

Selecting Spots by Explicitly Predicting Intention from Motion History Improves Performance in Autonomous Parking

Long Kiu Chung^{1,2}, David Isele¹, Faizan M. Tariq¹, Sangjae Bae¹, Shreyas Kousik², and Jovin D’sa¹

Abstract—In many applications of social navigation, existing works have shown that predicting and reasoning about human intentions can help robotic agents make safer and more socially acceptable decisions. In this work, we study this problem for autonomous valet parking (AVP), where an autonomous vehicle ego agent must drop off its passengers, explore the parking lot, find a parking spot, negotiate for the spot with other vehicles, and park in the spot without human supervision. Specifically, we propose an AVP pipeline that selects parking spots by explicitly predicting where other agents are going to park from their motion history using learned models and probabilistic belief maps. To test this pipeline, we build a simulation environment with reactive agents and realistic modeling assumptions on the ego agent, such as occlusion-aware observations, and imperfect trajectory prediction. Simulation experiments show that our proposed method outperforms existing works that infer intentions from future predicted motion or embed them implicitly in end-to-end models, yielding better results in prediction accuracy, social acceptance, and task completion. Our key insight is that, in parking, where driving regulations are more lax, explicit intention prediction is crucial for reasoning about diverse and ambiguous long-term goals, which cannot be reliably inferred from short-term motion prediction alone, but can be effectively learned from motion history. Website: <https://sites.google.com/view/chung2026selecting>.

I. INTRODUCTION

In social navigation, it is important for an autonomous agent to predict and reason with other human agents’ intentions (e.g. destination, choice of actions) in order to synthesize plans that are safe and socially acceptable. Depending on the application (walking in crowds, driving through traffic, etc.), many existing works have proposed how and where intention prediction should fit within a planning framework. For the specific problem of *autonomous valet parking* (AVP), where a car must drop off its passengers, explore the parking lot, find a parking spot, negotiate for the spot with other vehicles, and park into the spot without human supervision, we define intentions as the intended parking spots of the other vehicles, the role of which we wish to investigate. Specifically, we present an AVP pipeline that decides where an autonomous ego agent should park by explicitly predicting the intentions of other agents from motion history, which outperforms existing methods that infer intentions from trajectory predictions or handle them implicitly in an end-to-end fashion. An overview of our method is shown in Figure 1.

¹Honda Research Institute (HRI), Mountain View, CA. ² Georgia Institute of Technology, Atlanta, GA. Corresponding Authors: lichung33@gatech.edu, jovin_dsa@honda-ri.com. All work was done when Long Kiu Chung was employed by HRI.

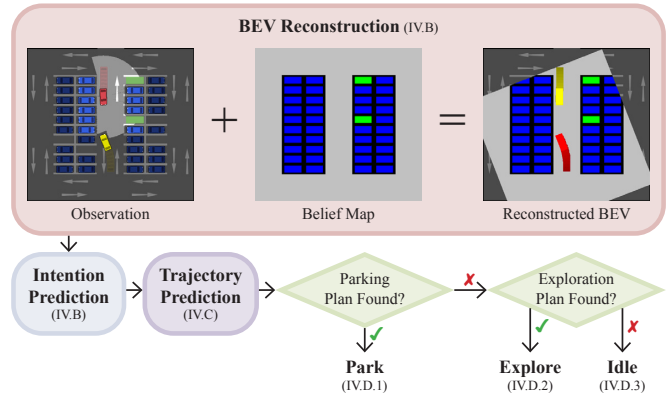


Fig. 1: A flow chart of our method, with each component marked by its corresponding section number. At each timestep, we combine information from the current observation and a probabilistic belief map of where cars are parked to reconstruct the bird’s-eye view (BEV) map of other agents. This enables the use of a learned intention model, which we use to make safe and socially acceptable parking decisions.

A. Related Work

We now review how intentions are leveraged in other social navigation scenarios, and argue why they cannot be straightforwardly applied to the AVP problem. We then review existing work on AVP and state our contribution.

1) *Intents in Social Navigation*: Many existing works have proposed how and where intention predictions should fit within a planning framework in other applications of human-robot interactions. For traffic scenarios such as lane merging, the driver’s intention (whether it yields to ego or not) has been explicitly predicted from motion history using Probabilistic Graphical Model [1] or from trajectory predictions using heuristic-based [2] and learning-based [3] intention models. For crowd navigation, [4] uses a generative adversarial network (GAN) to predict the intention (goal location) from past trajectories, whereas [5] inferred a pedestrian’s intention (to avoid, maintain course, or approach the ego) from gaze directions and trajectory predictions. There have also been many methods that implicitly handle other agents’ intentions by mapping directly from observations to ego actions [6]–[8].

2) *Autonomous Parking*: Unfortunately, results from Section I-A.1 may not directly apply to parking, because intentions can be defined very differently in a parking lot: there are many spots where an agent may park, and vehicles do not necessarily follow lane directions and boundaries. However, many existing works either address only parts of the AVP problem (e.g. exploration [9], [10], deadlock

resolution [11] and collision avoidance [12]–[14], generation of parking trajectories [15]–[18]), or do not have humans in the loop [19]. There is a significant lack of literature on spot selections and constructing a complete AVP pipeline; to the best of our knowledge, only [20] addresses this via an AVP pipeline where the ego selects spots with occupancy probability updates by inferring intentions from trajectory predictions.

That said, the results in [20] were obtained under certain simplifying assumptions: their sensing model ignores occlusion, simulated agents do not react to the ego’s actions, and most crucially, future trajectories of all agents are assumed to be known. Of course, there is no such thing as perfect trajectory prediction; but unlike other traffic scenarios, where good trajectory models can be trained from large amount of high quality motion data [21], [22], there is only one publicly available vehicle motion dataset for parking (to the best of our knowledge), the Dragon Lake Parking (DLP) Dataset [23], which contains only 3.5 hours of footage from a single parking lot.

While [23], [24] proposed trajectory prediction networks trained on this dataset, the trajectory predictions are conditioned on an intention prediction network. As such, applying them to [20] would be *paradoxical*: one would be predicting an intention from a trajectory predicted from an intention. Moreover, these methods assume semantic bird’s-eye view (BEV) information around the vehicle of interest, but BEV information is only realistically obtainable for the ego; predictions are usually needed for other agents, for which we only have partial BEV information. In this paper, we propose an AVP pipeline that enables using [23] on other agents, which outperforms existing baselines when examined under more realistic modeling assumptions.

B. Contributions

Our core contribution is a new AVP pipeline that selects parking spots by explicitly predicting other agents’ intention from motion history, which we find outperforms a variety of baselines. We tested the method by constructing a simulation environment with realistic modeling choices, which include occlusion-aware sensing models, reactive agents, and is modular to the trajectory prediction method. Our novel pipeline is enabled by three key supporting contributions:

- 1) We introduce a novel method to synthesize BEV information for surrounding agents using only information available to the ego, allowing intention models to predict where others are planning to park. This enables the use of existing learned intention models for planning and decision-making in AVP.
- 2) We propose a parking spot selection strategy for the ego agent to avoid contended spots using predicted intentions. We find that that this method performs better in prediction accuracy, social acceptance, and task completion compared to spot selection by explicitly inferring intents from trajectory predictions [20] and by implicit inferring intention in an end-to-end framework (using [23] on the ego only).

