2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Occupancy-aware Trajectory Planning for Autonomous Valet Parking in Uncertain Dynamic Environments

Farhad Nawaz*[1,2], Faizan M. Tariq[1], Sangjae Bae[1], David Isele[1], Avinash Singh[1], Nadia Figueroa[2], Nikolai Matni[2] and Jovin D'sa*[1]

Abstract— **Autonomous Valet Parking (AVP) requires planning under partial observability, where parking spot availability evolves as dynamic agents enter and exit spots. Existing approaches either rely only on instantaneous spot availability or make static assumptions, thereby limiting foresight and adaptability. We propose an approach that estimates probability of future spot occupancy by distinguishing initially vacant and occupied spots while leveraging nearby dynamic agent motion. We propose a probabilistic estimator that integrates partial, noisy observations from a limited Field-of-View, with the evolving uncertainty of unobserved spots. Coupled with the estimator, we design a strategy planner that balances goal-directed parking maneuvers with exploratory navigation based on information gain, and incorporates wait-and-go behaviors at promising spots. Through randomized simulations emulating large parking lots, we demonstrate that our framework significantly improves parking efficiency and trajectory smoothness over existing approaches, while maintaining safety margins. Simulation videos: https://sites.google.com/view/avp-hri/home.**

I. INTRODUCTION

In busy public spaces, parking consumes significant time, space, and fuel [1], [2] as drivers loop through parking lots searching for spots, interacting with other drivers and pedestrians, and performing tight maneuvers. While existing park assist systems can steer a car into a user-selected spot [3], [4], a Type-1 autonomous valet parking system [5] that operates without infrastructure support remains unavailable. To this end, we propose a framework for Type-1 AVP that integrates an observation model tailored for onboard sensing with unified spot selection and trajectory planning.

Several trajectory planners for autonomous parking have been proposed using optimization and learning-based approaches. Some optimization-based methods often focus on static obstacle avoidance [6], [7], while dynamic obstacle avoidance approaches [8], [9] typically rely on predefined reference paths. Time-optimal formulations use dynamic optimization [10], [11] to reduce parking time but require high computational effort, whereas heuristic planners that use Reeds–Shepp curves [12] trade optimality for faster real-world execution. Learning-based approaches leverage deep neural networks [13], [14] to generate diverse trajectories in highly constrained and partially unknown environments, but fail to guarantee safety. While these approaches demonstrate effective maneuver execution, they generally assume predefined spot availability and decouple spot selection from trajectory planning.

Methods that focus on parking spot selection often make simplifying assumptions that limit practical applicability. Some approaches ignore spot occupancy prediction [15], while others assume full connectivity between vehicles [16]. Reinforcement learning strategies [17], [18] optimize parking space allocation and reduce search time, but struggle to generalize and guarantee real-time safety. Vanilla Bayesian filtering methods [19] typically neglect the influence of dynamic agents on spot availability, and their observation models [15], [19] assume fixed sensor confidence within the Field-of-View (FoV), overlooking the practical degradation of perception reliability with distance. Exploration-driven strategies [19], [20] focus on information gain but often disregard the downstream task of actual parking. For instance, [20] uses a simplified entropy-based occupancy model that ignores vehicle dynamics, limiting its suitability in dynamic environments. Connected-vehicle frameworks [16], [21] assume Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, which are effective for offline spot allocation, but do not address the real-time strategic adjustments required in infrastructure-free scenarios.

To address these gaps, we propose an occupancy-aware trajectory planning framework tailored for Type-1 AVP:

- **Distance-aware partial FoV model**: We model partial observability by introducing a distance-dependent FoV model that maps range to observation confidence.
- **Spot occupancy estimator:** We propose a Bayes filter based approach that predicts the future occupancy of parking spots using the behavior of dynamic agents. We also model distinct arrival and departure equations for initially vacant and occupied spots, respectively, to reason about the future occupancy.
- **Strategy planner:** We integrate future spot occupancy probabilities into a cost-based policy that plans the ego vehicle's trajectory to either explore the parking lot or commit to a spot. The policy allows the ego to pause near promising spots by incorporating a waiting-time cost, enabling it to capture imminent vacancies while maintaining a balance between efficiency and safety.

We evaluate our approach in a simulated parking lot, combining foresighted occupancy reasoning and strategy planning. Our method reduces time and distance traveled while generating smoother and safer paths compared to existing spot selection methods and trajectory planners [16], [19].

[1]Honda Research Institute (HRI), San Jose, CA 95134, USA.
[2]GRASP Lab, University of Pennsylvania, PA 19104, USA.
*Corresponding authors: farhadn@seas.upenn.edu, jovin_dsa@honda-ri.com. All work was done when Farhad Nawaz was employed by HRI.

979-8-3315-8160-2/26/$31.00 ©2026 IEEE

II. Problem Formulation and Approach

We consider an ego vehicle navigating a parking lot, equipped with onboard sensors that provide observations of spot occupancy within a limited FoV, along with motion predictions of dynamic agents such as vehicles and pedestrians. The task for the ego is to plan trajectories that efficiently explore the lot, identify an available spot, and park safely.

A. Parking Lot Environment

Consider the parking lot in Fig. 1 with N_p spots, whose centers are given by $\mathcal{P} = \{s_1, \dots, s_{N_p}\}$, where s_i denotes the 2D coordinates of spot i. The ego vehicle has prior access to a static map of the lot (boundaries and lane centerlines/widths), but spot occupancy is unknown and must be observed in real time. Boundaries are treated as static obstacles, and vehicle dynamics are modeled using the kinematic bicycle model [22], that is suitable for low-speed maneuvers. The vehicle state is $x = [X, Y, \theta]^\top$, where (X, Y) is the rear axle center and heading is θ, while controls are longitudinal velocity v and front-wheel steering angle δ.

B. Observation Model

The vehicle's limited onboard sensing is modeled via a "scaled and shifted" infinity norm defining the FoV, though other shapes (e.g., ellipses or circles) can be modeled by modifying the *distance metric*. This formulation captures the limited and imperfect nature of real-world observations.

Definition 1 (Distance metric)**.** The scaled and shifted distance between the ego vehicle at state $x = \begin{bmatrix} X & Y & \theta \end{bmatrix}^\top$ and a point $s \in \mathbb{R}^2$ is defined as

$$d_{(\zeta,r)}(x,s) = \left\| r^\top R(\theta)^\top \left(s - \left(\begin{bmatrix} X \\ Y \end{bmatrix} + R(\theta) \begin{bmatrix} \zeta \\ 0 \end{bmatrix} \right) \right) \right\|_\infty,$$
$$\text{such that } r = \begin{bmatrix} \frac{1}{r_x} & \frac{1}{r_y} \end{bmatrix}^\top, \; R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \tag{1}$$

where r_x and r_y are the half-lengths of the FoV rectangle along the longitudinal and lateral directions, respectively, while ζ is the forward shift in ego frame to bias the sensing region ahead of the vehicle. The rotation matrix $R(\theta)$ transforms vectors from the ego frame to the world frame.

