applied sciences

MDPI

Article

GI-SAC: Geometric-Prior-Guided and Intent-Driven Soft Actor-Critic for Autonomous Parking

Jinhui Gao [1,2,3,4], Yijie Zhang [1,2,3,4], Mingxin Fan [1,2,3,4], Xukai Zhai [1,2,3,4], Hao Yuan [1,2,3,4], Longfei Jiang [1,2,3,4], Ying Xie [1,2,3,4] and Zhishuai Yin [1,2,3,4,*]

[1] Department of Automotive Engineering, Wuhan University of Technology, Wuhan 430070, China; 362329@whut.edu.cn (J.G.); 313758@whut.edu.cn (Y.Z.); mingxin.fan@whut.edu.cn (M.F.); zhaixukai@whut.edu.cn (X.Z.); 314251@whut.edu.cn (H.Y.); 362352@whut.edu.cn (L.J.); 362446@whut.edu.cn (Y.X.)
[2] Hubei Key Laboratory of Advanced Technology for Automotive Components, Wuhan University of Technology, Wuhan 430070, China
[3] Hubei Collaborative Innovation Center for Automotive Components Technology, Wuhan University of Technology, Wuhan 430070, China
[4] Hubei Research Center for New Energy & Intelligent Connected Vehicles, Wuhan University of Technology, Wuhan 430070, China
* Correspondence: zyin@whut.edu.cn

Featured Application

The proposed framework is designed for closed-loop autonomous parking, with potential applications in real-world intelligent vehicles to enhance parking success, decision efficiency, and steering continuity in complex and constrained environments.

Abstract

Autonomous parking is an essential component of safe and efficient urban autonomous driving. However, complex constraints in unstructured and confined environments pose significant challenges to parking decision-making. Rule-based planners explicitly account for geometric and kinematic constraints, but their computational demands can increase with the complexity of parking environments. Reinforcement learning offers flexible closed-loop decision-making, yet effective exploration remains challenging in confined spaces. To leverage the strengths of both approaches, this paper proposes Geometric-Prior-Guided and Intent-Driven Soft Actor-Critic (GI-SAC), a closed-loop autonomous parking method integrating geometric-prior guidance with intent-driven representation. First, a coarse Hybrid A* (HA*) trajectory is incorporated into the policy observation and used to construct an auxiliary reward based on the vehicle's distance to the reference path, with a gradually decaying reward weight regulating the guidance strength. Second, the target parking pose and coarse trajectory are jointly encoded as parking intent, which guides the aggregation of relevant local environmental features through cross-attention. Finally, a spatiotemporal action mask combines spatial feasibility weights with temporal weights based on candidate-action changes relative to the previously executed command, promoting steering continuity through joint weighting and resampling. Evaluations on the Tactics2D simulation platform show that GI-SAC achieves an average success rate of 99.05% across four test scenarios, with success rates of 98.30% and 98.70% in the Difficult and Dragon Lake Parking (DLP) scenarios, respectively, and reduces failures caused by boundary violations and timeouts. Ablation studies and action-variation analyses further demonstrate the contributions of the proposed modules to parking performance and steering-command continuity.

Check for updates

Academic Editor: Krzysztof Zboiński

Received: 23 September 2026
Revised: 31 October 2026
Accepted: 2 November 2026
Published: 6 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Keywords: autonomous parking; reinforcement learning; geometric prior; motion planning

1. Introduction

With the development of intelligent vehicle technologies, autonomous parking has become important for reducing driver workload and improving vehicle intelligence [1,2]. Unlike driving on structured roads, autonomous parking is often performed in space-constrained environments and imposes stricter requirements on target pose accuracy and motion feasibility. The vehicle must generate collision-free trajectories within limited space while satisfying nonholonomic kinematic constraints and maintaining sufficient continuity and trackability [3]. Therefore, motion planning in complex parking scenarios can be regarded as a closed-loop sequential decision-making problem under constraints, where executable motion commands are continuously generated from the vehicle state and environmental feedback [4]. Improving exploration efficiency and steering continuity in complex parking tasks while ensuring safety and motion feasibility remains an important research problem in autonomous driving. Traditional autonomous parking motion planning methods mainly include geometric curve-based, heuristic search-based, and numerical optimization-based methods [5–7]. Geometric methods are computationally efficient and interpretable, but their performance is limited by predefined curve structures, making them less flexible for complex maneuvers in constrained spaces. Heuristic search-based methods incorporate vehicle nonholonomic constraints through motion primitives and can generate kinematically feasible paths. However, complicated obstacle configurations may significantly increase the search space and reduce computational efficiency. Numerical optimization-based methods can improve trajectory smoothness and constraint satisfaction, but they usually depend on high-quality initial solutions and may suffer from unstable convergence under complex nonconvex constraints. Overall, traditional methods provide reliable and interpretable solutions but still face challenges in achieving both high success rates and computational efficiency for complex parking scenarios.