- 3) We propose an intention-conditioned trajectory prediction method for AVP using cubic Bézier curves. Compared to other trajectory generation methods such as constant velocity, Hybrid A*, and the learned trajectory prediction model from [23], our method best balances between low computation time, high prediction accuracy, and task completion rate.

II. PROBLEM FORMULATION

In this paper, we aim to control the ego agent such that it successfully parks into a parking spot without colliding with other vehicles and pedestrians as soon as possible, while remaining within the parking lot’s boundaries. We now detail our assumptions on the parking lot and on the dynamics of vehicles and pedestrians, and formalize our problem statement at the end of this section.

A. Parking Lot

We denote the parking lot as $L \subseteq \mathbb{R}^2$, which we assume is compact and connected. We assume the parking lot has one entrance at coordinates $\mathbf{p}_{\text{ent}} = [x_{\text{ent}} \ y_{\text{ent}}]^\top \in L$, containing $n_{\text{spot}} \in \mathbb{N}$ rectangular parking spots $S_i \subseteq L$, $i = 1, \dots, n_{\text{spot}}$, where each S_i is a rectangle parameterized by its length $\ell_{S_i} \in \mathbb{R}_{>0}$, width $w_{S_i} \in \mathbb{R}_{>0}$, 2-D coordinates of its center $\mathbf{p}_{S_i} = [x_{S_i} \ y_{S_i}]^\top \in L$, and heading in the direction that faces opposite of the road $\theta_{S_i} \in \mathbb{R}$.

Groups of parking spots are partitioned by $n_{\text{road}} \in \mathbb{N}$ straight roads $R_j \subseteq L$, $j = 1, \dots, n_{\text{road}}$, on which vehicles can travel. Each R_j is a rectangle parameterized by a start point $\mathbf{p}_{\text{start},j} = [x_{\text{start},j} \ y_{\text{start},j}]^\top \in \mathbb{R}^2$, an end point $\mathbf{p}_{\text{end},j} = [x_{\text{end},j} \ y_{\text{end},j}]^\top \in \mathbb{R}^2$ (which defines a center line C_j), and width $w_{R_j} \in \mathbb{R}_{>0}$. Mathematically, the rectangle is the Minkowski sum of C_j and a square with side length w_{R_j} , centered at the origin and rotated by $\theta_{R_j} = \text{atan2}(y_{\text{end},j} - y_{\text{start},j}, x_{\text{end},j} - x_{\text{start},j})$.

We assume the ego agent has knowledge about the structure of the parking lot, specifically \mathbf{p}_{ent} , $(\mathbf{p}_{S_i})_{i=1}^{n_{\text{spot}}}$, $(\theta_{S_i})_{i=1}^{n_{\text{spot}}}$, $(R_j)_{j=1}^{n_{\text{road}}}$, and L . This assumption is not too restrictive and is also carried by many existing methods [19], [20], [24]. We leave generalization to unmapped parking lots as future work.

B. Vehicle Model

We model the vehicle agents as a Reeds-Shepp car in discrete time [25], with dynamics

$$\mathbf{s}_{V_k,t+\Delta t} = \mathbf{s}_{V_k,t} + \begin{bmatrix} v_{V_k,t} \cos(\theta_{V_k,t}) \\ v_{V_k,t} \sin(\theta_{V_k,t}) \\ \omega_{V_k,t} \end{bmatrix} \Delta t, \quad (1)$$

where $k \in \{0, 1, \dots, n_{\text{other}}\}$, $\mathbf{s}_{V_k,t} = [\mathbf{p}_{V_k,t} \ \theta_{V_k,t}]^\top \in \mathbb{R}^2 \times [-\pi, \pi]$ is the state ($\mathbf{p}_{V_k,t} = [x_{V_k,t} \ y_{V_k,t}]^\top \in \mathbb{R}^2$ is the 2-D coordinates and $\theta_{V_k,t}$ is the heading), $\mathbf{u}_{V_k,t} = [v_{V_k,t} \ \omega_{V_k,t}]^\top \in \mathbb{R}^2$ is the control inputs of the vehicle $V_k \subseteq \mathbb{R}^2$ at time $t \in \{\tau \mid \text{mod}(\tau, \Delta t) = 0, \tau \in \mathbb{R}\}$, and $\Delta t \in \mathbb{R}_{>0}$ is the timestep size.

We denote by V_0 the ego vehicle and by $V_1, \dots, V_{n_{\text{other}}}$ the other $n_{\text{other}} \in \{0\} \cup \mathbb{N}$ vehicle agents. In particular, each V_k is a rectangle $\{[x \ y]^\top \mid -0.5\ell_{V_k} \leq x \leq 0.5\ell_{V_k}, -0.5w_{V_k} \leq$

$y \leq 0.5w_{V_k}$ with length $\ell_{V_k} \leq \min\{\ell_{S_i} \mid i = 1, \dots, n_{\text{spot}}\}$ and width $w_{V_k} \leq \min\{w_{S_i} \mid i = 1, \dots, n_{\text{spot}}\}$. At time t , the space each vehicle occupies $V_{k,t}$ is V_k rotated by the heading $\theta_{V_k,t}$ and translated by the coordinates $\mathbf{p}_{V_k,t}$.

C. Pedestrian Model

We model the pedestrians with holonomic dynamics:

$$\mathbf{p}_{P_m,t+\Delta t} = \mathbf{p}_{P_m,t} + \begin{bmatrix} v_{P_m,t} \cos(\theta_{P_m,t}) \\ v_{P_m,t} \sin(\theta_{P_m,t}) \end{bmatrix} \Delta t, \quad (2)$$

where $m \in \{1, \dots, n_{\text{ped}}\}$, $n_{\text{ped}} \in \{0\} \cup \mathbb{N}$ is the number of pedestrians, $\mathbf{p}_{P_m,t} = [x_{P_m,t} \ y_{P_m,t}]^\top \in \mathbb{R}^2$ is the 2-D position and $\mathbf{u}_{P_m,t} = [v_{P_m,t} \ \theta_{P_m,t}]^\top \in \mathbb{R}^2$ is the control input of the pedestrian $P_m \subseteq \mathbb{R}^2$ at time t .

We assume the volume of each pedestrian P_m is a disc with radius $r_{P_m} \in \mathbb{R}_{\geq 0}$. At time t , the space each pedestrian occupies $P_{m,t}$ is P_m translated to $\mathbf{p}_{P_m,t}$.

