios. Existing researches design such a planner by a unified scenario description.

However, it may significantly increase the planner complexity even in some simple tasks, e.g., car following, further resulting in unsatisfactory driving performance. This work aims to design a general planner which can 1) drive in all possible scenarios and 2) have lower complexity in some common scenarios.

To this end, this work proposes a pertinent boundary for multi-scenario driving planning. The total approach is named as Pertinent Boundary-based Unified Decision system. Based on the original drivable area, the pertinent boundary can further support motion status and semantics of the traffic elements, which provides the potential of pertinent performance for given scenarios. The pertinent boundary can support unified driving with the drivable area, in the meantime, can be pertinently modified to support the pertinent driving decisions for identified driving scenarios (e.g., car-following, junction left turning). It will further avoid the bump between the connections of the scenarios due to the continuity of space boundary. Thus, the planner is suitable for the fully autonomous driving. The proposed method is validated in different classical driving decision scenarios. Results show that the proposed method can support pertinent driving decisions in identified scenarios, in the meantime, assure generalized cross-scenario planning when no scenario information is available. Such a method shed light on fully autonomous driving by pertinence improvement of multi-scenario decision in the complex real world.
A General Autonomous Driving Planner Adaptive to Scenario Characteristics

Xinyu Jiao, Zhong Cao, Junjie Chen, Kun Jiang, Diange Yang*

Abstract—Autonomous vehicle requires a general planner for all possible scenarios. Current research designs such a planner by a unified scenario description. However, it may significantly increase the planner complexity even in some simple tasks, e.g., car following, further resulting in unsatisfactory driving performance. This work aims to design a general planner which can 1) drive in all possible scenarios and 2) have lower complexity in some common scenarios. To this end, this work proposes a pertinent boundary for multi-scenario driving planning. The total approach is named as pertinent boundary-based unified decision (PBUD) system. Based on original drivable area, the pertinent boundary can further support motion status and other properties of the traffic elements, which provides the potential of pertinent performance for given scenarios. The pertinent boundary can support unified driving with the drivable area, in the meantime, can be pertinently modified to support pertinent driving decision for identified driving scenarios (e.g., car-following, junction left turning). It will further avoid the bump between the connections of the scenarios due to the continuity of space boundary. Thus, the planner is suitable for the fully autonomous driving. The proposed method is validated in different classical driving decision scenarios. Results show that the proposed method can support pertinent driving decision in identified scenarios, in the meantime, assure generalized cross-scenario planning when no scenario information is available. Such a method shed light on fully autonomous driving by pertinence improvement of multi-scenario decision in the complex real world.

Index Terms—Autonomous vehicle · Environment cognition · Multi-scenario driving decision

I. INTRODUCTION

An ideal fully autonomous driving planner should be able to deal with various scenarios on the roads, as well as corner cases without specific definitions. This is a big challenge for the driving decision system, which should make safe decision dealing with this high scenario complexity.

A classical idea is to define scenarios as many as possible and design the corresponding algorithms to drive. Since the DARPA challenges [1][2], such an idea is widely adopted in the industry, which is the mainstream of ADAS and L3 autonomous driving systems. Such a method can consider the characteristics of the scenarios for a well-tuned performance, e.g. car-following [3][4], overtaking [5][6], and junction-turning [7][8]. However, autonomous driving is not the parallel connection of individual scenarios. Even though some researches tried to solve these problems by setting transition zone [9] or adding transition strategy in finite state machine [1], the lack of generalization ability in unexpected scenarios and the increasing system complexity still limit its potential towards fully autonomous driving.

A unified decision system is another choice for fully autonomous driving. It usually consists of a unified environment representation and a corresponding planner. The methods can be classified into: 1) raw sensor data based planner; 2) drivable area based planner and 3) element-property template based planner. The brief introductions are as follows:

The raw sensor data input is the most generalized input form without any special designed processing, thus theoretically, the corresponding end-to-end decision systems [10][11] have the best generalization ability. However, directly aiming at end-to-end driving in all scenarios needs huge amount of data to converge in network training, therefore, these methods are still far from mature in real road driving applications.

Drivable area based planners are also highly generalized, modeling the whole free space in a unified data structure of grid map [12][13] or drivable area boundary [14][15]. In these approaches, the decision-making process is the same as the decision-making of robots in non-road environments. There are also attempts to apply this free-space driving framework by setting state-lattice in structural roads. Lim [16] applied the lattice trajectory planning scheme in multi-lane driving, thus generalized the decision problem into junction driving problem. However, the typical multi-lane driving behaviors (e.g., car following, taking over) cannot be accurately modeled by free space planning. Also, different on-road elements (e.g., vehicles, lanes, traffic lights) are not pertinently considered.

To better meet the requirements of structural road decision, some researchers tried to use element-property templates in unified driving decision. By listing the environment elements and their properties in a tree structure like OpenDRIVE [17] and NDS [18], the road environment can be represented in detail. The most applied template is the multi-lane driving template defined as lanes and on-lane vehicles to support typical lane-changing and lane-keeping decisions. For example, Chen et al. [19] achieved unified driving decision by setting virtual lanes in non-lane area (e.g., junctions). However, in non-lane scenarios, the multi-lane template might not generate the optimal decision. To cover non-lane scenarios, more and more environment elements have to be enumerated, resulting in high complexity in both environment representation and decision. Also, the generality and consistency are weakened compared to above free space models.

In general, current unified planner can adapt to the different scenarios, including unexpected corner scenarios. The price is the lack of awareness on the driving scenarios’ characteristics. This will dramatically increase the planner’s difficulty to han-
dle some simple and well-defined cases, and still not reaching the optimal solution. Thus, it is necessary to further consider such scenario pertinence into a unified planning framework.