To reduce the dependence of traditional methods on manually designed rules and online search, learning-based methods have been introduced into autonomous parking planning and control [8–10]. Imitation learning can acquire parking behavior priors from expert trajectories and reduce rule-design effort, but its performance is constrained by the expert data distribution and may degrade under distribution shifts in unseen scenarios. Reinforcement learning (RL) directly learns closed-loop decision-making policies through environment interaction, providing better adaptability to complex parking conditions. However, limited parking space and vehicle motion constraints pose challenges to exploration efficiency and trajectory feasibility in reinforcement learning. To alleviate these challenges, existing studies have integrated reinforcement learning with trajectory optimization, expert demonstrations, and geometric planners in different ways. Specifically, RL-generated trajectories have been used for subsequent trajectory optimization [11], and demonstration-derived rewards have been employed for planner-parameter adaptation [12]. RL and Reeds–Shepp (RS) planning have also been combined through conditional policy switching [13], with further integration of LiDAR occupancy-grid representations to incorporate obstacle and free-space information into parking decisions [14]. These studies demonstrate the benefits of integrating reinforcement learning with conventional planning methods and prior knowledge for planning in constrained environments. However, how geometric trajectory information can assist exploration while helping the policy identify local spatial constraints relevant to the parking target warrants further investigation. In addition, control continuity is an important consideration in learning-based autonomous parking,

particularly in confined spaces. Existing studies have addressed control smoothness by incorporating steering-magnitude and steering-increment penalties into the reward function of a twin delayed deep deterministic policy gradient (TD3)-based parking controller [15], or by applying moving-average filters to SAC-generated control commands before execution [16]. These approaches improve control smoothness through reward shaping and output filtering, respectively. Beyond these approaches, another relevant question is how to incorporate steering-continuity preferences into action selection while accounting for spatial feasibility.

Motivated by these considerations, this paper proposes Geometric-Prior-Guided and Intent-Driven Soft Actor-Critic (GI-SAC), a closed-loop autonomous parking decision-making framework integrating geometric-prior guidance with intent-driven deep reinforcement learning. The framework incorporates a coarse global reference trajectory generated by a Hybrid A* (HA*) planner into Soft Actor-Critic (SAC) policy learning, providing global directional guidance alongside local environmental observations. Specifically, a dynamic guidance reward derived from the HA* trajectory prior provides auxiliary exploration feedback based on the relative relationship between the vehicle state and reference trajectory. A dynamic decay mechanism gradually reduces the guidance reward weight to facilitate early-stage exploration while reducing the policy's long-term reliance on the geometric reference path. Furthermore, the target pose and coarse HA* trajectory are jointly encoded as parking intent and fused with local environmental features through cross-attention, enhancing the representation of target-relevant environmental information. In the Critic network, continuous action variables, including vehicle speed and front-wheel steering angle, are processed separately according to their physical characteristics during action-value estimation. A spatiotemporal action masking mechanism further combines spatial feasibility weights with soft temporal weights based on candidate-action changes relative to the previously executed command. The combined weights guide candidate-action resampling by jointly considering spatial feasibility and temporal continuity preferences, thereby encouraging steering-command continuity during parking.

Overall, the main contributions of this work include the following:

- We propose a geometric-prior guidance mechanism that uses a coarse HA* trajectory to provide a global reference for policy exploration. An auxiliary reward with a dynamically decaying weight regulates the guidance strength, facilitating the acquisition of useful parking experience.
- We design an intent-driven cross-attention representation that jointly defines parking intent through the target parking pose and coarse HA* trajectory. This intent directs attention toward goal-relevant environmental features, connecting global geometric information with local spatial constraints.
- We propose a spatiotemporal action masking mechanism that incorporates spatial feasibility and temporal continuity preferences into action selection, promoting steering continuity in constrained parking environments.

2. Related Work

2.1. Rule-Based Autonomous Parking Algorithms

Rule-based motion planning methods for autonomous parking rely on explicit geometric structures and manually designed priors, including geometric curve-based, heuristic search-based, and trajectory optimization-based methods. Geometric methods typically connect the initial and target vehicle poses using predefined curves under kinematic constraints. The Dubins curve [17] generates the shortest forward-only path for a curvature-constrained vehicle, whereas the Reeds–Shepp curve [18] further considers reverse motion and is therefore more suitable for low-speed parking. Arcline combinations [19] and B-spline

curves [20] improve the representation capability for diverse parking maneuvers. However, predefined curve families limit adaptability to complex pose adjustments in constrained spaces. Heuristic search-based methods generate paths satisfying vehicle nonholonomic constraints by searching over discrete motion primitives. HA* [21] extends conventional A* search to continuous pose space and incorporates a vehicle kinematic model to generate feasible paths. Its performance in narrow parking scenarios has been improved through enhanced heuristic functions [22], multi-stage search strategies [23], and path smoothing and optimization [24]. Although geometric and heuristic search-based methods are reliable in simple scenarios, they still face challenges in balancing planning efficiency and success rate in complex and narrow environments. Trajectory optimization methods formulate autonomous parking as a constrained optimization problem to ensure motion feasibility and control executability [25]. For example, existing work [26] constructs safety corridors to simplify collision avoidance constraints and improve computational efficiency. Trajectory optimization is also combined with graph search or model predictive control [27,28], forming hierarchical frameworks that integrate front-end search with back-end optimization. Related studies transform collision avoidance conditions into optimization constraints or combine an initial HA* path with multi-stage dynamic optimization to improve trajectory safety and smoothness [29]. However, these methods are highly dependent on constraint design and initial solution quality and may still suffer from unstable optimization and high computational cost in narrow spaces with complex obstacles.