D. Problem Statement

Mathematically, we wish to tackle the following problem:

Problem 1 (AVP Problem). Find $(\mathbf{u}_{V_0,t})_{t=0}^{t_{\text{park}}-\Delta t}$ and minimize the parking time $t_{\text{park}} \in \{\tau \mid \tau \in \mathbb{R}_{\geq 0}, \text{ mod}(\tau, \Delta t) = 0\}$ such that $\exists i \in \{1, \dots, n_{\text{spot}}\}$, where

$$V_{0,t_{\text{park}}} \subseteq S_i, \quad (3)$$

$$V_{0,t} \cap V_{k,t} = \emptyset, \quad (4)$$

$$V_{0,t} \cap P_{m,t} = \emptyset, \quad (5)$$

$$V_{0,t} \subseteq L, \quad (6)$$

for all $t = 0, \dots, t_{\text{park}}$, $k = 1, \dots, n_{\text{other}}$, $m = 1, \dots, n_{\text{ped}}$.

In the next section, we detail the information available to the ego, and how each other vehicle and pedestrian agent reacts to the ego's actions at each timestep, which can make Problem 1 non-trivial and more realistic.

III. SIMULATION ENVIRONMENT

We now detail our modeling choices on sensors and agents, with the goal of simulating occlusion-aware observations and reactive drivers and pedestrians in our environment.

A. Observation Model

To simulate an occlusion-aware sensing model, we employ a ray tracing algorithm. To start, we assume we know a priori a compact, connected *base sensing region* $Q_0 \subseteq \mathbb{R}^2$ for the ego agent. For example, Q_0 could be a rectangle or an ellipse. At time $t = 0, \dots, t_f - \Delta t$, where $t_f \in \{\tau \mid \tau \geq t_{\text{park}}, \text{ mod}(\tau, \Delta t) = 0\}$ is the final time, we define the *sensing region* $Q_{0,t} \subseteq \mathbb{R}^2$ as Q_0 rotated by the ego's current heading $\theta_{V_0,t}$ and translated by the coordinates $\mathbf{p}_{V_0,t}$.

At each t , the ego shoots out $n_{\text{ray}} \in \mathbb{N}$ rays $B_{j,t} \subseteq Q_{0,t}$, $j = 1, \dots, n_{\text{ray}}$, from the ego until it either hits the boundary of the parking lot, the sensing region, or another vehicle agent. Mathematically, each $B_{j,t}$ is the shortest line in a direction that begins at $\mathbf{p}_{V_0,t}$ and ends at a point in either ∂L , $(\partial V_{k,t})_{k=1}^{n_{\text{other}}}$, or $\partial Q_{0,t}$, where $\partial \cdot$ refers to a set's boundary. The ego is then able to observe all vehicles, spots, and pedestrians that intersect with any of the rays.

Furthermore, we assume the ego is able to distinguish between vacant spots ($S_i \not\supseteq V_{k,t} \forall k \in \{1, \dots, n_{\text{other}}\}$), occupied spots ($S_i \supseteq V_{k,t}$ for some $k \in \{1, \dots, n_{\text{other}}\}$), vehicles that are parked ($V_{k,t} \subseteq S_i$ for some $i \in \{1, \dots, n_{\text{spot}}\}$, henceforth referred to as *static vehicles*), and vehicles that are not parked ($V_{k,t} \not\subseteq S_i \forall i \in \{1, \dots, n_{\text{spot}}\}$, henceforth referred to as *dynamic vehicles*). We denote the indices of observed vacant spots, occupied spots, static vehicles, dynamic vehicles, and pedestrians as $I_{\text{vac},t}, I_{\text{occ},t}, K_{\text{dv},t}, K_{\text{sv},t}, M_{\text{dp},t} \subseteq \mathbb{N}$ and their cardinality as $n_{\text{vac},t}, n_{\text{occ},t}, n_{\text{dv},t}, n_{\text{sv},t}, n_{\text{dp},t} \in \{0\} \cup \mathbb{N}$.

For each observed vehicle agent, the ego has access to their length ℓ_{V_k} , width w_{V_k} , and current and some historical states for the past t_{hist} time unit $(s_{V_k,\tau})_{\tau=t-t_{\text{hist}}}^t \forall k \in K_{\text{dv},t} \cup K_{\text{sv},t}$. For each observed pedestrian, the ego has access to their radius r_{P_m} , and positions of their current and previous timestep $\mathbf{p}_{P_m,t-\Delta t}, \mathbf{p}_{P_m,t} \forall m \in M_{\text{dp},t}$. The access to historical states is needed to enable the use of the learned intention model in [23], as well as some of the trajectory prediction methods.

We assume the ego correctly observes every spots and agents it senses, even though real life sensors can have uncertainty about its observations, usually as a function of distance from the ego [20]. We opt for this simpler model because it is unclear how uncertain, probabilistic observations can be straightforwardly used in the intention prediction model from [23], which we leave as future work.

B. Reactive Agents

We aim to simulate other vehicles and pedestrians such that they would react to the ego's actions and not actively seek out collisions. We accomplished this by introducing a parameter called *passiveness* $n_{\text{pass},A} \in \{0\} \cup \mathbb{N}$, where $A \in \{V_1, \dots, V_{n_{\text{other}}}, P_1, \dots, P_{n_{\text{ped}}}\}$ is another vehicle or pedestrian agent. Intuitively, passiveness encodes how far into the future an agent begins braking. Formally, it is the number of timesteps an agent looks ahead (based on its plan and other agents' current position) to check for collisions. Higher passiveness makes an agent more cautious by considering collisions further ahead. To implement this caution, our reactive agent brakes if it anticipates a collision, and resumes its plan otherwise; thus, a less passive agent is less likely to alter its behavior with stopping actions.

Mathematically, we assume each pedestrian and vehicle agent (except the ego) A have some predefined plans $\tilde{\mathbf{u}}_{A,t} \in \mathbb{R}^2$ for $t = -t_{\text{hist}}, \dots, t_f + (n_{\text{pass},A} - 1)\Delta t$, where $t_{\text{hist}} \in \{\tau \mid \tau \in \mathbb{R}_{\geq 0}, \text{ mod}(\tau, \Delta t) = 0\}$ is the time of some previous history. At each timestep before $t = 0$, we control them by following the predefined plan (i.e. $\mathbf{u}_{A,t} = \tilde{\mathbf{u}}_{A,t}$ for $t = -t_{\text{hist}}, \dots, -\Delta t$). Afterwards, for each $t = 0, \dots, t_f - \Delta t$, the agent would take the control $\mathbf{u}_{A,t} = \mathbf{0}$ if $A_\tau \cap V_{0,t} \neq \emptyset$ for some $\tau = t + \Delta t, \dots, t + n_{\text{pass},A}\Delta t$, or take the next unused plan from $(\tilde{\mathbf{u}}_{A,\tau})_{\tau=0}^{t_f-\Delta t}$ otherwise.