This work aims at a unified planner with the ability to consider the scenarios’ characteristics, which is named as Pertinent Boundary-based Unified Decision system (PBUD). The main idea is to represent the environment by the pertinent boundary, with motion status and boundary type as the additional status of the drivable space boundary. The objective existence of space boundary ensures the generalization ability of PBUD, in the meantime, the additional status information provides the potential for pertinent consideration of different on-road elements. The boundary can further be adjusted for the current driving scenario, if it can be identified. We further designed a corresponding planner for this pertinent boundary. The contribution of this work lies in:

1) A novel scenario representation method using some semantic boundaries (named as pertinent boundary), which can represent the drivable area as well as the scenarios’ characteristics.

2) A general driving planner using the pertinent boundary which can drive in various scenarios without improving the planning complexity in simple scenarios.

3) An open-source code-base of the proposed planner adaptive to various sensor configurations.

The remainder of this paper is organized as follows. Section II will formally define the problem to be solved in this study, and introduce the research framework. Section III will analyze the generalization ability of PB, and Section IV will introduce boundary adaption and driving decision that can be pertinent to given scenario. Section V will provide experiment results in real world driving to prove the scenario pertinence of PBUD. Section VI will conclude this work and point out the future research direction.

II. PROBLEM DEFINITION AND MAIN IDEA

A. Problem definition

The problem that this paper aims to solve can be defined as: proposing an environment model that can: 1) support unified driving decision in different scenarios, and 2) be pertinently adapted to given scenario while still keeping the unified decision.

To achieve the first point, the environment model and corresponding planner should be general, i.e., be able to unconditionally work in any scenario. However, if the environment model is too simple, e.g., occupancy grid, it cannot reach high performance due to the lack of details. Thus, it is required to involve more details (i.e., more environment elements and properties) to improve universality, esp. in uncommon complex scenarios. However, this will bring unnecessary complexity in simple scenarios, such as car following, lane changing, etc.

In well-defined scenarios, the decision problem can be depicted with only a few input parameters, e.g., $\Delta s$, $\Delta v$ and $v_f$ in car following scenario [3][4]. The simplified decision problem makes it possible to use pertinent high-level decision to reduce complexity, and further help in reaching optimal solution. However, simply redefining the decision problem and abruptly changing the strategy will break the consistency of the unified planner. Thus, the second point lies in scenario-pertinent decision with consistent framework.

Fig. 1 summarizes the problem by illustrating the developing trend of unified decision and the scenario pertinent adaption process:

Step 1: Developing unified decision

Step 2: Scenario pertinent adaption

Fig. 1. Problem definition

In all, the aim of this study is improving the pertinence to given scenarios, on the condition of maintaining the decision unified. Abstractly, 3 questions need to be answered:

1) How to involve rich details in the environment model while assuring the ability of unconditional application?

2) Is the environment model flexible for scenario pertinent adaption and simplification while keeping the form unchanged?

3) How can driving decision be pertinent to given scenario with the flexible environment model, while maintaining the decision unified?

B. Main idea

In order to solve the above 3 questions, we combine two basic types of environment models of unified decision, i.e., drivable area boundary and element-property list.

The proposed pertinent boundary (PB) is created by loading semantic and dynamic information of the environment elements as the drivable area boundary status. Thus, the original space boundary obtained the ability to accommodate environment details without losing data consistency. Also, the objective existence of drivable area boundary can provide safety insurance for unexpected or unidentified scenario. Furthermore, the element information can support scenario pertinent driving decision. In this way, the advantage of consistent drivable area and abundant element-property list could be combined in PB.

For identified scenarios, the semantic and dynamic information on PB can support scenario pertinent upper-level decision. In fact, it is a current mainstream to apply two-stage
decision with upper-level planner (usually behavioral decision for limiting trajectory) and lower-level planner (for trajectory output to controller). In PBUD, the scenario pertinent upper-level decision plays the role of reducing the searching scale of lower-level decision without breaking the consistency ensured by PB. Thus, scenario-pertinent decision can be achieved while keeping the decision logic unified.

The PBUD could be symbolically illustrated as follows:

\[
B = \{ B_i \mid i = 1, \ldots, H \} 
\]

where \( B_i \) denotes the pertinent boundary at time step \( i \), \( H \) represents the planning horizon, i.e., maximum time steps, and \( B \) is the set of pertinent boundaries in the planning horizon. In (2), a PB is consist of \( n \) points with state \( (x^s, x^d, \eta) \), referring to static position, dynamic status and semantic information. In (3), \( U \) refers to the trajectory searching area, determined by upper-level planner \( f_u \) with PB, driving task \( k \), and identified scenario \( S \). In (4), the final trajectory is determined by lower-level planner \( f_l \) with PB, driving task and \( U \).

The general framework of the PBUD is shown in Fig. 2.

In the remainder of this paper, we will solve the 3 questions in problem definition with PBUD. In Section 3, the first question, which is on the ability of involving rich details, will be answered. Afterwards, Section 4 will answer the second and third questions, which are about the pertinence on scenarios with maintained planning unity.

III. ENRICHING THE ORIGINAL SPACE BOUNDARY: IMPROVING EXPRESSIVENESS

The pertinent boundary (PB) is based on the objective space boundary, and loaded with additional environment details as boundary status. In this section, we will introduce how to build PB which is enriched by environment element properties.

The original space boundary is mathematically in the form of vertex position coordinate list:

\[
B = \{ x, y \} 
\]

This form consistently expresses the world by generalizing all the spatial constraints (e.g., vehicles, cyclists, road boundary, red light, etc.) surrounding the ego vehicle. However, the detailed characteristics of different scenarios are lost, which limits its potential of application in autonomous driving, especially in structural roads. Therefore, environment details should be loaded on the space boundary to provide the essential information for scenario pertinent driving.