2.2. Learning-Based Autonomous Parking Algorithms

Learning-based methods have been applied to parking tasks such as parking-space allocation and vehicle motion planning. Nakanishi et al. [30] use Q-learning to allocate parking spaces based on occupancy conditions and user information, balancing driving and walking distances. Sakib et al. [31] incorporate vehicle types and passenger numbers into an adaptive allocation objective to reduce emissions while maintaining user comfort. For vehicle motion planning, learning-based methods use neural networks to learn the mapping from environmental states to parking behaviors, reducing dependence on predefined rules and numerical optimization [32]. Among these methods, imitation learning (IL) trains policies from expert demonstrations to learn expert behavior patterns [33]. Some studies construct autonomous parking training and testing environments in CARLA and use end-to-end neural networks to map surround-view images and vehicle states to parking control commands, demonstrating the feasibility of imitation learning for simulated parking control [34]. Other studies develop Transformer-based end-to-end parking models that learn relationships between visual features and parking trajectories under expert trajectory supervision, improving the accuracy of vision-based trajectory generation [35]. In addition, some studies [36] incorporate expert demonstrations into reinforcement learning, using imitation learning for policy pretraining and reward adjustment to improve training efficiency. However, imitation learning relies heavily on expert data quality and scenario coverage. When the test distribution differs from the training distribution, learned policies may suffer from distribution shift and error accumulation. In contrast, reinforcement learning formulates parking as a Markov decision process, where the agent learns through interaction with the environment and optimizes vehicle behavior by maximizing expected cumulative rewards [37,38]. Du et al. [39] demonstrate the feasibility of RL-based parking trajectory generation in simulation. Tang et al. [40] introduce a multi-stage SAC-based training framework for non-ideal parking scenarios, including deviations in adjacent parked vehicle poses, to improve safety, accuracy, and comfort. Al-Mousa et al. [41] investigate an RL-based reverse parking system, demonstrating its applicability to reverse parking

control. However, learning feasible maneuvers through environmental interaction remains challenging in long-horizon parking tasks with tight spatial and vehicle-motion constraints.

To alleviate these difficulties, existing studies combine reinforcement learning with planning methods or expert demonstrations. Yuan et al. [11] improve the feasibility of RL-generated coarse trajectories through optimization subject to vehicle kinematic and collision-avoidance constraints. Lu et al. [12] use a scene-conditioned discriminator to provide imitation rewards for RL-based local-planner parameter adjustment. Jiang et al. [13] combine RL and RS policies through conditional switching, allowing the geometric planner to participate in exploration and decision-making. Wang et al. [14] further incorporate LiDAR occupancy grids to provide obstacle and free-space information for hybrid planning. Nevertheless, accurately representing the relationship between overall maneuver requirements and local spatial constraints remains challenging in complex parking environments.

3. Methods

3.1. Markov Decision Process Formulation for Autonomous Parking

The motion planning problem for autonomous parking can be formulated as a policy learning problem within a Markov decision process (MDP). The MDP is represented by the five-tuple (S, A, P, r, γ), where S denotes the state space, A denotes the action space, P denotes the state transition model, r denotes the reward function, and $\gamma \in [0, 1]$ denotes the discount factor.

At time step t, the state $s_t \in S$ includes the ego-vehicle pose $p_t = (x_t, y_t, \theta_t)$, environmental obstacle information, target parking space information, and global reference trajectory features generated by the front-end HA* planner. The HA* reference trajectory is incorporated as prior information in the state representation, and its specific role is described in the subsequent sections. Because autonomous parking is a low-speed continuous control task, vehicle motion is primarily constrained by nonholonomic kinematics, the minimum turning radius, and the front-wheel steering limits. Accordingly, the longitudinal velocity v and front-wheel steering angle δ are selected as continuous action variables. The action space is defined as follows: $\mathcal{A} = \{a_t = (v_t, \delta_t) \mid v_t \in [v_{\min}, v_{\max}], \delta_t \in [\delta_{\min}, \delta_{\max}]\}$. External stochastic disturbances are not considered in the parking task studied in this paper. The vehicle state is updated recursively at a fixed time interval Δt using the kinematic bicycle model. At each interaction step, the environment provides an immediate base reward $r = r_{\text{base}}(s_t, a_t)$ according to the current vehicle state and action.

The base reward is designed according to the task objectives and motion requirements of autonomous parking, comprising terminal rewards for parking success and failure events, as well as shaping rewards for target distance reduction, heading alignment, and parking efficiency. The base reward $r_{\text{base}}(t)$ is defined as:

$$r_{\text{base}}(t) = w_s r_{\text{succ}} + w_f r_{\text{fail}} + w_d r_{\text{dist}}(t) + w_\theta r_{\text{heading}}(t) + w_t r_{\text{time}}(t), \quad (1)$$

where $r_{\text{succ}} = 5$ is the positive reward assigned when the vehicle successfully enters the target parking space. The term $r_{\text{fail}} = -5$ denotes the penalty imposed when the vehicle collides with an obstacle, leaves the valid region, or fails to complete the parking task within the maximum number of interaction steps. The term $r_{\text{dist}}(t)$ denotes the distance reward, which encourages the vehicle to move toward the target position. The term $r_{\text{heading}}(t)$ denotes the heading error reward, which encourages the vehicle orientation to align with the target orientation. The term $r_{\text{time}}(t)$ denotes a per-step time penalty that encourages the vehicle to complete the parking task using fewer interaction steps. The coefficients w_s, w_f, w_d, w_θ, and w_t specify the weights of the corresponding reward terms, which are set to $w_s = w_f = w_d = 1$, $w_\theta = 0.5$, and $w_t = 0.1$, respectively.