While simple, this algorithm allows us to gauge the social acceptance of the ego agent by simulating spot competition, interruptions, and at-fault collisions, while also preventing the other agents from being self-destructive (as long as $n_{\text{pass},A} > 0$). One could argue, however, that such heuris-

tics may not necessarily reflect those of a real driver. In contrast, prior works in social navigation have used data-driven trajectory models to better capture human behaviors [26]. That said, we leave the development of more realistic reactive agents to future work, since existing trajectory models do not account for pedestrians and do not guarantee kinodynamically feasible trajectories [23].

IV. PROPOSED METHOD: AVP PIPELINE

We now propose an AVP pipeline with explicit intention prediction from motion history. Our overall plan is to enable the use of a learned intention prediction model such as [23] on the observed dynamic vehicles by reconstructing their BEV information using belief maps. We then predict their future trajectories by conditioning cubic Bézier curves on the predicted intentions. Finally, using Hybrid A* on the ego agent, we compute paths to park in spots less likely to be parked by other dynamic vehicles in the near future. If no such path exists, the ego agent will instead choose to explore the parking lot for vacant spots. An overview of our pipeline is shown in Figure 1.

A. Intention Model

Our goal is to predict the intentions of other observed dynamic vehicles (specifically, their probabilities of parking into each observed vacant spot, or to explore the parking lot by traveling to an *exploration point*) using a learned intention model. We opt for the model in [23], as it is the only learned parking intention model with publicly available code, to the best of our knowledge. Succinctly, at each time t for each vehicle agent V_k , the intention model in [23] is a convolutional neural network (CNN) that outputs $\eta_{k,i} \in [0, 1]$, the intention probability of a candidate parking spot S_i , for each i in the index set $I_{\text{can},k} \subseteq \mathbb{N}$, and $\hat{\eta}_{k,q} \in [0, 1]$, the intention probability of a candidate exploration point $\mathbf{s}_{\text{exp},k,q} \in \bigcup_{j=1}^{n_{\text{road}}} (C_j \times \{\theta_{R_j}, \theta_{R_j} + \pi\})$, for $q = 1$ to the number of candidate points $n_{\text{exp},k} \in \{0\} \cup \mathbb{N}$. It takes as inputs the candidate exploration points $\mathbf{s}_{\text{exp},k,q}$, the distance between the vehicle and the candidate spot $d_{k,i} \in \mathbb{R}_{\geq 0}$, the cosine similarity between the vehicle’s heading vector and the vector from the vehicle to the candidate spot $a_{k,i} \in [-1, 1]$, the predicted speed of the vehicle $\bar{v}_{V_k,t} \in \mathbb{R}_{\geq 0}$, the distance between the vehicle and the parking lot entrance $d_{\text{ent}} \in \mathbb{R}_{\geq 0}$, the time the vehicle has spent in the parking lot $t_{\text{lot}} \in \{\tau \mid \tau \in \mathbb{R}_{\geq 0}, \text{mod}(\tau, \Delta t) = 0\}$, and the semantic BEV image around the vehicle with the i^{th} spot marked $\mathbf{V}_{k,i,t} \in \mathbb{R}^{400 \times 400 \times 3}$ (whereas $\mathbf{V}_{k,0,t}$ do not have any spots marked differently).

Unfortunately, it is non-trivial to deploy this model in a realistic setting. Specifically, from our modeling choices in Section III, all required inputs are straightforward to obtain *except* for the semantic BEV images ($\mathbf{V}_{k,i,t}$) $_{i \in \{0\} \cup I_{\text{can},k}}$ for each observed vehicle V_k . This is because the construction of these images requires knowledge of the occupied volume of all roads, parking spots (assumed known from Section II-A), static vehicles, and current and past states (up to t_{hist} time units) of dynamic vehicles within a 40 m square centered

around the vehicle (of which we only observe a subset). Though training a model without semantic BEV images is possible, ablation studies from [23] show significant performance loss. [23] also only showed that their intention model improved performance in trajectory prediction, but did not suggest how it could be used for decision-making in AVP. As such, we now propose a method to estimate and reconstruct the required BEV images with belief maps using only observed information, and suggest heuristics for spot selection based on this technique.

B. BEV Reconstruction with Belief Map

Our strategy to reconstruct semantic BEV images for each observed dynamic vehicles (V_k) $_{k \in \mathcal{K}_{\text{dv},t}}$ is to update a belief map of where cars are parked, and using that to infer the BEV of other agents. We define the *belief* $b_{t,i} \in [0, 1]$ of a parking spot S_i at time t as the estimated probability that other vehicles has parked or will park into the spot in the near future. Since we know the number of spots n_{spot} in the parking lot a priori, we store the ego’s current belief of all parking spots into a *belief map* $\mathbf{b}_t = [b_{t,1} \ \dots \ b_{t,n_{\text{spot}}}]^{\top}$ for each timestep. We initialize $\mathbf{b}_{-\Delta t} = (0.5)\mathbf{1}_{n_{\text{spot}} \times 1}$ to show that we have no initial information about the occupancy probability of any of the parking spots.

At each timestep $t = 0, \Delta t, \dots$, we first update our belief with the current observation:

$$\tilde{b}_{t,i} = \begin{cases} 0 & \text{if } i \in I_{\text{vac},t}, \\ 1 & \text{if } i \in I_{\text{occ},t}, \\ b_{t-\Delta t,i} & \text{otherwise,} \end{cases} \quad (7)$$

for $i = 1, \dots, n_{\text{spot}}$, where $\tilde{b}_{t,i} \in [0, 1]$ is the initial belief. In other words, we initially believe that each observed vacant and occupied spot will stay vacant or occupied for the near future. For spots that we cannot observe, we keep our beliefs of them from the previous timestep.

We then estimate the BEV of each observed dynamic vehicles as follows: first, we draw the occupied volumes of the road, the spots (given from Section II-A), and the current and past states of each observed vehicles (including the ego). Then, for each unobserved spot (S_i) $_{i \notin I_{\text{vac},t} \cup I_{\text{occ},t}}$ within the BEV, we draw a vehicle of some default length ℓ_{def} and width w_{def} at the spot (centered at \mathbf{p}_{S_i} and rotated by θ_{S_i}) if $b_{t,i} \geq \beta$ (i.e. we believe it is *occupied*) for some threshold $\beta \in [0, 1]$.

Finally, before the next timestep, we use the intention model in Section IV-A to update our belief. For each observed dynamic vehicles, we define their candidate spots (S_i) $_{i \in I_{\text{can},k}}$ as all spots with $b_{t,i} < \beta$ (i.e. we believe it is *vacant*) within their BEV. We then define their candidate exploration points and compute the other inputs to the intention model the same way as in [23]. This enables us to compute the intention probabilities $\eta_{k,i}$ (and $\hat{\eta}_{k,q}$) of each observed dynamic vehicle parking into each observed vacant

spot, which we can use to update the belief as:

$$b_{t,i} = \begin{cases} 1 - \prod_{\eta \in \{\eta_{\bar{k},\bar{i}} \mid \bar{i}=i \in I_{\text{can},\bar{k}}, \bar{k} \in K_{\text{dv},t}\}} (1 - \eta) & \text{if } i \in I_{\text{vac},t}, \\ \tilde{b}_{t,i} & \text{otherwise.} \end{cases} \quad (8)$$

In other words, we update the belief of each observed vacant spot as the probability that *any* of the observed dynamic vehicle will park into the spot in the near future.