In order to make scenario-pertinent decision, the boundary needs to absorb the abundant and detailed information from the element-property list. Fundamentally, the semantic information (i.e., element type) should be absorbed as boundary forming source. Also, the dynamics of the elements should be recorded for safety consideration and collision avoidance, which further forms a temporal limit combining with prediction module. The pertinent boundary is shown as:

\[
S, D \rightarrow B
\]

\[
B = \{ x^s, x^d, \eta \}_{n} 
\]

where \( S, D \) refers to static and dynamic environment elements, \( B \) refers to the pertinent boundary. Equation (7) is the boundary representation, with \( n \) vertex points, each represented by static status \( x^s \), dynamic status \( x^d \) and semantic information \( \eta \). Finally, the tempo-spatial limit \( B \) is formed by a set of pertinent boundaries in the planning horizon \( H \).

The semantic information basically refers to the type, e.g., vehicles, traffic lanes, road boundary, etc. Additional information can also be considered, e.g., vehicle tail light status, door status, etc. This provides equal complexity to the environment model of element-property template, allowing detailed scenario-oriented strategies to be involved. In other words, the PB has equal ability with element-property list in representing complex world after the status extension, while the boundary consistency is not broken.

It should be noted that the indicating information, e.g., traffic lane, speed limit, influence ego vehicle by providing guidance or setting limit to the driving action. These are not
spatial driving limit, thus not covered in the PB. The influence of the indicating information on planning will be discussed in Section 4.

For PB construction, online detection from onboard sensors and priori knowledge from static map are the two main input. The onboard sensors can provide original space boundary, for example, LiDAR ground extraction [20] and drivable area segmentation with camera [21]. Also, detailed semantic information of obstacles can be obtained from object detection. The static map can offer the physical limits from road structure, along with virtual limits from traffic rule. The construction of the status extended space boundary is shown in Fig. 3.

\[ B = \{(r, \theta, v, \theta_v, \eta) \mid \theta \in [0, 2\pi)\} \]  

(9)

where \( B \) denotes the pertinent boundary in the polar coordinate, \( v \) refers to the speed value of the boundary, \( \theta_v \) is the direction of the speed vector, and \( \eta \) is the boundary type.

The position components, i.e., \( (r, \theta) \), represent the objectively existing space boundary. The other properties \( (v, \theta_v, \eta) \) can be correspondingly set according to the properties of the element that forms the boundary section. We tested the PB construction in Carla simulator [22], as shown in Fig. 4.

In Fig. 4, the color of boundary section indicates the type of the boundary, along with additional properties (e.g., vehicle behavior). The boundary velocity is also shown. From this we can see that the PB can be constructed from map, objects and drivable area detection, and can load more information than pure drivable area boundary, thus has the potential for scenario pertinent decision.

**IV. SCENARIO-PERTINENT PLANNING IN PBUD**

In this Section, we will answer the second and third questions in Section 2. We will analyze the flexibility of PB for pertinence improved application in different scenarios, and introduce how pertinent decision is made while keeping planner unified.

For clarity, we will formally define the concept of pertinence, and then give a brief introduction of decision framework. Then, we will instantiate the framework for clarity, application and validation. Boundary form consistency and planning unity will be checked along.

**A. The concept of pertinence**

Literally, pertinence refers to the ability to deal with the current matter in a dedicated manner. To give a formal definition of pertinence in driving decision, the pertinence of planning should be defined within the mathematical framework of driving decision. Generally, driving decision (both rule-based [16][19] and learning-based decision [23][24]) could be regarded as a constrained optimization problem (COP):

\[
\begin{align*}
\min & \ (C_f(T(A))) \\
\text{s.t.} & \ A \in \mathcal{A}, \mathcal{A} = \mathcal{A}_e \cap \mathcal{A}_s
\end{align*}
\]

(10)

where \( T \) is the final trajectory output, \( A \) refers to the trajectory parameter, \( C_f \) refers to the cost function, and \( \mathcal{A} \) denotes the feasible region in the parameter space, formed by the intersection of environment constrained region \( \mathcal{A}_e \) and the trajectory searching region \( \mathcal{A}_s \).

Within this framework, driving decision, pertinent planning could be defined as the ability to efficiently find the optimal driving decision within the current scenario:

**Definition 4.1:** Pertinent planning in a given scenario is the ability to improve the efficiency of correctly solving the planning optimization problem in (10).

From Definition 4.1, pertinent decision can be achieved by reducing the feasible region or accelerating the searching process, at the premise of not losing the optimal solution. In this paper, since we focus on the environment representation, we only talk about the reducing of feasible region. Since the environment constraint \( \mathcal{A}_e \) is an objective existence that
changes with the real traffic environment, only trajectory searching region $A_s$ could be set in prior, which represents the complexity of the decision process. In this sense, the decision pertinence $\chi$ could be mathematically expressed as:

$$
\chi \begin{cases} 
\propto 1/V(A_s), & T^* \in A_s \\
0, & T^* \notin A_s 
\end{cases}
$$

(11)

where $V(A_s)$ denotes the volume of trajectory searching region in the parameter space.

B. Planning framework in PBUD

1) General symbolic form: According to the main idea of this study, the PBUD framework should not only support general planning across different scenarios, but also improve pertinence in well-defined scenarios. Following the mainstream of two-stage driving decision with upper-level and lower-level planners [16][19], the planning framework of PBUD can be illustrated in Fig. 5:

![Fig. 5. Planning framework in PBUD](image)

In Fig. 5, the main stream (i.e., black arrow) works unconditionally in different scenarios. It is shown that the trajectory is generated according to $B$, $k$ and $U$. The roles of the three parts are shown below:

a) $B$: With PB as the tempo-spatial environment constraints, the trajectory should lie inside PB during the whole time horizon, i.e., $A_e$ is set by non-collision constraint to the PB time series $B$:

$$
A_e : \forall A, \quad \text{s.t.} \quad c(T(A), B) = 0
$$

(12)

where $c$ refers to the collision check between $T$ and $B$.