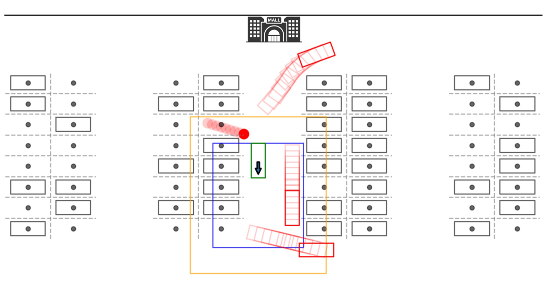


Fig. 1: Autonomous ego vehicle (green) with limited field of view (blue/orange) navigating an environment simulating a mall parking lot, with the entrance at the top. Predicted motions of dynamic vehicles are shown as red rectangles, and a pedestrian is indicated by a red circle. Spots are fully observed inside blue rectangle ($\epsilon = 1$), unobserved outside orange ($\gamma = 1.5$), and partially observed in between. Static vehicles are black, and parking spot centers are grey dots.

Definition 2 (Spot observability)**.** The *observability* of a spot $s \in \mathcal{P}$ from vehicle state x is defined as:

$$\text{Spot } s \text{ is} \begin{cases} \text{fully observed if} & d_{(\zeta,r)}(x,s) \leq \epsilon, \\ \text{unobserved if} & d_{(\zeta,r)}(x,s) \geq \gamma, \\ \text{partially observed} & \text{otherwise}, \end{cases} \tag{2}$$

where $\epsilon < \gamma$ and are obtained from the vehicle sensing limits.

The different observability regions are illustrated in Fig. 1.

Definition 3 (Observation accuracy)**.** Let $O_t^s \in \{0, 1\}$ be the ground-truth occupancy and $Z_t^s \in \{0, 1\}$ be the observed occupancy of spot s at time t, where $1 :=$ occupied and $0 :=$ vacant. Observation accuracy is the likelihood of correctly observing spot s, modeled using the distance-dependent probability

$$p_c(d_{(\zeta,r)}(x,s)) = P(Z_t^s = 1 | O_t^s = 1) = P(Z_t^s = 0 | O_t^s = 0),$$
$$p_c(d_{(\zeta,r)}(x,s)) = e^{-\ln(2)\left(1+e^{-\alpha_c\left(d_{(\zeta,r)}(x,s)-\left(\frac{\gamma-\epsilon}{2}+\epsilon\right)\right)}\right)^{-1}}, \tag{3}$$

where $p_c(d)$ is the exponential of a scaled sigmoid centered at $\frac{\gamma-\epsilon}{2} + \epsilon$, such that $p_c(d) \to 1$ as d decreases to ϵ and $p_c(d)$ decreases to 0.5 as $d \to \gamma$.The probability (3) models observation confidence degrading with distance from ego vehicle, unlike prior work [19] that assumes fixed uncertainty.

The formal problem statement is given below.

Problem 1. Given the ego vehicle state $x_t \in \mathbb{R}^3$ at time t, parking spot observations Z_t^s as defined in Section II-B, predictions of D dynamic agents $\{\{\hat{y}^j(k)\}_{k=t}^{t+T}\}_{j=1}^{D}$ over a horizon T, where $\hat{y}^j(k) \in \mathbb{R}^3$ is the position and heading of agent j if it is a vehicle, and $\hat{y}^j(k) \in \mathbb{R}^2$ is the position if agent j is a pedestrian; compute a reference trajectory that satisfies the vehicle dynamics and steers the vehicle toward a feasible parking spot or explore the lot while avoiding static and dynamic obstacles.

Remark: We assume the availability of predicted trajectories of dynamic agents over a horizon T at each planning step t to solve Problem 1, while aiming to develop and integrate a trajectory prediction model in future work.

C. Proposed Approach

Fig. 2 illustrates our approach, which integrates spot occupancy estimation with planning. The perception module provides dynamic agent predictions and occupancy observations using our partial FoV model, and the occupancy estimator recursively updates to predict future spot availability. The strategy planner computes and evaluates feasible paths, corresponding to one of the three actions—*park immediately*, *wait*, or *explore the lot*—by minimizing a cost that trades off efficiency and safety. The chosen trajectory is executed by the vehicle controller, closing the loop.

III. Spot Occupancy Estimator

At each planning step t, the spot occupancy estimator predicts occupancy probabilities for all spots in the lot over a horizon T, combining current observations with predicted motion of dynamic agents. This enables the ego vehicle to

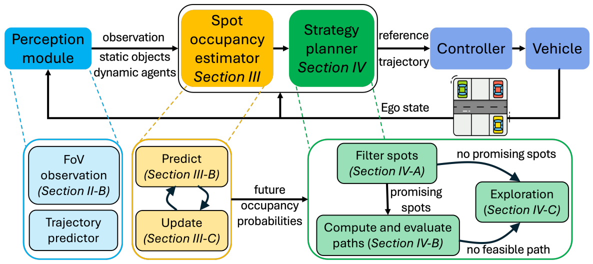


Fig. 2: Outline of our proposed approach.

plan based on anticipated occupancies rather than instantaneous observations. Probabilities are computed via a Bayes filter [23], with the *prediction* step modeling agent interactions and the *update* step incorporating FoV observations.

A. Effect of Dynamic Agents

Each agent j's position at time k is modeled as a Gaussian:

$$y^j(k) \sim \mathcal{N}(\hat{y}^j(k), \Sigma(k)), \quad k \in \{t, \dots, t+T\}, \tag{4}$$

where $\hat{y}^j(k)$ is the nominal prediction and $\Sigma(k)$ is the covariance. These uncertain trajectories define the probability that a spot $s \in \mathcal{P}$ is occupied by an agent j at time k:

$$p(s, j, k) = E_d \cdot A_f, \tag{5}$$

where

$$E_d = \left((\alpha_d + 1)(\alpha_d + e^{\alpha_1 m_d(y^j(k), s)})\right)^{-1} \tag{6}$$

captures spatial proximity via the Mahalanobis distance [24]:

$$m_d(y^j(k), s) = (\hat{y}^j(k) - s)^T \Sigma(k)^{-1} (\hat{y}^j(k) - s)$$

and

$$A_f = \left(1 + e^{-\alpha_2 \sigma_v(k)^{-1} \hat{v}^j(k)^T (s - \hat{y}^j(k))}\right)^{-1} \tag{7}$$

is an alignment factor that reflects whether the agent is moving toward the spot, based on its predicted velocity $\hat{v}^j(k)$ and velocity uncertainty $\sigma_v(k)$. Together, E_d and A_f encode the intuition that an agent is more likely to occupy a spot if the agent is both closer and moving towards the spot, while accounting for motion uncertainty.