C. Trajectory Prediction

To maximize the utility of our method in Section IV-B, we now condition trajectory predictions of other agents on their predicted intentions for collision avoidance. At each timestep t , for each observed dynamic vehicle $(V_k)_{k \in K_{\text{dv},t}}$, we wish to predict trajectories for all intentions with probability higher than some threshold $\mu \in [0, 1]$. That is, for each $\eta_{k,i} \geq \mu$ and $\hat{\eta}_{k,q} \geq \mu$, we wish to predict $(\mathbf{s}_{V_k,\tau})_{\tau=t+\Delta t}^{t+t_{\text{pred}}}$ with predictions $(\bar{\mathbf{s}}_{V_k,\tau})_{\tau=t+\Delta t}^{t+t_{\text{pred}}}$, where $\bar{\mathbf{s}}_{V_k,\tau} \in \mathbb{R}^2 \times [-\pi, \pi]$ is the prediction of the ground truth $\mathbf{s}_{V_k,\tau}$ and $t_{\text{pred}} \in \{\tau \mid \tau \in \mathbb{R}_{\geq 0}, \text{mod}(\tau, \Delta t) = 0\}$ is the prediction time.

To accomplish this, we employ a goal-conditioned trajectory prediction scheme using cubic Bézier curves. A cubic Bézier curve is a 2-D spline defined by two control points $\mathbf{p}_{\text{bez},0}, \mathbf{p}_{\text{bez},3} \in \mathbb{R}^2$ that define the start and end points, and two control points $\mathbf{p}_{\text{bez},1}, \mathbf{p}_{\text{bez},2} \in \mathbb{R}^2$ that determines the slope at the beginning and the end of the curves [27]. To connect the agent to the intention with the appropriate end points and slopes, we define the control points as:

$$\mathbf{p}_{\text{bez},0} = \mathbf{p}_{V_k,t}, \quad (9)$$

$$\mathbf{p}_{\text{bez},3} = \mathbf{p}_{\text{goal}}, \quad (10)$$

$$\mathbf{p}_{\text{bez},1} = \mathbf{p}_{V_k,t} + \zeta \bar{v}_{V_k,t} \begin{bmatrix} \cos(\theta_{V_k,t} + \pi) \\ \sin(\theta_{V_k,t} + \pi) \end{bmatrix}, \quad (11)$$

$$\mathbf{p}_{\text{bez},2} = \mathbf{p}_{S_i} + \zeta \bar{v}_{V_k,t} \begin{bmatrix} \cos(\theta_{\text{goal}} + \pi) \\ \sin(\theta_{\text{goal}} + \pi) \end{bmatrix}, \quad (12)$$

where the goal is the center and heading of the spot if the intention is to park ($\mathbf{p}_{\text{goal}} = \mathbf{p}_{S_i}$, $\theta_{\text{goal}} = \theta_{S_i}$) or as the exploration point if the intention is to explore ($[\mathbf{p}_{\text{goal}} \ \theta_{\text{goal}}] = \mathbf{s}_{\text{exp},k,q}$), and $\zeta \in \mathbb{R}_{\geq 0}$ adjusts the distance between the end control points and the control points in-between.

We can now predict the dynamic vehicles' trajectories. For $\tau = t + \Delta t, \dots, t + t_{\text{pred}}$, we compute $\bar{\mathbf{s}}_{V_k,\tau}$ by assuming that the vehicle travels at a constant velocity $\bar{v}_{V_k,t}$ along the curve. In the case where $t_{\text{plan}} - t_{\text{pred}}$ more timesteps of predictions are needed for some $t_{\text{plan}} \in \{\tau \mid \tau > t_{\text{pred}}, \text{mod}(\tau, \Delta t) = 0\}$, we fill the rest of the predicted timesteps using a constant velocity model.

Note that we do not predict the intentions of the pedestrians due to the lack of intention models for pedestrians in a parking lot. Instead, for each observed pedestrians $(P_m)_{m \in M_{\text{dp},t}}$, we only use a constant velocity model to predict their trajectories $(\bar{\mathbf{s}}_{P_m,\tau})_{\tau=t+\Delta t}^{t+\max\{t_{\text{pred}}, t_{\text{plan}}\}}$, where $\bar{\mathbf{s}}_{P_m,\tau} \in \mathbb{R}^3$ is the prediction of the ground truth $\mathbf{s}_{P_m,\tau}$. We predict the velocity of the pedestrian as $\bar{v}_{P_m,t} = \frac{\|\mathbf{p}_{P_m,t} - \mathbf{p}_{P_m,t-\Delta t}\|_2}{\Delta t}$.

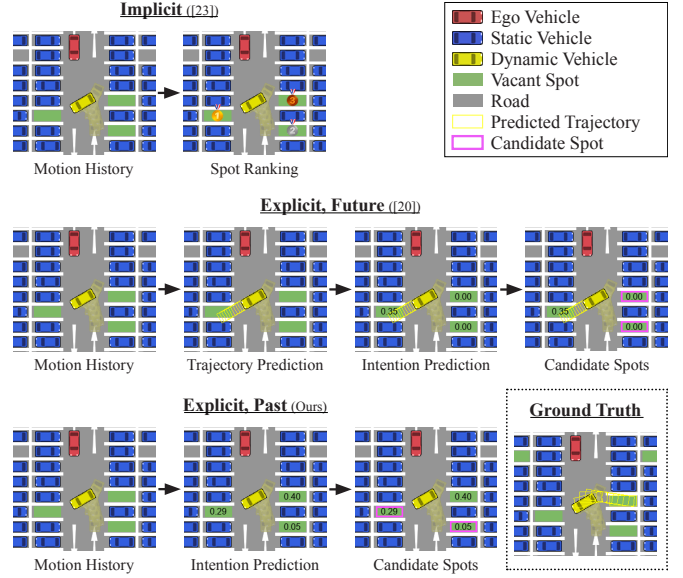


Fig. 2: An overview of the methods under comparison, with legend shown on top right. In this scenario, the dynamic vehicle (other agent) is backing into a spot after moving forward to adjust its angle (a “right-south-up” maneuver [19]). Our proposed method is the only one to explicitly and correctly predicted this behavior.

D. Final Trajectory Planning

After the intention and trajectory prediction in Sections IV-A and IV-B, the ego carries out its decisions similar to the pipeline in [20] at each timestep t , which we succinctly summarize here:

1) *Parking*: Our ego first considers parking if a safe path to a non-contended spot is found. We define the ego's candidate parking spots $(S_i)_{i \in I_{\text{can},0}}$ as all observed vacant spots with belief less than some threshold $\delta \in [0, 1]$, i.e. $I_{\text{can},0} = \{i \mid b_{t,i} \leq \delta, i \in I_{\text{vac},t}\}$.