The target distance reward is defined as follows:

$$r_{\text{dist}}(t) = \frac{d_{t-1} - d_t}{\max(d_0, d_{\min})}, \tag{2}$$

where d_t denotes the distance between the current vehicle position and the target position, d_{t-1} denotes the corresponding distance at the previous time step, d_0 denotes the initial distance to the target position, and $d_{\min}$ is a positive constant that prevents the denominator from becoming excessively small.

The heading error reward is defined as follows:

$$r_{\text{heading}}(t) = \frac{|\Delta\theta_{t-1}| - |\Delta\theta_t|}{\pi}, \tag{3}$$

where $\Delta\theta_t$ denotes the heading error between the current vehicle orientation and the target orientation, and $\Delta\theta_{t-1}$ denotes the heading error at the previous time step.

The time penalty is defined as follows:

$$r_{\text{time}}(t) = -\frac{t}{T_{\max}}, \tag{4}$$

where t denotes the current interaction step and $T_{\max}$ denotes the maximum number of interaction steps in an episode.

The objective of reinforcement learning is to learn an optimal policy $\pi^*(a_t \mid s_t)$ that enables the vehicle to start from the initial pose p_0 and progressively generate a feasible parking trajectory $\mathcal{P} = \{p_0, p_1, \ldots, p_N\}$ through closed-loop interaction while maximizing the discounted cumulative reward R_t from time step t:

$$R_t = \sum_{i=t}^{N} \gamma^{i-t} r(s_i, a_i). \tag{5}$$

To evaluate and optimize the policy, the action-value function $Q^\pi(s_t, a_t)$ is introduced. It represents the expected cumulative return obtained by taking action a_t in state s_t and subsequently following policy π:

$$Q^\pi(s_t, a_t) = \mathbb{E}_{s_{i>t} \sim P,\, a_{i>t} \sim \pi}[R_t \mid s_t, a_t], \tag{6}$$

the state-value function $V^\pi(s_t)$ represents the expected return obtained by selecting actions according to the current policy in state s_t:

$$V^\pi(s_t) = \mathbb{E}_{a_t \sim \pi(\cdot|s_t)}[Q^\pi(s_t, a_t)], \tag{7}$$

according to the Bellman equation, the action-value function can be written recursively in terms of the immediate reward and the value of the next state:

$$Q^\pi(s_t, a_t) = \mathbb{E}_{s_{t+1} \sim P(\cdot|s_t, a_t)}[r(s_t, a_t) + \gamma V^\pi(s_{t+1})]. \tag{8}$$

Autonomous parking typically involves high-dimensional state representations, continuous action spaces, and nonlinear vehicle kinematic constraints. Consequently, conventional value iteration methods cannot directly produce stable and effective control policies for such tasks. To enhance exploration and training stability in continuous control, Soft Actor-Critic (SAC) is adopted as the base reinforcement learning framework. The optimization objective $J_\pi(\theta)$ of the policy network parameterized by θ is formulated as follows:

$$J_\pi(\theta) = \mathbb{E}_{s_t \sim \mathcal{D}}\Big[\mathbb{E}_{a_t \sim \pi_\theta(\cdot|s_t)}\big[\alpha \log \pi_\theta(a_t \mid s_t) - Q_\phi(s_t, a_t)\big]\Big], \tag{9}$$

where $\alpha \log \pi_\theta(a_t \mid s_t)$ denotes the policy entropy regularization term, α is the temperature coefficient that balances policy exploration and return maximization, and $\mathcal{D}$ denotes the

experience replay buffer. By minimizing this objective, SAC updates the policy to favor actions with high estimated values while preserving sufficient entropy. This formulation promotes diversity in the policy distribution and enables the agent to explore different parking adjustment strategies in constrained spaces.

Based on the MDP formulation described above, SAC is employed as the base framework for policy optimization. The subsequent sections introduce HA* coarse-trajectory guidance, a feature fusion network, and a spatiotemporal action masking mechanism to further improve exploration efficiency and steering continuity in narrow parking scenarios. The overall architecture of the proposed method is illustrated in Figure 1.