Uncertainty Propagation: We propagate the positional covariance at each timestep using the discrete-time model:

$$\Sigma(k+1) = R(\theta(k)) \left(\Sigma(k) + Q\right) R(\theta(k))^\top, \tag{8}$$

where $R(\theta(k))$ is the rotation matrix, $\theta(k)$ is vehicle's heading and Q is a diagonal matrix representing process noise in ego frame. To estimate the uncertainty in velocity, we use the finite difference $v^j(k) = \frac{y^j(k+1) - y^j(k)}{\Delta t}$ and (8).

From (5), the probability that at least one of the D dynamic agents will occupy spot s at time k is:

$$q(s, k) = 1 - \prod_{j=1}^{D} \left(1 - p(s, j, k)\right). \tag{9}$$

At each step $k \in \{t, \dots, t+T\}$, we predict the prior belief $\bar{b}_k^s = P(O_k^s = 1)$ using (9), and update it with the observations Z_k^s to obtain the posterior belief $b_k^s = P(O_k^s = 1 \mid Z_k^s)$, as described in Section III-B.

B. Prediction

Prior work [19] propagates beliefs without considering the effect of dynamic agents on parking spots. We explicitly incorporate dynamic agent motion and model asymmetric interaction with spots: the ego cares about a currently occupied spot becoming vacant, while it is concerned about a vacant spot becoming occupied in the future. This distinction enables faster parking and smoother trajectories, as shown in Section V-A.1. The current occupancy of a spot is determined from its observation Z_t^s. No occupancy observations and agent interactions are available outside the FoV.

1) For initially **vacant** spots, our transition model emphasizes incoming occupancy, and the prior belief is recursively updated as

$$\bar{b}_{k+1}^s = q(s, k+1)(1 - b_k^s) + \mu_1 b_k^s, \tag{10}$$

where $q(s, k+1) = P(O_{k+1}^s = 1 | O_k^s = 0)$ is computed from (9) and $\mu_1 = P(O_{k+1}^s = 1 | O_k^s = 1)$ models the departure probability of a parked vehicle using an exponential distribution [25], reflecting the memory-less nature of departures. If λ_d denotes the average departure rate of a parked vehicle, the probability of departure within a time interval Δt is

$$P(O_{k+1}^s = 0 | O_k^s = 1) = 1 - e^{-\lambda_d \Delta t} \Rightarrow \mu_1 = e^{-\lambda_d \Delta t}. \tag{11}$$

2) For initially **occupied** spots, the emphasis is on potential departure of the occupied agent, captured as

$$1 - \bar{b}_{k+1}^s = \mu_2 (1 - b_k^s) + (1 - q(s, k+1)) b_k^s, \tag{12}$$

where $1 - q(s, k+1) = P(O_{k+1}^s = 0 | O_k^s = 1)$ is computed from (9) and $\mu_2 = P(O_{k+1}^s = 1 | O_k^s = 0)$ models the arrival probability of a vacant spot using a Poisson distribution, which commonly describes arrival processes. If λ_a denotes the average number of vehicles that arrive with the time interval Δt, the probability that no vehicle parks in a vacant spot is

$$P(O_{k+1}^s = 0 | O_k^s = 0) = e^{-\lambda_a} \Rightarrow \mu_2 = 1 - e^{-\lambda_a}. \tag{13}$$

3) For **unobservable** spots, the prediction equation depends only on the average departure and arrival rates:

$$\bar{b}_{k+1}^s = \mu_2 (1 - b_k^s) + \mu_1 b_k^s. \tag{14}$$

The prior $\bar{b}_{k+1}^s$ serves as the input to the update step, where it is fused with observations.

C. Update

The update combines current spot observations with $\bar{b}_{k+1}^s$ to estimate future occupancy. For fully observed spots, i.e., $p_c(d_{(\zeta,r)}(x_t, s)) = 1$, we completely trust the current observation and propagate occupancy using only agent predictions. For partially observed or unobserved spots, i.e., $p_c(d_{(\zeta,r)}(x_t, s)) < 1$, we apply Bayes' rule to fuse the observation with the prior belief:

$$b_{k+1}^s = \begin{cases} \bar{b}_{k+1}^s & \text{if } p_c(d_{(\zeta,r)}(x_t, s)) = 1 \\ \frac{P(Z_{k+1}^s | O_{k+1}^s = 1) \bar{b}_{k+1}^s}{P(Z_{k+1}^s)} & \text{otherwise}, \end{cases} \tag{15}$$

where x_t is the current ego state. Since only the current occupancy observation Z_t^s is available, we set $Z_{k+1}^s = Z_t^s$ for all $k \in \{t, \dots, t+T-1\}$. For unobserved spots s, from (3), we have $p_c(d_{(\zeta,r)}(x_t, s)) \approx 0.5$. Hence, using (15), the update equation reduces to $b_{k+1}^s \approx \bar{b}_{k+1}^s$, i.e., the prior belief remains unchanged.

IV. Strategy Planner

Given the probabilistic forecasts from the spot occupancy estimator (Sec. III), the strategy planner selects actions (park, wait, or explore) and generates a reference trajectory that balances efficiency and safety in dynamic parking scenarios.

A. Filter Spots

At each planning step, we filter candidate spots inside the FoV based on the occupancy predictions, considering two cases: (i) initially vacant and remain vacant (ii) initially occupied and become vacant. Formally, spot s is vacant in the future (t+T) if:

$$\begin{cases} s \text{ is } \textbf{initially vacant} \text{ (time } t\text{) and } b_{t+T}^s \leq P_v, \\ s \text{ is } \textbf{initially occupied} \text{ (time } t\text{) and } b_{t+T}^s \leq P_o. \end{cases} \quad (16)$$

This filtering ensures that the strategy planner only considers feasible parking spots, enhancing decision-making efficiency in dynamic environments. We set a spot is more likely to be occupied if initially vacant ($P_v < P_o$) and vice-versa to reflect the asymmetry in availability reasoning.