Generally, a trajectory $T$ is composed of $H$ trajectory points corresponding to the time steps in the planning horizon. Thus, in each time step, the trajectory point should lie inside the corresponding PB. The collision check becomes a geometric problem of checking whether a point lies in a closed polygon:

$$
\forall i, j, \quad \min \left( \text{dist} \left( e_j, \left( p_{ij}, p_{(i+1)j} \right) \right) \right) > d_{thres}
$$

(13)

As shown in Fig. 6(a), signed distance could be defined based on the positional relationship to the directed line segment. In Fig. 6(b), the PB is defined in counterclockwise order. If a point lies inside PB, the minimum signed distance $\text{dist}$ should be bigger than zero, i.e., the point is to the left side of the nearest boundary segment. In Fig. 6(c), the safety margin is introduced to endure the detection (current PB) or prediction (future PB) error, thus the minimum signed distance should be bigger than some positive safety margin $d_{thres}$. This process is repeated in each time step to provide safety constraints from $B$.

The safety margin $d_{thres}$ is settled according to the boundary status:

$$
d_{thres} = d_0 + k_1 v_{ego} + k_2 \eta_i v_i
$$

(14)

where $d_0$ denotes the size of ego vehicle, $v_{ego}$ refers to ego velocity, and $v_i$ is the speed of $i$th boundary section. $k_1$, $k_2$ are speed ratio parameters, $\eta_i$ is the boundary type ratio.

The safety margin reflects the risk assessment on PB, where different types of objects are considered in a consistent way, as shown in Fig. 7.

![Fig. 7. Consistent constraints in PB](image)

As shown in Fig. 7, the safety margins of different PB segments form the safety boundary together, summarizing the element properties into a consistent form. The dynamic boundary segments has higher safety margin, reflecting the higher risk they might generate.

b) $k$: Task guidance $k$ provides the optimization goal, i.e., cost function, in the COP of (10). Basically, the vehicle should proceed along the preset global guidance line. Besides,
customized optimization goal such as comfort [25], fuel saving [26], could also become parts of the driving task.

Generally, with the environment constraints and task guidance, the trajectory planner can generate an optimized, safe and legal trajectory.

c) $U$: In well-defined scenarios, dedicated upper-level decision could give some direction to the trajectory planner. In PBUD, this direction from upper-level decision is realized by setting an additional optimization constraint $A_c$. Cooperating with $B$, the feasible area of trajectory parameters is determined as the intersection of $A_c$ and $A_s$, as illustrated in (10). In this way, scenario-dedicated pertinent decision can merge into the main stream, sharing the same mechanism with $B$. In summary, the PBUD works unconditionally as a constrained optimization problem, with environment constraint $B$ and driving task $k$. Despite the scenario and element complexity, the decision problem can be uniformly abstracted into an optimization problem with geometric constraints, thanks to the consistency of PB. Furthermore, the optional upper-level planner can improve decision pertinence without breaking the consistency and universality of general PBUD.

2) Instantiation: In this part, we will instantiate the above abstract PBUD framework. Since the focus of our study does not lie in the planning optimization algorithm or developing any scenario-dedicated decision, we take a classical planner proposed by Werling et al. [25] for instantiation, defining trajectory parameter space, cost function and mechanism of constraints.

According to Fig. 5, the Werling planner acts as the trajectory planner, generating trajectory according to input $A_c$, $A_s$ and $C_f$. First, candidate trajectories are sampled according to $A_s$, then candidate trajectories must pass the check of environment constraints $A_c$. Finally, the best trajectory selected by the cost function $C_f$ will be the final output.

It should be noted that other solvers to the COP of (10), including analytical solvers and sample-based solvers, could be applied in the place of the Werling planner. In sample-based solution, with fixed sampling step, the number of sampled trajectories $n(A_s)$ is proportional to the volume of $A_s$, which is therefore an objective measurement of pertinence $\chi$, as shown in (11).

a) Parameter space

Werling planner is carried out in the lane Frenet coordinate, with $s$ axis set along the lane center line, indicating the longitudinal position, and $d$ axis set vertical to $s$, indicating the lateral position. This coordinate is similar to $s-t$ coordinate in OPENDRIVE [17].

Trajectory sampling in the Werling planner is through the end state sampling. With a given end state $\tau_e$, a candidate trajectory $T_e$ will be generated as a quintic polynomial to connect start state and end state with the optimal cost function:

$$T_s = \arg\min_{T} C_f (T (\tau_0, \tau_e))$$  \hfill (15)

$$\tau_e = (d_e, T_e, v_e)$$  \hfill (16)

where $\tau_0$ and $\tau_e$ denote the start state (i.e., current vehicle state) and end state respectively, whereas $d_e$, $T_e$, $v_e$ denote the lateral decision of the end state, temporal duration of the trajectory and the speed of the end state respectively.

Since a trajectory is determined by the end state triplet $(d_e, T_e, v_e)$, the parameter space is settled as the three-dimensional space with axis of $d_e$, $T_e$, and $v_e$.

b) Environment constraint $A_c$:

$A_c$ provides the safety constraints in PBUD. With Werling planner, (12-14) work as collision checker, the candidate trajectories that does not satisfy the environment constraint $A_c$ are disqualified.

c) Trajectory sampling constraint $A_s$

$A_s$ is the trajectory searching region in the parameter space. In Werling planner, $A_s$ is set as:

$$d_e \in [d_L, d_H]$$  \hfill (17)

$$T_e \in [T_L, T_H]$$  \hfill (18)

$$v_e \in [0, v_H]$$  \hfill (19)

where $d_L$ and $d_H$ are the lateral limit of end state sampling, $T_L, T_H$ are the least and most planning horizon, and $v_H$ is the highest sampled speed.