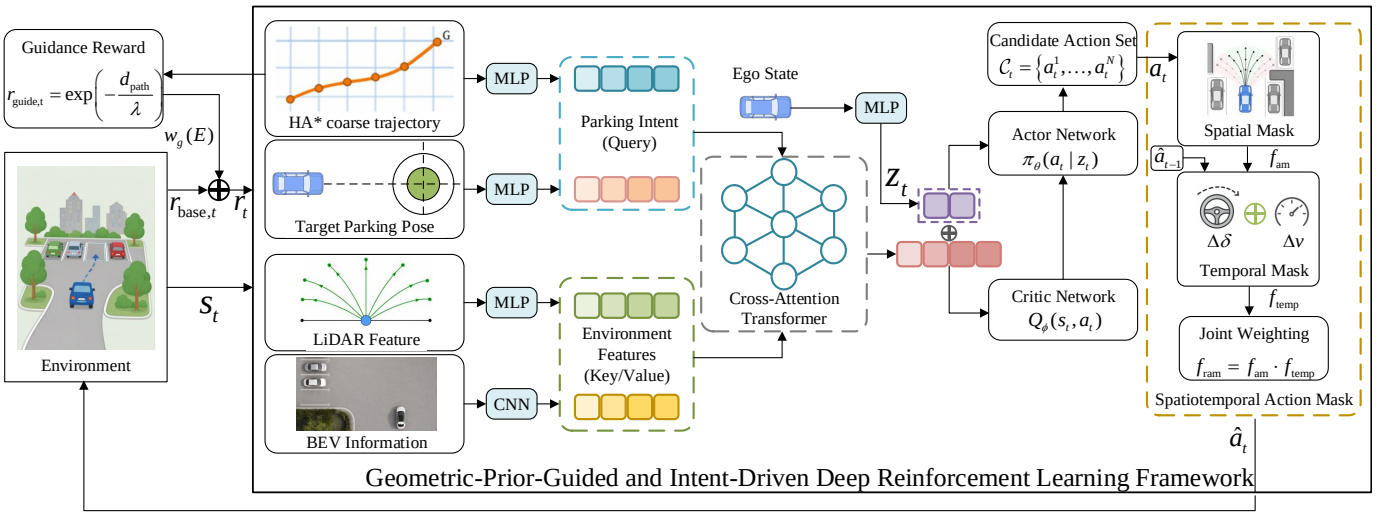


Figure 1. Overall framework of the proposed method. Arrows indicate the flow of information, colored blocks represent different feature types, and dashed boxes highlight the main functional modules.

3.2. Dynamic Guidance Reward Based on the HA Coarse-Trajectory Prior*

To alleviate inefficient exploration and sparse reward feedback in standard SAC for long-horizon parking tasks, this paper introduces an HA* planner to generate a coarse global reference trajectory and incorporates the trajectory into the reinforcement learning framework as a geometric prior. Rather than directly tracking the HA* trajectory, the proposed method uses it only to provide global directional guidance and auxiliary reward feedback. This design facilitates the acquisition of effective parking experience during the early stage of training. The influence of the geometric prior is then gradually reduced during the later stage, thereby encouraging the policy to develop autonomous decision-making capability.

At the beginning of each training episode, a two-dimensional grid map is constructed by inflating the static obstacles. The initial vehicle pose p_0 and the center pose of the target parking space p_G are then specified as the start and goal states, respectively. The HA* algorithm searches for a coarse reference path that satisfies the collision avoidance constraints defined by the grid map. Each search node is represented by a three-dimensional state (x, y, θ) that contains the vehicle position and heading angle. Basic motion primitives, including forward and reverse motions with left and right steering, are employed to generate a directionally continuous reference path at a low computational cost. To incorporate this geometric prior into the policy network, the path is discretized into a coarse trajectory sequence:

$$\mathcal{T}_{HA^*} = \{w_1, w_2, \ldots, w_i, \ldots, w_K\} \in \mathbb{R}^{K \times d_w}, \tag{10}$$

where K denotes the number of sampled points, d_w denotes the feature dimension of each trajectory point, and $w_i = (x_i, y_i, \theta_i)$ denotes the position coordinates of the ith sampled point and its local tangent heading estimated from adjacent sampled points.

Since HA* may fail to generate a feasible path in complex parking scenarios, we introduce a trajectory-prior degradation mechanism. When HA* planning succeeds, the resulting trajectory sequence is used to construct the prior observation. When planning fails, the trajectory prior is replaced with a zero vector of fixed dimensions, and the guidance reward is set to zero in the subsequent reward calculation. In this case, the policy operates in a closed loop without trajectory guidance.

The extracted coarse trajectory vector $\widetilde{\mathcal{J}}_{HA^*}$ is subsequently incorporated into the observation space as an independent spatial prior. The complete state at time step t is expressed as follows:

$$s_t = \left[o_t^{ego}, o_t^{env}, \widetilde{\mathcal{J}}_{HA^*}\right], \tag{11}$$

where o_t^{ego} denotes the kinematic features of the ego vehicle and o_t^{env} denotes the perceptual features of the local environment.

To further improve exploration efficiency, an auxiliary guidance reward is constructed from the lateral deviation between the current vehicle position and the HA* coarse trajectory. At time step t, the minimum orthogonal projection distance d_t^{path} from the vehicle's rear-axle center q_t to all trajectory segments is calculated as follows:

$$d_t^{\text{path}} = \min_{i \in [1, K-1]} \text{dist}_{\perp}(q_t, \overline{w_i w_{i+1}}), \tag{12}$$

where $\text{dist}_{\perp}(\cdot)$ denotes the minimum Euclidean distance from a point to a line segment.

During the early stage of exploration, the vehicle position may deviate substantially from the reference trajectory. To prevent a linear penalty from producing excessively large reward variations when the vehicle is far from the trajectory, a bounded exponential guidance reward is defined as follows:

$$r_{\text{guide}}(t) = \exp\left(-\frac{|d_t^{\text{path}}|}{\lambda}\right), \tag{13}$$

where λ is a smoothing parameter and $r_{\text{guide}}(t) \in (0, 1]$.

Maintaining a strong HA* guidance signal throughout training may cause the policy to rely excessively on the coarse trajectory. To balance geometric-prior guidance and autonomous exploration, a dynamic prior modulation mechanism based on cosine annealing is further introduced. Let E denote the current training episode and E_{trans} denote the duration of the decay transition. The guidance reward weight $w_g(E)$ is defined as follows:

$$w_g(E) = \begin{cases} w_{\text{end}} + \dfrac{1}{2}(w_{\text{start}} - w_{\text{end}}) \times \left[1 + \cos\left(\pi \dfrac{E}{E_{\text{trans}}}\right)\right], & E \le E_{\text{trans}}, \\ w_{\text{end}}, & E > E_{\text{trans}}, \end{cases} \tag{14}$$

where w_{start} and w_{end} denote the guidance weights used during the early and late stages of training, respectively. At the beginning of training, a larger $w_g(E)$ strengthens the guidance provided by the HA* prior. As training progresses, the weight gradually decreases, allowing the policy to rely more on local environmental observations and vehicle motion states when making parking control decisions.