B. Generate and Evaluate Paths

At planning step t, let $\mathcal{S}_t \subset \mathcal{P}$ denote spots classified as vacant by (16). We generate trajectories $\mathcal{T} = \{\tau_g\}_{g \in \mathcal{S}_t}$ using Hybrid A*[1], where each τ_g leads to a goal state g from the ego state x_t. Each trajectory is defined as $\tau_g = \{x(k)\}_{k=t}^{t+N_g}$, where $x(t) = x_t$, $x(t+N_g) = g$ and N_g is the time required to reach g from x_t. While static obstacles are considered in trajectory generation, dynamic obstacles are handled via the cost function and waiting time parameter, defined in the next section. Trajectory generations and evaluations are parallelized across all goal states g for efficiency.

1) Cost function: The cost function primarily evaluates efficiency (in time and distance), safety (w.r.t. collision avoidance), and smoothness (w.r.t. acceleration and gear changes):

$$\mathcal{C}(\tau_g) = N_g + t_w + \sum_{k=t}^{t+t_w} c_o(x(k), \mathcal{Y}(k)) + \sum_{k=t+t_w+1}^{t+t_w+N_g} c_o(x(k-t_w), \mathcal{Y}(k)) + \sum_{k=t}^{t+N_g-1} c_{\text{smooth}}(x(k)), \quad (17)$$

where

$$c_o(x(k), \mathcal{Y}(k)) = \sum_{j=1}^{D} h(x(k), \hat{y}^j(k)) + \sum_{i=1}^{W} h(x(t), w_i), \quad (18)$$

$$c_{\text{smooth}}(x(k)) = \|a(k)\| + \mathrm{D_C}(v(k), v(k+1)) + \mathbb{1}(\text{reverse}(k)) + \mathbb{1}(\text{change\_gear}(k)), \quad (19)$$

[1]Alternatives include sampling or spline-based planners [7], [26], [27].

t_w denotes the *waiting time* (explained in the next section), $\mathcal{Y}(k) = \{\hat{y}^j(k)\}_{j=1}^D$ represents the configuration of dynamic agents at time k and the static obstacles are given by $\{w_i\}_{i=1}^W$ with $w_i \in \mathbb{R}^2$. At each time step, $c_o(\cdot,\cdot)$ penalizes proximity to obstacles and $c_{\text{smooth}}(\cdot)$ promotes smooth trajectories.

In $c_o(\cdot,\cdot)$, the obstacle avoidance cost $h(\cdot,\cdot)$ is

$$h(x, y) = e^{-\alpha_o d_o(x,y)}, \quad (20)$$

where $d_o(x, y)$ is the distance between the ego vehicle x and obstacle y. The barrier cost $h(\cdot,\cdot)$ penalizes proximity to obstacles, with $\alpha_o = 2$ for dynamic agents and $\alpha_o = 3$ for static obstacles, reflecting lower penalties for static ones. We extend the 3-circle distance from [28] to a 5-circle model for $d_o(\cdot,\cdot)$ to improve accuracy in dynamic collision avoidance. For static obstacles, the exact distance between the ego vehicle's edge and obstacle points $\{w_i\}_{i=1}^W$ is computed as given in Section III of [29], since static obstacles are more reliably perceived. To balance terms, $\|a(t)\|$ and the cosine distance are scaled to $[0, 1]$ using velocity limits.

In (19), $\mathrm{D_C}(\cdot,\cdot)$ is the cosine distance and

$$a(k) = \left[\ddot{X}(k), \ddot{Y}(k)\right]^\top, \; v(k) = \left[\dot{X}(k), \dot{Y}(k)\right]^\top,$$

$$\mathbb{1}(\text{reverse}(k)) = \begin{cases} 1, & \text{if } \mathrm{D_C}\left(v(k), \begin{bmatrix}\cos(\theta(k)) \\ \sin(\theta(k))\end{bmatrix}\right) > 1, \\ 0, & \text{otherwise} \end{cases},$$

$$\mathbb{1}(\text{change\_gear}(k)) = \begin{cases} 1, & \text{if } \mathrm{D_C}\left(v(k), v(k+1)\right) > 1, \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

2) Waiting time: We incorporate *wait-and-go* behaviors in the trajectories τ_g by introducing a waiting time t_w in (17). This enables the ego vehicle to pause and yield to dynamic agents before moving to park in a spot g, thereby reducing collision risks in constrained parking lots. To ensure feasibility, we impose a maximum wait time T_w and consider only paths with $t_w \leq T_w$. The value of T_w is set based on parking lot density, with future work aimed at dynamically adapting it to congestion.

Remark: If $T < N_g$, we extrapolate the predictions using a constant velocity model: $\hat{y}^j(k) = \int_{s=t+T}^{k} \hat{v}^j(t+T-1)ds$ for all $k \geq t+T+1$ and $j \in \{1, 2, \dots, D\}$, where $\hat{v}^j(k) = \frac{\hat{y}^j(k+1) - \hat{y}^j(k)}{\Delta t}$.

Once we compute and evaluate the trajectories $\mathcal{T}$, we choose the one that has the minimum cost $\arg\min_{\tau \in \mathcal{T}} \mathcal{C}(\tau)$ and avoids all obstacles respecting the safety thresholds. We set larger safety thresholds for dynamic agents—50cm for vehicles and 90cm for pedestrians—compared to 20cm for static obstacles. These values are tailored to parking scenarios but can be adjusted for other applications.

C. Exploration

If no feasible trajectory exists to a spot in $\mathcal{S}_t$, the ego vehicle switches to exploratory mode, selecting a trajectory that maximizes information gain about spot occupancy. In typical parking lots like Fig. 1, maneuver options are limited to continuing straight within a row, entering a row from outside, or turning left or right at the end. We generalize this by evaluating paths to a finite set of exploration goals $\mathcal{E} \subset \mathbb{R}^3$, where each $e \in \mathcal{E}$ represents a candidate goal pose.