For each of the candidate spot, we use the Hybrid A* implementation from [20] to compute $(\mathbf{u}_{V_0,\tau})_{\tau=t}^{t+t_{\text{plan}}-\Delta t}$ such that our ego agent is parked by the end of the plan while avoiding spots we believed to be occupied. That is, $V_{0,t+t_{\text{plan}}} \subseteq S_i$ and $V_{0,\tau} \subseteq L$, $V_{0,\tau} \cap \{S_i \mid b_{\tau,i} \geq \beta, i = \{1, \dots, n_{\text{spot}}\}\} = \emptyset$ for $\tau = t + \Delta t, \dots, t + t_{\text{plan}}$.

If this path also avoids collisions with the predicted trajectories of vehicles and pedestrians, the ego executes the plan until obstructed by obstacles from new observations and predictions. In the case of multiple feasible plans, the ego agent would choose the path with the lowest cost $c_{\text{park}} \in \mathbb{R}$, computed as in [20].

2) *Exploration*: If no safe parking plan is found, our ego considers exploring the parking lot at the exploration points $(\mathbf{s}_{\text{exp},0,q})_{q=1}^{n_{\text{exp},0}}$, where $\mathbf{s}_{\text{exp},0,q} = [\mathbf{p}_{\text{exp},0,q} \ \theta_{\text{exp},0,q}]^T \in \bigcup_{j=1}^{n_{\text{road}}}(C_j \times \{\theta_{R_j}, \theta_{R_j} + \pi\})$, are defined as the intersection points between the center lines of the road $(C_j)_{j=1}^{n_{\text{road}}}$ and the boundary of the sensing region $\partial Q_{0,t}$.

We compute and select the lowest-cost paths with Hybrid A* similar to Section IV-D.1. In addition, we condition that our ego to first consider exploration points in front ($[\cos(\theta_{V_0,t}) \ \sin(\theta_{V_0,t})] (\mathbf{p}_{\text{exp},0,q} - \mathbf{p}_{V_0,t}) \geq 0$) before considering those behind ($[\cos(\theta_{V_0,t}) \ \sin(\theta_{V_0,t})] (\mathbf{p}_{\text{exp},0,q} -$

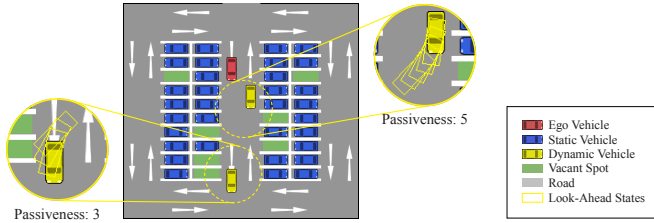


Fig. 3: Setup of our experiment at $t = 0$. At each iteration, some of the bottom 10 spots are randomly chosen to be vacant. Then, 1 or 2 dynamic vehicles would spawn, with pots being their goal. There are always 2 vacant spots on the sides in case the ego failed to compete.

$\mathbf{p}_{V_0,t} < 0$). In practice, the ego would only consider exploring backwards if the road ahead of it stays blocked for the foreseeable future.

3) *Idling*: If no safe plans were found in both Sections IV-D.1 and IV-D.2, the ego would stay put at the current position for this timestep (i.e. $\mathbf{u}_{V_0,t} = \mathbf{0}$) to wait for the other agents to move out of the ego’s way.

V. EXPERIMENTS

We now compare our method, which predicts intention from motion history, with methods that infer intentions from trajectory predictions or implicitly in an end-to-end framework, as summarized in Figure 2. We also evaluate alternative trajectory prediction methods against our cubic Bézier baseline from Section IV-C. All experiments were performed using Python on a laptop computer with a 24-core i9 CPU, 64 GB RAM, and an NVIDIA RTX 4090 GPU.

A. Environment Setup

We conduct our experiments in a small parking lot with four columns of 10 spots, divided by 3 vertical and 2 horizontal roads. At each iteration, between 1 and 10 of the bottom 10 spots in the 2 middle columns are vacant, depending on whether 1 or 2 dynamic vehicles are present. 2 additional spots on the outer columns are also vacant in case the ego fails to compete for the bottom spots. The ego agent always starts at the top of the middle vertical road, facing downward. We illustrate this setup in Figure 3.

At the start of each iteration, 1 or 2 dynamic vehicles spawn, each randomly assigned a vacant spot at the bottom. Their spawn locations and parking maneuvers are chosen randomly from the 8 maneuvers defined in [19], which are characterized by whether the vehicle is in the closer or the further lane, whether it starts before or after the spot, and whether it parks head-in or tail-in (see [19, Figure 4]). If these predefined plans collide with each other, a new target spot and a parking maneuver are randomly chosen again.

All scenarios were ran with the hyperparameters $t_f = 100$ s, $Q_0 = \{\mathbf{p} \mid \|\mathbf{p}\|_2 \leq 11.5 \text{ m}, \mathbf{p} \in \mathbb{R}^2\}$, $n_{\text{ray}} = 360$, $t_{\text{hist}} = 4$ s, $\ell_{\text{def}} = 4.97$ m, $w_{\text{def}} = 1.86$ m, $\beta = 0.5$, $\mu = 0.3$, $\zeta = 3$ s, and $\delta = 0.3$. Any unmentioned variables were chosen to match those in [20] for fair comparison. Note that all compared methods only differs in how they choose the ego’s parking spot; exploration, pedestrian interaction, sensing, and other components were kept constant. As such,

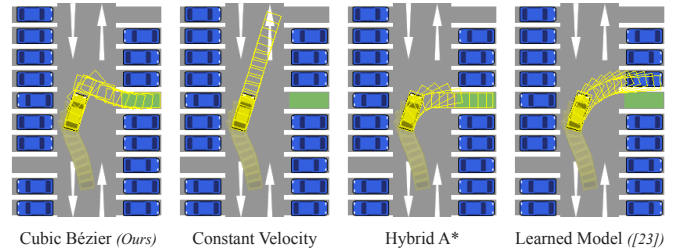


Fig. 4: Example trajectory predictions (yellow outline) conditioned on the predicted intention (green spot) which we compared in our experiment.

the experiments were conducted in a small parking lot with no pedestrians to best observe the effects of the different methods.

B. Comparison Setup

To gauge the methods’ ability to predict and avoid collision, we randomized 500 different environment setups as in Section V-A, and simulate them with reactive agents with passiveness chosen uniformly from 2 to 6. To assess how the methods perform under more challenging scenarios, we also ran simulations in the same 500 setups with non-reactive vehicle agents (i.e. $n_{\text{pass},V_1} = \dots = n_{\text{pass},V_{n_{\text{other}}}} = 0$).