Note that $T_L$ and $T_H$ tuning is based on the control requirements and system temporal properties, thus is fixed in the planner. Thus, when the scenario is well-defined, the task of improving pertinence lies in limiting the sampling range of $d_e$ and $v_e$, while not losing the optimal solution. In undefined scenarios, $A_s$ is set as the maximum sampling range.

d) Cost function $C_f$

The cost function in the planner is set as:

$$C_f = k_3 J_t + k_t T_e + k_d d_e^2 + k_v (v_e - v_{des})^2$$  \hfill (20)

In cost function 20, $k_3$, $k_t$, $k_d$ and $k_v$ are the weights, $v_{des}$ is the desired speed from the upper-level planner (if any). When there is no scenario-dedicated upper-level planner, the last term is set to 0. Note that this cost function is slightly different from the original Werling planner.

With the cost function, each sampled candidate trajectory is labeled with its $C_f$ value. The final output trajectory is the candidate trajectory with the lowest $C_f$, among the candidate trajectories that passed the collision check of $A_s$.

In the next section, we will examine the performance of PBUD in different types of scenarios with the above instantiation.

V. EXPERIMENTAL RESULT

In this section, the proposed PBUD is tested in some typical driving scenarios. In each type of scenario, we will first examine the performance of PBUD, esp. the mechanism of pertinence improvement by involving upper-level planner for this type of scenario. The framework consistency will be checked along. Then, comparative analysis will be carried out to validate PBUD in pertinence improvement.

Three classical unified decision systems are taken as benchmarks, namely SUD, LaneUD and BUD. The SUD is a unified sample-based driving method set in the element-list environment model, regarding the entire autonomous driving process
as collision avoidance against different types of elements. The LaneUD is the extended multi-lane driving method, with multi-lane driving strategy IDM taken as upper-level planner in all the scenarios. The BUD is also a unified sample-based planner, different from SUD, it applies driving space boundary instead of the element list as the environment model.

The cases are set in Carla simulator [22]. Multi-lane scenarios, junction scenarios and an undefined scenario are inspected. Note that other custom scenario-dedicated decisions, e.g., railway crossing, ramp merging, could also be involved in PBUD with the same mechanism.

According to the preliminaries stated in Section 2, perception and control system are required for the test of the proposed PBUD. Here we take ground truth of map and vehicles from the simulator as the perception input. In addition, for off-road driving, a 32-line LiDAR will detect the drivable area boundary. Also, a PID controller is applied to follow the planned trajectory.

A. In-lane scenario

1) PBUD for in-lane scenarios: In-lane driving is the most common type of urban and highway driving. To be pertinent to in-lane scenarios, the PB should support the upper-level planner in making lane-changing and lane-keeping decision, thus limiting the scale of sampling process.

Fig. 8 is the flowchart of PBUD for in-lane driving with a standard in-lane scenario:

\[ R = \cup \{ R_l, \ l \in G(M, k) \} \]
\[ B = \partial R \]

where \( l \) refers to a traffic lane, \( G \) is the set of guidance lines derived from map and driving task, and \( R_l \) is the free section on the lane.

From this case, we can see that the PB could be modified according to scenario requirement as long as the data form is not changed. The physical shape, the extra information on semantic and motion status should still be in the extended status of the boundary vertexes.

Thanks to the extended status in PB, the upper-level planner could find required information from PB. In this example, to make lane-following decision, the front status (i.e., longitudinal position and velocity) \{sf, vfsf\} and the back status \{sb, vsb\} in the lane are required. In PBUD, these can be equivalently obtained by the extended boundary status of the front and back intersecting points of the center line and boundary, as shown in Fig. 8.

Then, lateral decision of desired lane index (lane-changing) and longitudinal planning of desired speed (lane-following) could be made:

\[ U = \{ g, v_{des} \} = f_u(B, M) \]
\[ g = \arg\max_l (R(l)), \ l \in G(M) \]

where \( \{ g, v_{des} \} \) is the final output of upper-level planner, i.e., guidance line and desired speed. \( G \) is the set of lane center lines derived from map \( M \). \( l \) is a lane center line, and \( R_l \) denotes its score, which could be derived from the car-following strategy.

Here we take intelligent driver model (IDM) [3] as the lane-following strategy. IDM model in PBUD is shown as below:

\[ a_i = m \left[ 1 - \left( \frac{v_{ego}}{v_{free}} \right)^\delta - \left( \frac{\Delta s_0 + T v_{ego} + \frac{v_{ego}(v_{2i} - v_{ego})}{2s_2}}{s_{2i} - s_{ego}} \right)^2 \right] \]

where \( a_i \) refers to the planned longitudinal acceleration, and \( v_i \) denotes the planned velocity in the \( i \)th lane; \( s_{ego} \) and \( v_{ego} \) are the longitudinal position and velocity of ego vehicle; \( v_{free} \) means the aim free flow velocity; \( \Delta s_0 \) refers to the actual gap between the ego vehicle and its front vehicle. \( g_0, T, \delta, m \) and \( n \) are desired gap parameters. \( \Delta t \) is the decision time step length.