1) Exploration goals: We define three exploration goals $\mathcal{E} = \{e_1, e_2, e_3\}$ at each time step: going straight (e_1), turning left (e_2), and turning right (e_3). These options suit structured parking lots with parallel rows (Fig. 1), though we plan to extend them to more general layouts such as multi-level garages in future work. The straight goal is:

$$e_1 = \begin{bmatrix} R(\theta_t) \begin{bmatrix} \epsilon - \eta \\ 0 \end{bmatrix} + \begin{bmatrix} X_t \\ Y_t \end{bmatrix} \\ \\ \theta_t \end{bmatrix}, \tag{22}$$

where $x_t = [X_t, Y_t, \theta_t]^\top$ is the ego state, ϵ is the full observability threshold (Sec. II-B), and $\eta \in [0, \epsilon)$ denotes how far the ego moves forward. Left and right turn goal states for right-hand traffic convention are described as

$$e_\sigma = \begin{bmatrix} R(\theta_t) \begin{bmatrix} x_{\text{road}} + \sigma \cdot \left(\frac{l_w}{2}\right) \\ \sigma \cdot \epsilon \end{bmatrix} + \begin{bmatrix} X_t \\ Y_t \end{bmatrix} \\ \theta_t + \sigma \cdot \left(\frac{\pi}{2}\right) \end{bmatrix}, \quad \sigma \in \{+1, -1\}, \tag{23}$$

where $e_2 = e_{+1}$(left), $e_3 = e_{-1}$(right), x_{road} is the center of the newly observed road, and l_w its lane width.

2) Information gain: We use Hybrid A* to generate trajectories to e_2 and e_3, and a 5$^{\text{th}}$-order spline to interpolate from x_t to e_1. Let $\mathcal{T}' = \{\tau_e\}_{e \in \mathcal{E}}$ denote the set of trajectories to each exploration goal. The information gain of a trajectory is evaluated using the entropy $\mathcal{H}(\cdot)$ of a spot s at time k:

$$\mathcal{H}(b_k^s) = -b_k^s \log_2(b_k^s) - (1 - b_k^s) \log_2(1 - b_k^s), \tag{24}$$

where $b_k^s = P(O_k^s = 1 | Z_k^s)$. We simulate future observations $\{Z_k^s\}_{k=t+1}^{t+T}$ for each trajectory τ_e and denote $Z_k^s(\tau_e)$ to be the observation of spot s at time k when the ego follows τ_e. Then, we recursively update beliefs $b_k^s(\tau_e)$ using the update rule (15) and the simulated observations $Z_k^s(\tau_e)$ for each τ_e. The total information gain of a trajectory is defined using the entropy reduction of all parking spots when following τ_e:

$$\mathcal{I}(\tau_e) = \sum_{i=1}^{N_p} \left(\mathcal{H}(b_t^{s_i}(\tau_e)) - \mathcal{H}(b_{t+N_e}^{s_i}(\tau_e)) \right), \tag{25}$$

where N_e is the length of $\tau_e = \{x(k)\}_{k=t}^{t+N_e}$. To balance exploration with motion efficiency, we use (17) and (25) to define the exploration cost

$$\mathcal{C}_e(\tau_e) = \mathcal{C}(\tau_e) - \mathcal{I}(\tau_e), \tag{26}$$

and select the optimal trajectory as $\arg\min_{\tau_e \in \mathcal{T}'} \mathcal{C}_e(\tau_e)$, subject to safety thresholds. If no safe trajectory exists, a contingency planner keeps the vehicle stationary or steers it away from adversarial dynamic agents on a collision course.

V. Experimental Results

The ego vehicle's task is to navigate from its initial position to a feasible parking spot, assuming real-time on-board sensor data on static vehicles, driving aisles, and dynamic agents within the FoV. We evaluate our framework in a simulated shopping mall parking lot (Fig. 1), with randomized initial spot occupancy via a binomial distribution

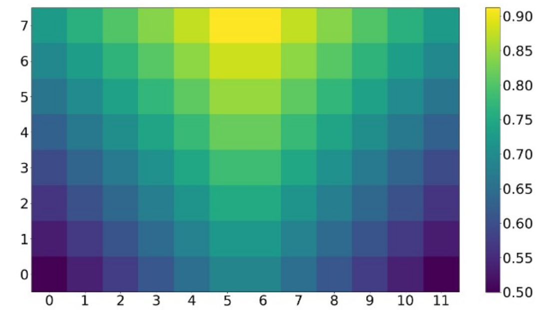


Fig. 3: Initial occupancy probabilities for the parking lot in Fig. 1, consisting of 8 rows and 12 columns of spots.

reflecting higher congestion near the entrance (Fig. 3). We conduct ablation studies and comparisons with existing valet parking methods that are summarized in Table III, with 50 randomized runs per method varying initial occupancy, ego start position, and dynamic agent behavior. The ego is initialized either inside a parking row or outside, while other vehicles may start in vacant spots or outside, with their paths generated by a local Hybrid A* planner for exiting or entering spots, respectively. Dynamic pedestrians are randomly spawned outside the occupied spots with velocities $v \sim \mathcal{U}[-1.5, 1.5]$ m/s. All simulations are run in Python 3.13 on Ubuntu 20.04 with an Intel Xeon E5-2643 v4 CPU.

Simulation parameters: The vehicle dynamics are modeled using the kinematic bicycle model [22], discretized with a time step of 0.1 s for simulation. The vehicle parameters are listed in Table I based on the 2024 Honda Accord [30]. Our Hybrid A* planner employs a state grid resolution of 0.2 m in both X and Y directions, and 10 deg for heading. Steering and velocity inputs are each discretized into 5 values over their limits $[-\delta_{\max}, \delta_{\max}]$ and $[-v_{\max}, v_{\max}]$, respectively. Parking lot parameters in Table II are adopted from [31]. The prediction horizon is $T = 5$ s with a discretization step of 0.1 s, and the maximum wait time is $T_w = 5$ s. The average arrival and departure rates are $\lambda_a = 0.000624$ s^{-1} (13) and $\lambda_d = 0.000378$ s^{-1} (11), respectively, obtained from [25]. The probability thresholds in (16) are $P_v = 0.3$ and $P_o = 0.7$. The scaling factor is $r = \left[(2.5V_L)^{-1} \quad (3V_W)^{-1}\right]^\top$ and forward shift is $\zeta = 0.5V_L$ in (1), while $\eta = 0.5$ m in (22). The observability thresholds are $\epsilon = 1$ and $\gamma = 1.5$, and $\alpha_c = 25$ in (3).

TABLE I: Vehicle parameters

Parameter	Value
Length V_L	4.97 m
Width V_W	1.86 m
Wheelbase L	2.83 m
Speed limit $v_{\max}$	3.5 m/s
Steering limit $\delta_{\max}$	34.9 deg

TABLE II: Parking lot parameters

Parameter	Value
Spot length	6.1 m
Spot width	2.74 m
Road width l_w	7.62 m

A. Ablation Studies

We assess our contributions by performing ablation studies on the spot occupancy estimator and strategy planner. In each case, we fix one module and vary components of the other, comparing the variants with our complete approach.