The total reward is therefore written as follows:

$$r_t = r_{\text{base}}(s_t, a_t) + w_g(E) r_{\text{guide}}(t). \tag{15}$$

3.3. Scale-Decoupled and Intent-Driven Observation Feature Representation

Complex parking decisions require the policy to extract intent-relevant information from multi-source observations. Simple feature concatenation cannot adequately capture the relationships among the target parking space, local environment, and trajectory prior. Moreover, SAC action variables differ from state features in physical meaning and numerical scale. To address these issues, this paper develops a state-action decoupled normalization scheme and an intent-driven cross-attention fusion network. The network first normalizes and encodes state features while excluding action variables from Critic state normalization. It then constructs a parking-intent representation from the target parking space and HA* coarse-trajectory prior. This representation serves as the cross-attention query to selectively aggregate local environmental features, producing an intent-enhanced state representation. The Actor generates continuous control actions from this representation, while the Critic combines it with unnormalized action variables to estimate the corresponding state-action value. The normalized state feature at time step t is defined as follows:

$$\hat{s}_t = \mathrm{Norm}(s_t), \tag{16}$$

where $\mathrm{Norm}(\cdot)$ denotes the state normalization operation.

During off-policy temporal-difference updates, the current states s_t and next states s_{t+1} within the same batch are jointly normalized to reduce the discrepancy between their batch statistics. The action variable a_t is excluded from this normalization process and is instead concatenated with the state representation z_t in the Critic network:

$$[\hat{s}_t, \hat{s}_{t+1}] = \mathrm{Norm}([s_t, s_{t+1}]). \tag{17}$$

To further establish the relationship between parking intent and environmental constraints, an intent-driven cross-attention fusion mechanism is introduced. The target pose p_{tgt} and the HA* coarse trajectory $\tilde{\mathcal{T}}_{HA^*}$ are first fed into an intent encoder to obtain the intent feature h_t^g. Meanwhile, the local environmental observation o_t^{env} is processed by an environmental encoder to extract a token sequence H_t^e composed of multiple types of environmental features:

$$h_t^g = f_g\left(p_{\mathrm{tgt}}, \tilde{\mathcal{T}}_{HA^*}\right), \tag{18}$$

$$H_t^e = f_e(o_t^{env}), \tag{19}$$

where $f_g(\cdot)$ denotes the encoder for the target pose and trajectory prior, and $f_e(\cdot)$ denotes the local environmental feature encoder. The environmental token sequence satisfies $H_t^e \in \mathbb{R}^{n_{\mathrm{env}} \times D}$.

The intent feature is then projected into a query vector, while the environmental features are projected into key and value vectors. The cross-attention feature F_{attn} is computed as

$$\mathbf{Q} = W_q h_t^g, \quad \mathbf{K} = W_k H_t^e, \quad \mathbf{V} = W_v H_t^e, \tag{20}$$

$$F_{\mathrm{attn}} = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}, \tag{21}$$

where W_q, W_k, and W_v are learnable linear projection matrices, and d_k denotes the dimension of the key vectors.

Finally, the cross-attention output is concatenated with the ego-vehicle motion state to construct the state representation used for policy optimization:

$$z_t = [F_{\mathrm{attn}}, o_t^{ego}]. \tag{22}$$

The Actor network generates a continuous action distribution based on z_t. The Critic network concatenates the physically independent action variables with the state representation to estimate the corresponding state-action value:

$$z_t^Q = [z_t, a_t], \tag{23}$$

$$Q_\phi(s_t, a_t) = f_Q(z_t^Q), \tag{24}$$

where $f_Q(\cdot)$ denotes the Critic value estimation network and ϕ denotes its parameters.

3.4. Spatiotemporal Action Masking Based on Relative Action Variations

To reduce excessive steering and abrupt control changes inconsistent with human driving behavior during autonomous parking exploration, a soft temporal action mask based on consecutive executed commands is introduced into the existing spatial action masking mechanism [13]. The basic spatial action mask suppresses candidate actions that may cause collisions or boundary violations by estimating the safe travel distance under different steering conditions, while the soft temporal action mask further improves action continuity by limiting velocity and steering variations between consecutive control cycles.

Let $\hat{a}_{t-1} = (\hat{v}_{t-1}, \hat{\delta}_{t-1})$ denote the control command actually applied during the previous control cycle, and $C_t = \{a_t^i\}_{i=1}^N$ denote the candidate action set, where $a_t^i = (v_t^i, \delta_t^i)$. All variations in the control commands are calculated in physical units after converting the normalized policy outputs. The relative variations between each candidate action and the previously executed action are defined as follows:

$$\Delta v_t^i = v_t^i - \hat{v}_{t-1}, \qquad \Delta \delta_t^i = \delta_t^i - \hat{\delta}_{t-1}. \tag{25}$$

Based on the relative action variations, a continuous temporal mask is further constructed to assign sampling weights to the candidate actions:

$$f_{\text{temp}}(a_t^i; \hat{a}_{t-1}) = \exp\left[-\lambda_v \left(\frac{\Delta v_t^i}{s_v} \right)^2 - \lambda_\delta \left(\frac{\Delta \delta_t^i}{s_\delta} \right)^2 \right], \tag{26}$$

where s_v and s_δ denote the characteristic scales of the velocity and steering variations, respectively, while λ_v and λ_δ control the corresponding weight-decay rates.