To make lane-changing decision, the lane with the best score will be chosen. Here we take available speed, turning request and ego lane compensation to calculate the score:

\[ U (l_i) = u_{speed} v_i + \frac{u_{task}}{r_i} \Delta n_i + u_{ego} \]
\[ i_{des} = \arg\max_i (U (l_i)) \]
\[ g = l_{i_{des}} \]
\[ v_{des} = v_{i_{des}} \]
where \( u_{\text{speed}} \), \( u_{\text{task}} \) and \( u_{\text{ego}} \) are weights of the corresponding terms, \( \Delta n_i \) denotes the index, \( r_i \) is the remaining longitudinal distance from current position to the required turning point (e.g., intersection), and \( i_{\text{des}} \) refers to the index of the desired lane.

Note that other custom lane-changing strategies are also possible.

Finally, the trajectory searching region \( A_s \) is set as:

\[
d_e \in \left[ -\frac{w_l}{2} + d_{\text{des}}, \frac{w_l}{2} + d_{\text{des}} \right]
\]

\[
v_e \in [0, v_{\text{des}}]
\]

where \( d_{\text{des}} \) is the lateral distance of desired lane \( g \), and \( w_l \) denotes the lane width.

With the upper-level decision, the end state sampling only needs to be carried out in the desired lane, and the end speed \( v_e \) only needs to be sampled below the desired speed. Thus, the volume of \( A_s \) is reduced without losing optimal solution, i.e., pertinence is improved. In addition, the involving of \( A_s \) does not break the consistency of PBUD main stream.

Fig. 9 further shows the performance of PBUD in some other in-lane scenarios:

In (a), lane-changing decision is made by upper-level planner due to higher utility of adjacent lane. Sampling is carried out in the aimed lane by setting end state in the lane.

In (b), there are 3 types of surrounding vehicles. As for the lane-keeping vehicles, same adaption method is adopted as that in Fig. 8. As for the roadside stopping vehicle, the vehicle occupies part of the lane space laterally. If the remaining width is less than half lane width, the occupied lane section is regarded as blocked, thus can be adapted as in Fig. 8. Else, the vehicle is not regarded as leading or following vehicle, thus is represented by the accurate boundary for obstacle avoidance. As for the lane-changing vehicles, two lanes are partially occupied. Similar to the roadside stopping vehicles, if the remaining width is less than half lane width, the lane is regarded as blocked, or else, the accurate boundary will be shown.

In (c), a pedestrian is unexpectedly crossing the road, and the ego vehicle should perform an emergency stop. As shown in Fig. 9(c), the end state should not only follow (30-31), but also lie in the temporal corresponding PB. \( A_s \) is then further constrained based on (32):

\[
\forall i, j, \min \left( \text{dist} \left( p_e, \left( p_{ij}, p_{(i+1)j} \right) \right) \right) > d_{\text{thres}}
\]

where \( p_e \) denotes the position of end state.

By involving the environment constraints in advance, sampling number \( n(T_s) \) could be further reduced. In this way, the trajectories with unsafe end state are disqualified, saving time for the collision check with \( A_e \) in the whole time horizon. This trick can further reduce the volume of \( A_s \) without changing the planning result, since (32) is only part of the subsequent environmental constraints put in advance.

With the collision check module, the lower-level planner finally adopted the braking trajectory, and the vehicle performed an emergency stop.

From the three in-lane scenario cases, it can be seen that the PBUD is highly flexible, and perform pertinently in different derived in-lane scenarios.

2) Comparative analysis: For comparative analysis, a multi-lane scenario in Carla Town03 is applied for test. Fig. 10 shows the scenario layout.

The multi-lane scenario is a well-defined scenario, where ego vehicle is required to make smooth car-following or lane-changing decision. Here we will check the performance of the PBUD and three benchmarks, as shown in Fig. 11 and Table I. Note that average samples and average acceleration are counted in the steady car following period (over 80% of final speed). For PBUD and BUD, the gray line shows the predicted boundary at the end of planning horizon, which is applied to limit the sampling process of the trajectory end state.
In the table, average samples reflect the decision complexity, and average acceleration reflects the comfort (the indicator of performance for smooth car following). These two indicators are both supposed to be lower.

**TABLE I**

<table>
<thead>
<tr>
<th>Planning method</th>
<th>Average samples ↓</th>
<th>Average acceleration (m/s²) ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUD</td>
<td>384</td>
<td>0.370</td>
</tr>
<tr>
<td>BUD</td>
<td>198</td>
<td>0.768</td>
</tr>
<tr>
<td>LaneUD</td>
<td>48.0</td>
<td>0.196</td>
</tr>
<tr>
<td>PBUD</td>
<td>46.3</td>
<td>0.213</td>
</tr>
</tbody>
</table>

From Fig. 11 and Table I, we can see that the SUD samples in the full range in the static map limit, resulting in the highest number of samples. Although safety is ensured by collision check with all the obstacles, the car following process is not steady, with a relatively high fluctuation of speed. Similarly, the BUD also treats the scenario as free driving, making samples within the whole drivable area. In decision illustration of BUD, the predicted boundary at the end of planning horizon is shown in the gray color. With the limit of drivable area boundary, the sample number is lower than SUD. However, since semantic and motion status of the boundary is not considered in the plain BUD, the car following is not steady, with an even higher fluctuation than SUD. Different from the above two unconditional decisions, LaneUD has a steady car following performance. The sampling process is directed by the upper level planner (car following strategy), thus the sample number is greatly reduced. This further contributes to the searching of optimal decision, resulting in a steady car following performance. As for the PBUD, with semantic and motion status involved, the potential of compatibility with the car following scenario is obtained compared to the BUD. This results in similar sampling scale and average acceleration with scenario dedicated LaneUD. This proved that the PBUD can make pertinent decision in well-defined scenario.

**B. Junction scenario**

1) **PBUD for junction scenarios:** Fig. 12 shows the performance of PBUD for the typical junction scenario.