1) Spot Occupancy Estimator: We fix our strategy planner (Section IV) and compare the performance of three different spot occupancy estimators with ours (Section III).

- **Greedy** approach assumes a horizon of $T = 0.1$ s (one step estimation) in Section III, primarily relying on the current occupancy probabilities.
- **Identical prediction** method applies the same prediction equation (10) regardless of whether a spot is initially vacant or occupied.
- **Position only** approach models dynamic agents without incorporating velocity by removing the alignment factor (7) in (5) so that $p(s, j, k) = E_d$.

Fig. 4 compares our estimator with the Greedy and Identical prediction methods. While Greedy relies only on current occupancy, our approach predicts over a horizon by modeling vacant and occupied dynamics separately, thus lowering occupancy probabilities when a vehicle departs (Fig. 4e) and reducing parking time to 11.5 s versus 16.5 s for others. When the red vehicle passes near the vacant spot V, its occupancy probability temporarily rises but later decreases due to velocity modeling in (5). Although spot V remains vacant, the planner avoids it due to surrounding static vehicles and chooses a more accessible spot.

Fig. 5 compares our estimator with the Position only method, which overestimates the occupancy probability (Fig. 5b) and causes the ego to skip the spot, leading to longer parking times of 24 s versus 11 s for our approach. The ego performs a backup maneuver (Fig. 5c) before briefly waiting for a pedestrian initially outside the FoV (Fig. 5a).

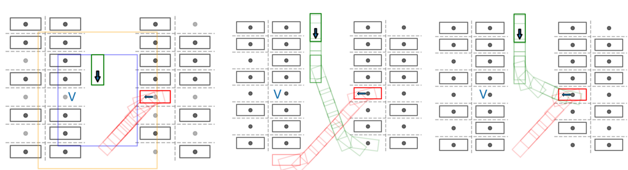
As shown in Table III, our complete approach (row 9) outperforms other estimator variants (rows 1–3), reducing parking time by 22% while maintaining at least 2.3× greater clearance from dynamic agents.

2) Strategy Planner: We fix the spot occupancy estimator (Section III) and compare two strategy planner variants against ours (Section IV):

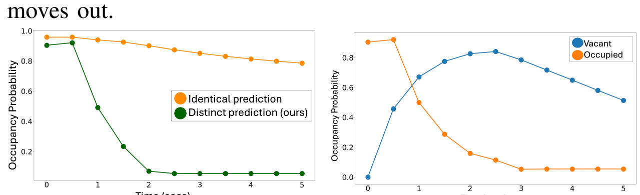
- **No wait time** strategy always sets $t_w = 0$ in (17) so that the ego never pauses near a potentially vacant spot.
- **Lawn-mower** strategy inspired by [32] performs pure exploration ($\mathcal{C}_e(\tau_e) = -\mathcal{I}(\tau_e)$ in (26)) when no promising spots are available as classified by (16).

Fig. 6 compares the no wait time strategy with ours. Although our spot occupancy estimator predicts the initially occupied spot to become vacant (Fig. 4d), the no wait time strategy first explores further before backing up, resulting in a longer parking time of 27 s versus 18 s for our strategy. Our planner balances waiting and exploration, enabling the ego to park in an initially occupied spot (Fig. 6c) through short, bounded waits, which is effective in dynamic environments.

In Fig. 7, the Lawn-mower strategy wastes time exploring a crowded row. As a purely exploration-driven strategy, the ego waits for other vehicles, leading to a longer parking time of 43.5 s (Fig. 7b). Our strategy planner balances exploration

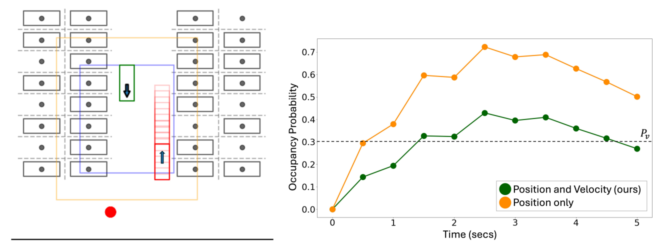


(a) FoV and predictions when red vehicle moves out. (b) Greedy and Identical prediction. (c) Distinct prediction (Ours).

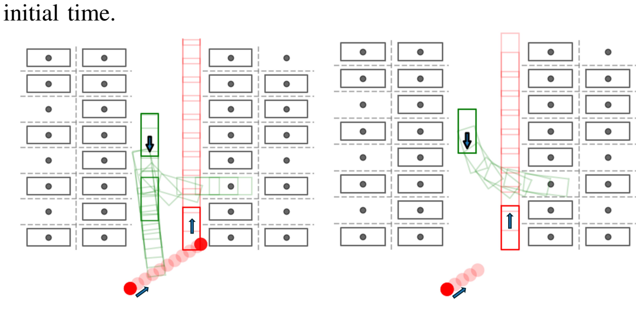


(d) Estimated beliefs of the spot occupied by the red vehicle. (e) Estimated beliefs of spots that are occupied and vacant.

Fig. 4: Comparison of ego vehicle (green) trajectories ((b) and (c)) and occupancy probabilities ((d) and (e)) when a vehicle (red) vacates a spot with a nearby vacant spot (V).

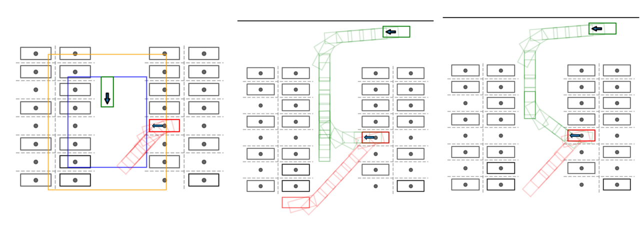


(a) FoV and predictions at initial time. (b) Estimated beliefs of vacant spot.



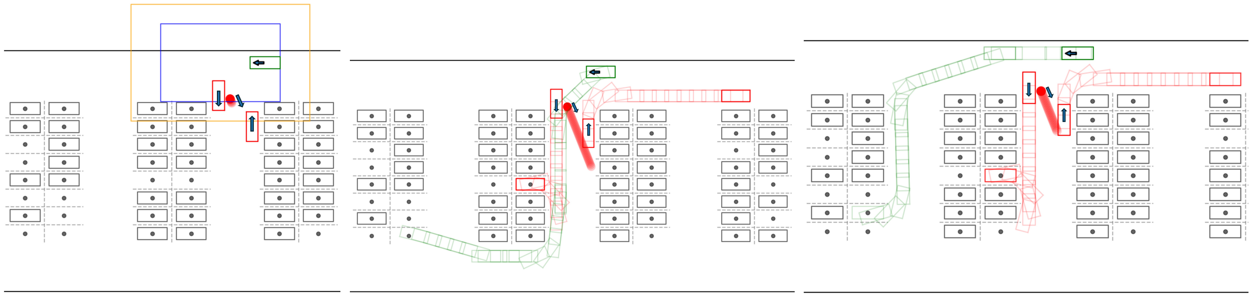
(c) Position only. (d) Position and Velocity (Ours).