We compare the methods under a variety of evaluation metrics. To measure task completion, we recorded success rate, where a task is considered unsuccessful if collision occurred or the ego failed to arrive at a goal before t_f . To measure social acceptance, we recorded the number of iterations where the ego has “stolen” a spot originally targeted by a dynamic vehicle, as well as the average number of timesteps interrupting the dynamic agents’ predefined plans. Since all dynamic agents were initialized closer to their intended spots, any takeover by the ego represents a true case of spot stealing. To measure the quality of the resulting trajectories, we recorded t_{park} , the average time the ego takes to park into a goal. To measure computation efficiency, we recorded the average time it took for an ego to select the goal at each timestep (since each method differs in how they select spots), and the average time it took to plan for a trajectory after selecting the goal (which we expect to be the same). Finally, to measure the accuracy of the trajectory prediction, we recorded the average minimum average displacement error (minADE) and the minimum final displacement error (minFDE) [22] of each episode.

We compared our method against different trajectory conditioning methods, [20], and an implementation of [23]. The implementation details are as follows:

1) *Effects of Trajectory Prediction Method*: To validate our choice of using cubic Bézier curve in Section IV-C, we replaced it with a constant velocity model, with Hybrid A*, and with the learned trajectory prediction model from [23] (a transformer-based model) to predict the trajectories of each other agent conditioned on the predicted intent location. Some example trajectories are shown in Figure 4.

2) *Effects of Inferring Intentions from Trajectory Prediction*: We compared against [20] to investigate the effect of inferring intentions from the future vs. the past and to

evaluate its performance under more realistic assumptions. Due to the lack of non-intention-driven trajectory prediction methods for parking agents, we replaced their ground truth trajectory prediction with a constant velocity model.

3) *Effects of Implicit Intent Reasoning*: To investigate whether explicit reasoning of intentions are beneficial, we implemented [23] by applying the intention prediction model on *the ego only*. At each timestep, we apply the intention prediction model on the ego, and constant velocity trajectory prediction on other agents (for the same reason as Section V-B.2). The ego would then prioritize spots with the highest intention probability that it can find a safe plan to.

C. Results and Discussion

The results of the experiments are shown in Table I. For non-reactive agents, our proposed method has a significantly higher success rate ($\approx+9\%$) than [20] and [23], possibly due to our better accuracy in trajectory prediction, as seen from the lower average minADE (≈-1 m) and minFDE (≈-3 m). Since our environment and agent behavior were not from the DLP dataset [23], this shows that the intention prediction model generalizes to our setup.

For reactive agents, since agents no longer actively seek out collision, our method’s improvement in success rate against [20] and [23] is less ($\approx+1\%$), but we perform better in social acceptance, stealing less spots ($\approx-4\%$) and interrupting the plans of the other agents less (≈-0.2), which we attribute to the improved accuracy in trajectory prediction.

Of the four trajectory prediction methods in our proposed AVP pipeline, intention-conditioned cubic Bézier curve consistently perform better in task completion, social acceptance, and computation time across both non-reactive and reactive agents. Surprisingly, it generated trajectories with lower minADE and minFDE than those from constant velocity and Hybrid A*, and with comparable results with the learned model in [23], which may explain its better performance.

In general, all methods produce trajectories of similar quality when comparing their average parking time. That said, our proposed method suffers from a higher computation time in spot selection ($\approx+0.005$ s) compared to [20] and [23], likely due to our need to reconstruct the semantic BEV map and predict others’ intents with neural network inference. However, this increase is quite insignificant compared to the time needed for generating the paths, which takes $\approx 1-1.5$ s to compute. While such planning times fall short of the 10 Hz cycle typically desired for deployment, our pipeline is modular and can readily incorporate faster trajectory planners: whether learning-based [15], optimization-based [16], reachability-based [17], or sampling-based [18]. Since our study centers on the role of intention modeling and spot selection, we leave planner comparisons and computation optimizations to future work.

VI. CONCLUSION

This work investigated the role of intention prediction on the AVP problem. By proposing a new AVP pipeline and conducting extensive experiments in a realistic simulation

environment, we conclude that explicit intention prediction from motion history is beneficial to selecting spots in a safer, more socially acceptable manner compared to intention prediction from trajectory models and implicit, end-to-end reasoning of intentions. We argue that this is because explicit reasoning of intentions are necessary to handle the diverse and ambiguous long-term goals in parking, but the maneuvers executed to achieve these intentions are often too complex to be reliably inferred from short-term cues (such as using constant velocity models) alone. We speculate that this finding may also be applicable to robotic social navigation with similar conditions, such as service robots operating in shared indoor spaces, or autonomous delivery systems with docking stations in public space.

Limitations: We observed three limitations of our method. Firstly, our current implementation does not yet achieve real-time performance, as Hybrid A* dominates computation for path planning. However, because the pipeline is modular, it could readily incorporate faster planners in future work. Secondly, our experiments were performed on simulations with predefined agent paths rather than real humans, which may limit the ecological validity of our findings. While we are preparing human-in-the-loop simulations and real-vehicle experiments, they must be preceded by achieving real-time computation. Finally, our intention model does not yet reason with pedestrians, vehicles exiting from their spots, or agent types such as bicycles, motorcycles, and trucks. This limitation is carried over from the literature that we use for comparison [23]. Towards this, we are working on constructing and training our own model from the DLP dataset [23].

REFERENCES

- [1] C. Dong, J. M. Dolan, and B. Litkouhi, “Intention estimation for ramp merging control in autonomous driving,” in *2017 IEEE intelligent vehicles symposium (IV)*, IEEE, 2017, pp. 1584–1589.
- [2] J. Schmidt, J. Jordan, D. Raba, T. Welz, and K. Dietmayer, “Meat: Maneuver extraction from agent trajectories,” in *2022 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2022, pp. 1810–1816.
- [3] D. Park, H. Ryu, Y. Yang, J. Cho, J. Kim, and K.-J. Yoon, “Leveraging future relationship reasoning for vehicle trajectory prediction,” *arXiv preprint arXiv:2305.14715*, 2023.
- [4] K. D. Katyal, G. D. Hager, and C.-M. Huang, “Intent-aware pedestrian prediction for adaptive crowd navigation,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 3277–3283.
- [5] P. Ratsamee, Y. Mae, K. Kamiyama, M. Horade, M. Kojima, and T. Arai, “Social interactive robot navigation based on human intention analysis from face orientation and human path prediction,” *Robomech Journal*, vol. 2, no. 1, p. 11, 2015.
- [6] M. Bojarski, D. Del Testa, D. Dworakowski, *et al.*, “End to end learning for self-driving cars,” *arXiv preprint arXiv:1604.07316*, 2016.
- [7] A. Li, S. Bae, D. Isele, R. Beeson, and F. M. Tariq, “Predictive planner for autonomous driving with consistency models,” in *2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2025.
- [8] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, “End-to-end driving via conditional imitation learning,” in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 4693–4700.
- [9] Y. Hu, J. Chen, K. Zhou, H. Gao, Y. Li, and C. Liu, “Informative path planning of autonomous vehicle for parking occupancy estimation,” in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2023, pp. 3304–3310.