   As shown in Fig. 12, junction scenarios are freer than in-lane scenarios, so the drivable space is no longer the combinations of free lane sections. Therefore, in PB, the dynamic objects are represented by their accurate boundaries.

   For upper-level planner, virtual lanes are set in the scenarios as in NDS [18]. Lateral decision can be made similarly to the in-lane scenarios considering the front and back status of the boundary, which can be obtained at intersecting points of virtual lane center line and PB, as shown in Fig. 12.

   As for the sampling process of low-level planner, \( A_k \) is not determined by the road boundary or solid lane boundary like in-lane scenarios. The position of end state should lie in the temporal corresponding PB. The sampling process is illustrated in the following formulae:

\[
\begin{align*}
    d_e & \in \left\{ \begin{array}{c}
    \left[ -d_{\text{max}}, \frac{w_l}{2} + d_{\text{des}} \right], \\
    \left[ \frac{w_l}{2} + d_{\text{des}}, d_{\text{max}} \right], \\
    \left[ -d_{\text{max}}, d_{\text{max}} \right], \\
    \left[ \frac{w_l}{2} + d_{\text{des}}, \frac{w_l}{2} + d_{\text{des}} \right]
    \end{array} \right\}, \\
    i_{\text{des}} & = \begin{cases} i_{\text{min}} & \text{if } \min(\text{dist}(p_e, \{p_{ij}, \text{for } i+1\})) > d_{\text{thres}} \\
    i_{\text{max}} & \text{otherwise}
    \end{cases}
\end{align*}
\]

\[
    \forall i, j, \min(\text{dist}(p_e, \{p_{ij}, \text{for } i+1\})) > d_{\text{thres}}
\]

where \( d_{\text{max}} \) is the maximum sampling range, and \( p_e \) denotes the position of end state.

With (33), possible sampling range is extended to the full drivable space, allowing for possible better solution than limiting sampling range with half lane width in each virtual lane. With (35), the number of end states is reduced than unconditional free sampling, improving the scenario pertinence.

In Fig. 13, the performance of PBUD in two complex junction scenarios is illustrated.

In (a), the ego vehicle has to perform a left-turn in a complex unprotected crowded junction. Different types of elements are shown in different colors, and set with corresponding collision checking safety margin by (12-14). This example shows the ability of PBUD in handling complex scenarios.
In (b), a consecutive lane-junction driving scenario is presented. Two snapshots are given successively. In the first snapshot, the vehicle is driving from the lane section to the junction, and the junction is revealed by the front half of the boundary. The back part is still shown in the in-lane mode of Fig. 8. In the second snapshot, the ego vehicle is leaving the junction. Based on the global planning result, the guidance lines stretch into the lane section, and the lane section about to enter is shown as the front part of the boundary. This consecutive process of scenario changing shows the coherence of PBUD, with continuous guidance lines, and consistent PB that can reveal the characteristics of the scenarios with different boundary parts in a continuous and consistent form.

From the performance of PBUD in junction scenario, it could be seen that PBUD could support pertinent upper-level planning while keeping the framework unified.

2) Comparative analysis: We set an unprotected junction scenario (i.e. no traffic light) in Carla Town05. Fig. 14 shows the scenario layout. The ego vehicle will drive into the junction following a leading vehicle, and another vehicle enters the junction from the right side, requiring ego vehicle to make proper reaction.

The performance of the PBUD and three benchmarks are tested in this scenario, as shown in Fig. 15 and Table II. Since the junction driving is a 2D process rather than car-following, we use trajectory instead of speed profile. To illustrate the driving speed, the positions are marked every 0.5s, like a ticker-tape timer. In this scenario, the right side vehicle will interrupt the car-following process, and is the most affective factor. Therefore, the trajectory of the right side vehicle will be shown in blue color. Note that the blue trajectory only reveals part of the right side vehicle trajectory before ego vehicle passed it in the longitudinal direction.

In the table, minimum distance is the indicator of safety, and is supposed to be higher. Average samples indicates the complexity, while the other two indicate the performance.

<table>
<thead>
<tr>
<th>Planning method</th>
<th>Average samples</th>
<th>Average acceleration (m/s²)</th>
<th>Minimum distance/m</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUD</td>
<td>576</td>
<td>1.04</td>
<td>3.38</td>
</tr>
<tr>
<td>BUD</td>
<td>199</td>
<td>1.24</td>
<td>1.48</td>
</tr>
<tr>
<td>LaneUD</td>
<td>52.5</td>
<td>2.21</td>
<td>1.92</td>
</tr>
<tr>
<td>PBUD</td>
<td>176</td>
<td>0.95</td>
<td>3.52</td>
</tr>
</tbody>
</table>

From Fig. 15, we can see that SUD, BUD and PBUD chose to avoid the right vehicle by driving to the left side, and the right vehicle stopped to wait for ego vehicle to pass, as shown in the dense blue points at the end of its trajectory. With LaneUD, ego vehicle stopped to wait for the right vehicle, forming dense yellow points before the intersecting point. This is because the LaneUD cognized the scenario as multi-lane driving, and regarded the right vehicle as a front vehicle, calling for braking to avoid longitudinal collision. This strategy is safe and reasonable, however, with a wider sampling range, SUD and BUD found the better solution,
thanks to the freer junction scenario. The PBUD chose to use junction driving strategy rather than multi-lane driving strategy, showing that the PBUD can make pertinent decision in this scenario.