Fig. 5: Comparison of ego vehicle (green) trajectories ((c) and (d)) and occupancy probabilities (b) when a vehicle (red rectangle) passes a vacant spot and a pedestrian (red circle) is entering another vehicle.



(a) FoV and predictions when red vehicle moves out. (b) No wait time. (c) Our strategy.

Fig. 6: Comparison of ego vehicle (green) trajectories ((b) and (c)) when there is no wait time in (17) and another vehicle (red) moves out of a spot.



(a) FoV at initial time. (b) Lawn-mower strategy. (c) Our strategy planner.

Fig. 7: Comparison of trajectories ((b) and (c)) when the ego vehicle (green) is initially outside the row of parking spots and a row is crowded with other dynamic agents (red).

and path efficiency, avoiding detours and reducing parking time to 25.5 s (Fig. 7c).

Table III shows that our complete approach (row 9) improves upon other strategy planner variants (rows 4–5), decreasing parking time and path length by at least 39%, while producing smoother paths, as indicated by lower heading rates and curvature.

B. Comparison with Existing Work

We benchmark our valet parking framework—combining a spot occupancy estimator and strategy planner—against three existing approaches: *InfPath - Traversal* [19], *InfPath - MCBFT* [19], and *Rule-OBCA* [16]. *InfPath* estimates spot occupancies using a vanilla Bayes filter that ignores dynamic agents and employs an observation model independent of distance from ego vehicle, and presents two planning strategies: *Traversal* (Section IV-B of [19]) and Monte Carlo Bayes Filter Tree (MCBFT, Section IV-C of [19]). Although [19] focuses on exploration without dynamic obstacle avoidance, we enforce safety thresholds to discard unsafe paths and select the closest vacant spot according to (16).

- **InfPath - Traversal** samples feasible trajectories over a planning horizon of $T = 5$ s, and selects trajectories that either parks at the closest spot or explores using information gain that is computed using the occupancy estimation method presented in Section III of [19].
- **InfPath - MCBFT** builds a policy tree from spot occupancy probabilities and selects trajectories using the Upper Confidence Bound.
- **Rule-OBCA** selects the nearest available spot within the FoV and executes rule-based dynamic collision avoidance using Model Predictive Control with smooth nonlinear constraints [6].

Quantitatively, rows 6–9 in Table III show that our approach outperforms existing methods [16], [19], reducing path length and parking time by at least 25% while achieving lower smoothness costs and maintaining safe clearance from obsta-

TABLE III: Planning metrics for 50 randomized experiments across different methods: ablations on spot occupancy estimator and strategy planner, existing work and **our approach**. The numbers in green denote the best value for each metric.

No.	Method	Runtime ↓ (s)	Path length ↓ (m)	Parking time ↓ (s)	Average distance to closest dynamic obstacle ↑ (m)	Minimum distance to closest static obstacle ↑ (m)	Smoothness cost ↓	Average heading rate ↓ (deg/s)	Average curvature ↓ (m^{-1})
1	Greedy	0.041 ± 0.002	34.603 ± 3.113	23.702±5.211	3.335±0.103	0.315 ± 0.041	4.655 ± 2.83	12.995 ± 2.341	0.061 ± 0.02
2	Identical prediction	0.051 ± 0.002	32.589 ± 2.773	22.575±4.097	3.687±0.074	0.278 ± 0.092	4.302 ± 1.12	12.16 ± 1.154	0.056 ± 0.002
3	Position only	0.049 ± 0.008	33.735 ± 5.298	21.539±1.359	8.331±0.949	0.314 ± 0.131	5.045 ± 2.45	14.6±2.881	0.058 ± 0.012
4	No wait time	0.065 ± 0.002	45.735 ± 5.328	28.524±2.533	9.228 ± 1.25	0.264 ± 0.066	32.668 ± 4.987	18.989±3.1	0.058 ± 0.015
5	Lawn-mower	0.127 ± 0.03	41.858 ± 3.37	32.532 ± 3.58	3.273±0.119	0.273 ± 0.013	12.536 ± 3.1	17.032 ± 1.725	0.059 ± 0.008
6	InfPath - Traversal	0.059 ± 0.001	42.364 ± 3.15	26.725 ± 5.71	5.232±0.579	0.261 ± 0.078	13.583 ± 1.541	16.86±2.97	0.061 ± 0.034
7	InfPath - MCBFT	0.281 ± 0.012	35.742 ± 2.05	23.35 ± 2.13	6.634 ± 1.02	0.276 ± 0.044	14.543 ± 2.146	14.655 ± 2.51	0.077 ± 0.009
8	Rule-OBCA	0.416 ± 0.055	32.94 ± 2.11	25.195 ± 2.54	9.376±0.974	0.335 ± 0.095	20.54 ± 1.981	15.375 ± 1.632	0.062 ± 0.04
9	**Ours**	0.045 ± 0.005	24.001 ± 2.992	17.179±4.333	8.564 ± 1.03	0.271 ± 0.052	3.513 ± 0.95	11.119±2.2	0.024 ± 0.002

cle. The runtime presented in Table III is for our combined spot occupancy estimator and strategy planner.

VI. Conclusion and Future Work

We present a trajectory planning framework for autonomous valet parking with limited FoV that integrates a novel *spot occupancy estimator* and a *strategy planner* in uncertain, dynamic environments. We model initially vacant and occupied spots separately to estimate future occupancy probabilities, while also incorporating the motion of dynamic agents. The estimator is integrated with our strategy planner that balances information gain with goal-directed planning, and employs wait-and-go behaviors for promising spots. Simulation results show that our method outperforms existing approaches in efficiency, safety, and smoothness.

The following limitations in our work suggest directions for future research. First, the Hybrid $A^\star$ planner limits steering and velocity samples. Thus, exploring sampling-based methods like MPPI [26] could enable richer control strategies. Second, incorporating intent inference (e.g., parking, yielding, exiting) for other vehicles and evaluating the performance across a wide range of learned trajectory prediction models could enhance interaction-aware planning, thereby improving realism and safety. Third, occlusion-aware sensing models could provide more realistic occupancy estimates. Additionally, developing adaptive waiting behaviors based on lot density may improve efficiency. Finally, implementing our approach in high-fidelity simulators with interactive agents and real-world environments is our future goal.