TABLE I: Results of our proposed method with different trajectory prediction methods, a method that predicts intentions from trajectory prediction [20], and a method that reasons with intentions implicitly [23] for 500 parking simulations with non-interactive agents and 500 simulations with interactive agents. The best results in each evaluation metric (\uparrow : higher is better, \downarrow : lower is better) are highlighted.

Method		Success Rate (%) \uparrow	Social Acceptance		Computation Time (s) \downarrow		Parking Time (s) \downarrow	Accuracy	
			Spots Stolen (%) \downarrow	Interrupted Timesteps \downarrow	Spot Selection	Path Planning		minADE (m) \downarrow	minFDE (m) \downarrow
Non-Reactive Agents									
Explicit, Past	Cubic Bézier (Ours)	99.8	0.0	0.000 ± 0.000	0.0052 ± 0.0026	1.2812 ± 0.2516	20.1 ± 9.6	2.02 ± 1.22	2.40 ± 1.89
	Constant Velocity	92.0	0.2	0.000 ± 0.000	0.0058 ± 0.0031	1.2848 ± 0.2479	19.3 ± 9.2	3.09 ± 1.33	5.59 ± 2.32
	Hybrid A*	94.6	0.0	0.000 ± 0.000	0.7360 ± 0.3257	1.3435 ± 0.2825	20.1 ± 9.7	2.46 ± 1.39	3.81 ± 2.04
	Learned Model ([23])	96.4	0.2	0.000 ± 0.000	0.0535 ± 0.0278	1.3476 ± 0.2717	20.9 ± 10.5	1.82 ± 1.16	2.46 ± 1.63
Implicit ([23])		90.4	0.2	0.000 ± 0.000	0.0006 ± 0.0000	2.9143 ± 1.5890	21.1 ± 8.8	2.77 ± 1.30	5.00 ± 2.30
Explicit, Future ([20])		90.8	0.0	0.000 ± 0.000	0.0010 ± 0.0001	1.5907 ± 0.4023	19.4 ± 9.4	3.13 ± 1.35	5.64 ± 2.33
Reactive Agents									
Explicit, Past	Cubic Bézier (Ours)	100.0	0.0	0.072 ± 0.599	0.0052 ± 0.0027	0.7564 ± 0.1698	20.4 ± 9.4	2.02 ± 1.22	2.39 ± 1.88
	Constant Velocity	99.0	2.8	0.870 ± 8.316	0.0057 ± 0.0031	0.7621 ± 0.1796	19.2 ± 8.8	2.98 ± 1.23	5.38 ± 2.18
	Hybrid A*	98.4	2.8	0.672 ± 8.216	0.4274 ± 0.2041	0.7966 ± 0.1907	20.0 ± 9.1	2.39 ± 1.28	3.75 ± 1.92
	Learned Model ([23])	99.0	4.4	0.824 ± 11.032	0.0529 ± 0.0266	0.8062 ± 0.2205	20.7 ± 10.1	1.80 ± 1.16	2.46 ± 1.64
Implicit ([23])		98.6	3.4	0.208 ± 1.022	0.0006 ± 0.0000	1.6834 ± 0.8825	20.9 ± 8.5	2.52 ± 1.01	4.59 ± 1.86
Explicit, Future ([20])		98.8	4.6	0.408 ± 7.334	0.0010 ± 0.0001	1.0238 ± 0.3210	19.2 ± 9.0	2.90 ± 1.16	5.28 ± 2.06

- [10] J.-L. Lupien, A. Alhadlaq, Y. Tang, *et al.*, "Entropy-based dynamic programming for efficient vehicle parking," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2024, pp. 2793–2798.
- [11] F. Baldini, F. M. Tariq, S. Bae, and D. Isele, "Don't get stuck: A deadlock recovery approach," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2024, pp. 3688–3695.
- [12] X. Shen and F. Borrelli, "Multi-vehicle conflict resolution in highly constrained spaces by merging optimal control and reinforcement learning," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 3308–3313, 2023.
- [13] X. Shen and F. Borrelli, "Reinforcement learning and distributed model predictive control for conflict resolution in highly constrained spaces," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2023, pp. 1–6.
- [14] X. Shen, E. L. Zhu, Y. R. Stürz, and F. Borrelli, "Collision avoidance in tightly-constrained environments without coordination: A hierarchical control approach," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 2674–2680.
- [15] M. Jiang, Y. Li, S. Zhang, S. Chen, C. Wang, and M. Yang, "Hope: A reinforcement learning-based hybrid policy path planner for diverse parking scenarios," *IEEE Transactions on Intelligent Transportation Systems*, 2025.
- [16] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, "Autonomous parking using optimization-based collision avoidance," in *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 4327–4332.
- [17] X. Chi, J. Zeng, J. Huang, Z. Liu, and H. Su, "Fast path planning for autonomous vehicle parking with safety-guarantee using hamilton-jacobi reachability," *arXiv preprint arXiv:2310.15190*, 2023.
- [18] F. Nawaz, M. Sung, D. Gadginmath, *et al.*, "Graph-based path planning with dynamic obstacle avoidance for autonomous parking," *arXiv preprint arXiv:2504.12616*, 2025.
- [19] X. Shen, Y. Choi, A. Wong, F. Borrelli, S. Moura, and S. Woo, "Parking of connected automated vehicles: Vehicle control, parking assignment, and multi-agent simulation," *arXiv preprint arXiv:2402.14183*, 2024.
- [20] F. Nawaz, F. M. Tariq, S. Bae, *et al.*, "Occupancy-aware trajectory planning for autonomous valet parking in uncertain dynamic environments," *arXiv preprint arXiv:2509.09206*, 2025.
- [21] P. Sun, H. Kretschmar, X. Dotiwalla, *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.
- [22] M.-F. Chang, J. Lambert, P. Sangkloy, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8748–8757.
- [23] X. Shen, M. Lacayo, N. Guggilla, and F. Borrelli, "Parkpredict+: Multimodal intent and motion prediction for vehicles in parking lots with cnn and transformer," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2022, pp. 3999–4004.
- [24] T. Ji, C. Zhao, Y. Ji, and Y. Du, "A two-stage framework for parking search behavior prediction through adversarial inverse reinforcement learning and transformer," *Expert Systems with Applications*, vol. 255, p. 124548, 2024.
- [25] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific journal of mathematics*, vol. 145, no. 2, pp. 367–393, 1990.
- [26] A. Shamsah, K. Agarwal, N. Katta, A. Raju, S. Kousik, and Y. Zhao, "Socially acceptable bipedal robot navigation via social zonotope network model predictive control," *IEEE Transactions on Automation Science and Engineering*, 2024.
- [27] M. E. Mortenson, *Mathematics for computer graphics applications*. Industrial Press Inc., 1999.