From the statistics in Table II, it can be seen that the SUD had the largest sampling scale, while the LaneUD samples the least. Although the complexity was reduced in LaneUD, the better avoiding solution was excluded. In other words, the LaneUD did not make the pertinent decision. As for the acceleration, the braking of LaneUD makes it less smooth than the others. In the last column of Fig. 15, safety is reflected with the minimum distance. Since the BUD does not involve semantic and motion status, the plain drivable area boundary cannot tell static and dynamic obstacle, resulting in the smallest distance and a higher risk. Differently, the SUD and PBUD could set bigger safety threshold for the moving vehicle than static map boundary. Thus, the avoiding decision is made earlier than the BUD, resulting in a safer distance, as shown in the zoomed trajectories in Fig. 15. In summary, the better performance of PBUD than LaneUD and PBUD, along with the lower sampling complexity, proved the pertinence improvement of PBUD in this scenario.

C. Off-road scenario

1) PBUD for off-road scenarios: Fig. 16 shows the performance of PBUD for free off-road scenarios.

![Fig. 16. PBUD for free off-road scenarios](image)

Fig. 16 is set in a non-road square. In off-road driving, e.g., square driving in residential area, a unified decision system should make safe decision and planning while driving along the task guidance. No structural road rule limit needs to be considered in such scenarios, and there is only one guidance line, i.e., the driving task line.

Since the scenario is not well-defined, common decision logic of the PBUD mainstream is adopted without upper-level decision. Thus, sampling range is set to maximum. End state checking with corresponding PB could still be adopted for simplification. \( A_e \) is then set as:

\[
d_c \in [-d_{\text{max}}, d_{\text{max}}] \\
v_c \in [0, v_{\text{max}}] \\
\forall i,j, \min(\text{dist}(p_e, (p_{ij}, p_{(i+1)j}))) > d_{\text{thres}}
\]

In the unconditional solution of PBUD, no scenario characteristics need to be considered. In this case, the PBUD is nearly equivalent to unified decision based on unconditional sampling.

2) Comparative analysis: The above off-road scenario is applied for comparative analysis. Similar to the environment of robotic application, the ego vehicle has to drive in an irregular space, avoiding vehicles, pedestrians and unknown obstacles (tables and flowerpots in this example).

The performance of the PBUD and three benchmarks are tested in this scenario, as shown in Fig. 17 and Table III. Here we showed the trajectories with the ticker-tape timer with gap of 1s. In this scenario, the two vehicles are parked, shown as two static rectangles. Three pedestrians will interrupt the ego vehicle, walking with a low speed of 0.5m/s. Note that the blue trajectories only reveals the part of the pedestrian trajectories before ego vehicle drove past them.

In this scenario, we further considered time consumption as statistical indicator due to the huge difference of the decision methods in efficiency.

<table>
<thead>
<tr>
<th>Planning method</th>
<th>Average samples ↓</th>
<th>Average acceleration (m/s²) ↓</th>
<th>Minimum distance/m ↑</th>
<th>Time consumption/s ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUD</td>
<td>595</td>
<td>0.451</td>
<td>1.40</td>
<td>21.86</td>
</tr>
<tr>
<td>BUD</td>
<td>358</td>
<td>0.317</td>
<td>0.50</td>
<td>20.98</td>
</tr>
<tr>
<td>LaneUD</td>
<td>21.5</td>
<td>0.281</td>
<td>1.25</td>
<td>47.62</td>
</tr>
<tr>
<td>PBUD</td>
<td>126</td>
<td>0.363</td>
<td>1.75</td>
<td>22.12</td>
</tr>
</tbody>
</table>

From Fig. 17 and Table III, we can see that only LaneUD chose to stop and wait for the third pedestrian to cross. SUD, BUD and PBUD chose to avoid, like in the junction scenario. This is because of LaneUD recognize the pedestrian as the leading obstacle, misuse of car-following strategy and the strict limit of sampling missed the optimal solution in this scenario. SUD and BUD are suit for free scenario, where obstacle avoidance in SUD and the drivable area limit in BUD helped them in looking for optimal solution in a wide sample range. The drivable area limit in trajectory sampling reduced the sampling scale in BUD. However, since the BUD does not consider semantic and motion status in safety threshold setting, the minimum distance to collision is smaller than SUD. As for the proposed PBUD, the objective space boundary ensures the basic driving performance in off-road or unrecognized scenario. Also, the semantic and motion status information helped it to achieve comparable performance to SUD (better than BUD). In summary, the better performance of PBUD than
LaneUD and BUD, along with the lower sampling complexity, proved the pertinence improvement of PBUD in this scenario.

VI. CONCLUSION

This paper proposed a novel unified driving system that allows scenario pertinent performance. The pertinent boundary was established to represent the environment details in a consistent form for the sake of a unified planner. Experiment results indicate that the proposed PBUD has pertinent performance according to different driving scenarios, reducing the complexity of decision for optimized performance, whereas a general solution is assured in undefined or unidentified scenarios. The environment model consistency and the planner unity are maintained in scenario-oriented adaption, providing coherent performance across different scenarios. The PBUD presented in this paper is suitable for both undefined scenarios and well-defined scenarios. Although the highly complex driving scenarios are not completely discussed due to space limit, our proposed system could be easily extended to other scenarios due to its flexibility.

The PB developed in this paper is based on geometric closed curve in polar coordinate, which can be used to reorganize driving environment elements in a mathematically consistent form, fundamentally supporting the scenario pertinent decision system. In this paper, the boundary sections are independently considered without an overall understanding of the environment. Hence, our future work will be focused on the overall driving environment understanding, which can be derived from the curvilinear integral properties of the closed PB. Furthermore, the mechanism of scenario influence on overall environment understanding will be further studied. The scenario-pertinent flexibility of a general and comprehensive environment model could provide higher performance potential for a united driving system, thereby benefiting for the development of multi-scenario driving system of autonomous vehicles.

ACKNOWLEDGMENT

The authors would like to thank the support by National Natural Science Foundation of China (U1634203 and 52102396).

REFERENCES