References

[1] Automotive Fleet Staff. 48% of drivers find parking to be stressful. *Automotive Fleet*, 2022.

[2] Zuzanna Chmielewska. Parking is the most stressful part of driving – say nearly half of drivers. *appway*, 2022.

[3] Tesla Model 3 Owner's Manual, 2024. [Online].

[4] Kathryn Tomzcak, Adam Pelter, Corey Gutierrez, Thomas Stretch, Daniel Hilf, Bianca Donadio, Nathan L. Tenhundfeld, Ewart J. de Visser, and Chad C. Tossell. Let tesla park your tesla: Driver trust in a semi-automated car. In *2019 Systems and Information Engineering Design Symposium (SIEDS)*, pages 1–6, 2019.

[5] International Organization for Standardization. Intelligent transport systems — Automated valet parking systems (AVPS) — Part 1: System framework, requirements for automated driving and for communications interface, 2023.

[6] Xiaojing Zhang, Alexander Liniger, and Francesco Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29(3):972–983, 2020.

[7] Zhouheng Li, Lei Xie, Cheng Hu, and Hongye Su. A rapid iterative trajectory planning method for automated parking through differential flatness. *Robotics and Autonomous Systems*, 182:104816, 2024.

[8] Xu Shen, Edward L Zhu, Yvonne R Stürz, and Francesco Borrelli. Collision avoidance in tightly-constrained environments without coordination: a hierarchical control approach. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2674–2680. IEEE, 2021.

[9] Xuemin Chi, Zhitao Liu, Jihao Huang, Feng Hong, and Hongye Su. Optimization-based motion planning for autonomous parking considering dynamic obstacle: A hierarchical framework. In *2022 34th Chinese Control and Decision Conference (CCDC)*, pages 6229–6234. IEEE, 2022.

[10] Bai Li, Kexin Wang, and Zhijiang Shao. Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach. *IEEE Transactions on Intelligent Transportation Systems*, 17(11):3263–3274, 2016.

[11] Bai Li and Zhijiang Shao. A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles. *Knowledge-Based Systems*, 86:11–20, 2015.

[12] Jong Min Kim, Kyung Il Lim, and Jung Ha Kim. Auto parking path planning system using modified reeds-shepp curve algorithm. In *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 311–315. IEEE, 2014.

[13] Dongchan Kim and Kunsoo Huh. Neural motion planning for autonomous parking. *International Journal of Control, Automation and Systems*, 21(4):1309–1318, 2023.

[14] Runqi Chai, Derong Liu, Tianhao Liu, Antonios Tsourdos, Yuanqing Xia, and Senchun Chai. Deep learning-based trajectory planning and control for autonomous ground vehicle parking maneuver. *IEEE Transactions on Automation Science and Engineering*, 20(3):1633–1647, 2022.

[15] Yutong Li, Nan Li, H Eric Tseng, Ilya Kolmanovsky, Anouck Girard, and Dimitar Filev. A game theoretic approach for parking spot search with limited parking lot information. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.

[16] Xu Shen, Yongkeun Choi, Alex Wong, Francesco Borrelli, Scott Moura, and Soomin Woo. Parking of connected automated vehicles: Vehicle control, parking assignment, and multi-agent simulation. *arXiv preprint arXiv:2402.14183*, 2024.

[17] Xinyuan Zhang, Cong Zhao, Feixiong Liao, Xinghua Li, and Yuchuan Du. Online parking assignment in an environment of partially connected vehicles: A multi-agent deep reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 138:103624, 2022.

[18] Muhammad Khalid, Liang Wang, Kezhi Wang, Nauman Aslam, Cunhua Pan, and Yue Cao. Deep reinforcement learning-based long-range autonomous valet parking for smart cities. *Sustainable Cities and Society*, 89:104311, 2023.

[19] Yunze Hu, Jiaao Chen, Kangjie Zhou, Han Gao, Yutong Li, and Chang Liu. Informative path planning of autonomous vehicle for parking occupancy estimation. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3304–3310. IEEE, 2023.

[20] Jean-Luc Lupien, Abdullah Alhadlaq, Yuhan Tang, Jiayu Joyce Chen, and Yutan Long. Entropy-based dynamic programming for efficient vehicle parking. *arXiv preprint arXiv:2411.17014*, 2024.

[21] Xu Shen, Xiaojing Zhang, and Francesco Borrelli. Autonomous parking of vehicle fleet in tight environments. In *2020 American Control Conference (ACC)*, pages 3035–3040. IEEE, 2020.

[22] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[23] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[24] Mahalanobis Prasanta Chandra et al. On the generalised distance in statistics. In *Proceedings of the National Institute of Sciences of India*, volume 2, pages 49–55, 1936.

[25] Jun Xiao, Yingyan Lou, and Joshua Frisby. How likely am i to find parking?–a practical model-based framework for predicting parking availability. *Transportation Research Part B: Methodological*, 112:19–39, 2018.

[26] Alberto Bertipaglia, Dariu M Gavrila, and Barys Shyrokau. Multimodal model predictive path integral control for collision avoidance. *arXiv preprint arXiv:2508.21364*, 2025.

[27] Wenda Xu, Qian Wang, and John M Dolan. Autonomous vehicle motion planning via recurrent spline optimization. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7730–7736. IEEE, 2021.

[28] Sangjae Bae, David Isele, Alireza Nakhaei, Peng Xu, Alexandre Miranda Añon, Chiho Choi, Kikuo Fujimura, and Scott Moura. Lane-change in dense traffic with model predictive control and neural networks. *IEEE Transactions on Control Systems Technology*, 31(2):646–659, 2022.

[29] Farhad Nawaz, Minjun Sung, Darshan Gadginmath, Jovin D'sa, Sangjae Bae, David Isele, Nadia Figueroa, Nikolai Matni, and Faizan M Tariq. Graph-based path planning with dynamic obstacle avoidance for autonomous parking. *arXiv preprint arXiv:2504.12616*, 2025.

[30] 2024 Honda Accord Specifications and Features , 2023. [Online].

[31] La Puente, CA, Municipal Code §10.30.070 "Parking Space and Drive Aisle Dimensions", 2024. [Online].

[32] Howie Choset and Philippe Pignon. Coverage path planning: The boustrophedon cellular decomposition. In *Field and service robotics*, pages 203–209. Springer, 1998.