The temporal mask is then combined with the basic spatial action mask to obtain the spatiotemporal action mask:

$$f_{\text{ram}}(s_t, a_t^i, \hat{a}_{t-1}) = f_{\text{am}}(s_t, a_t^i) \cdot f_{\text{temp}}(a_t^i; \hat{a}_{t-1}), \tag{27}$$

where $f_{\text{am}}(s_t, a_t^i) \in [0, 1]$ denotes the basic spatial action mask, and $f_{\text{ram}}(s_t, a_t^i, \hat{a}_{t-1}) \in [0, 1]$ the joint sampling weight incorporating spatial feasibility and temporal smoothness.

The action selected through joint masking and resampling is executed in the environment. By adjusting the sampling weights of candidate actions, this mechanism encourages smooth changes between consecutive executed actions.

4. Results

4.1. Experimental Setup

4.1.1. Simulation Environment and Scenario Construction

Experiments are conducted in Tactics2D (version 0.1.9; Shanghai Jiao Tong University, Shanghai, China) [42], an open-source simulator for autonomous driving decision-making. The platform provides LiDAR and bird's-eye-view (BEV) observations, vehicle kinematic modeling, obstacle modeling, collision detection, and parking visualization. It supports both randomly generated parking tasks and DLP-based scenarios within a unified framework.

Parking-space dimensions, initial vehicle poses, and obstacle configurations can be adjusted for perpendicular and parallel parking tasks. These capabilities provide a closed-loop interaction environment for reinforcement learning-based low-speed parking, supporting large-scale policy training and evaluation. The experimental vehicle is represented by a kinematic bicycle model commonly used in low-speed parking scenarios. The vehicle state consists of the rear-axle center position, heading angle, velocity, and front-wheel steering angle. At each interaction step, the reinforcement learning policy generates a continuous action from the current observation, comprising the longitudinal velocity and front-wheel steering angle. The vehicle state is subsequently updated according to the kinematic model. An episode terminates when the vehicle successfully parks within the target parking space, collides with an obstacle, exits the drivable region, or reaches the maximum number of interaction steps.

To evaluate the adaptability of the algorithms under different parking conditions, randomized parking scenarios and dataset-based scenarios derived from real-world parking lot layouts are constructed. The randomized scenarios include perpendicular and parallel parking tasks. The dimensions of the target parking space, the positions of surrounding obstacles, and the initial vehicle pose are randomly sampled from predefined ranges. Specifically, initial vehicle positions are uniformly sampled, while initial headings and selected obstacle parameters are generated through Gaussian sampling followed by clipping. The scenario parameters are configured according to the autonomous parking test standard GB/T 47031-2026 [43] and the Code for Design of Parking Garage JGJ 100-2015 [44], with consideration of the vehicle dimensions and the scale of the simulation environment. The length and width of the target parking space are denoted by L_{park} and W_{park}, respectively, while the vehicle length and width are denoted by L and W. The distance between the surrounding obstacles and the target parking space is denoted by D_{obst}, and the distance from the initial vehicle position to the center of the target parking space is denoted by D_{init}. The parameters L_{park} and W_{park} define the spatial constraints of the target parking space, D_{obst} indicates the available maneuvering space around it, and D_{init} represents the distance required for the vehicle to move from the initial pose toward the target pose. The value of D_{init} primarily ranges from 5 to 20 m. Based on these parameters, the randomized parking scenarios are categorized into Normal, Complex, and Difficult levels. The corresponding scenario configurations are illustrated in Figure 2 and Table 1.

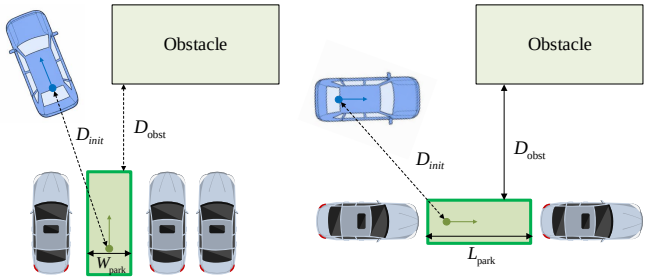


Figure 2. Illustration of parking scenarios.

Table 1. Parameter settings of parking scenarios.

Scenario	Parking Length (m)	Parking Width (m)	Clearance (m) Parallel/Perpendicular	No. of Obstacles
Normal	5.0–6.0	2.50–2.95	4.5/7.0	3
Complex	4.8–5.0	2.40–2.50	3.8/6.0	5
Difficult	4.5–4.8	2.30–2.40	3.5/5.5	8

The dataset-based scenarios derived from real-world parking lot layouts are constructed using the open-source preprocessed Dragon Lake Parking (DLP) dataset [13]. The dataset originates from real-world parking observations and is preprocessed by filtering out non-parking trajectories and dynamically interfering vehicles to obtain static parking scenarios. These scenarios reflect real-world parking-space layouts and parked-vehicle distributions and are reconstructed in Tactics2D for simulation-based training and evaluation. The original observations comprise approximately 3.5 h of drone-recorded video, covering about 400 parking spaces and 5188 vehicles. The preprocessed data used here contain 248 base parking cases. Starting poses are selected from recorded vehicle trajectories, with Gaussian perturbations applied to the initial position and heading. The initial and target headings are each reversed by π with a probability of 0.5. Training and evaluation use the same underlying parking layouts, with initial conditions randomized in each phase. During training, parking tasks with different types and difficulty levels are generated through dynamic random sampling rather than from a predefined fixed training set, thereby increasing scenario diversity. Each reinforcement learning algorithm is trained for 100,000 episodes. During testing, 2000 parking tasks are randomly generated for each scenario category. For synthetic scenarios, training and test tasks are independently generated using the same parameter ranges and sampling rules.

All experiments are conducted on a workstation equipped with two NVIDIA GeForce RTX 4090 GPUs (NVIDIA Corporation, Santa Clara, CA, USA; 24 GB VRAM each) and a 40-core, 80-thread CPU. Each training task is executed independently according to the availability of computational resources.

4.1.2. Comparison Methods

To evaluate the effectiveness of the proposed method, the Reeds–Shepp curve, HA*, Proximal Policy Optimization (PPO), and SAC are selected as comparison methods. The Reeds–Shepp curve generates paths consisting of forward and reverse motions while satisfying the minimum turning radius constraint. HA* incorporates continuous vehicle states and kinematic constraints into the discrete search process to generate kinematically feasible paths. PPO and SAC are adopted as representative on-policy and off-policy reinforcement learning baselines, respectively. All methods are evaluated under the same scenario configurations and vehicle kinematic constraints. PPO and SAC use standard continuous-action policies without action masking. The proposed method adopts the complete module configuration, while the contribution of each module is further examined through ablation studies. For the three learning-based methods, the Actor and Critic learning rates were set to 5×10^{-6} and 2.5×10^{-5}, respectively, with a gradient update batch size of 32. SAC and GI-SAC employed an entropy temperature learning rate of 5×10^{-6} and a replay buffer capacity of 10,240, whereas PPO used a clipping coefficient of 0.2.

All three learning-based methods used the same basic observations, including ego-vehicle states, target parking pose, and local environmental information, with a feature encoding dimension of 128. SAC and PPO employed feature concatenation followed by an MLP for feature fusion, whereas GI-SAC additionally incorporated a coarse HA* trajectory as geometric prior information and adopted intent-driven cross-attention to fuse parking intent with local environmental features. For value estimation, SAC and GI-SAC employed twin Q-networks, while PPO used a state-value network.

4.1.3. Evaluation Metrics

To objectively evaluate parking motion planning algorithms in complex scenarios, task completion and safety are considered fundamental requirements. Decision efficiency distinguishes the interaction steps required during execution, while training efficiency and

control smoothness evaluate the learning performance and command continuity targeted by the proposed method. These requirements motivate the five evaluation perspectives adopted in this study.

Task completion is assessed using the success rate (SR) and timeout rate (TR), while safety is evaluated using the collision rate (CR) and out-of-bounds rate (OOB). These metrics represent the proportions of their corresponding outcomes among all test tasks N_{total}.

Parking success is determined by the overlap ratio between the current and target vehicle footprints. B_v and B_g denote the vehicle footprints at the current and target parking poses, respectively. The coverage ratio is defined as follows:

$$C = \frac{\text{Area}(B_v \cap B_g)}{\text{Area}(B_g)}, \tag{28}$$

when $C > 0.95$ and the vehicle neither collides with an obstacle nor leaves the drivable region, it is considered to have reached the target parking space and completed the parking task successfully.

Decision efficiency is evaluated using successful steps L_{success} and average steps L_{avg}. Successful steps L_{success} denote the mean number of interaction steps required to complete parking over successful episodes and thus measure completion efficiency among successful cases. The corresponding standard deviation σ_{success} reflects consistency across successful episodes. In contrast, average steps L_{avg} denote the mean number of interaction steps over all test episodes, including successful cases and failures caused by timeout, out-of-bounds events, or collisions, thereby measuring overall decision efficiency.

Training efficiency is measured using E_{90}, the number of training episodes required for the success rate to first reach 90%, defined as follows:

$$E_{90} = \min\{e \mid \bar{S}_r(e) \geq 0.9\}, \tag{29}$$

where e denotes the number of training episodes and $\bar{S}_r(e)$ the smoothed success-rate curve at episode e. A smaller E_{90} indicates that fewer training episodes are required to reach the target success rate. If an algorithm never reaches a success rate of 90% during training, E_{90} is reported as N/A.

Control smoothness is evaluated using the mean velocity variation $\overline{\Delta v}$ and the mean front-wheel steering variation $\overline{\Delta \delta}$, which quantify the average absolute changes in velocity and front-wheel steering angle between consecutive control steps, respectively.

4.2. Quantitative Analysis

4.2.1. Overall Performance

To evaluate the task completion capability of the proposed method across different parking scenarios, GI-SAC is compared with four baseline methods, namely RS, HA*, SAC, and PPO, in four types of test scenarios. The evaluation metrics include success rate, collision rate, out-of-bounds rate, and timeout rate, which characterize parking task outcomes and failure modes. As shown in Table 2, GI-SAC achieves the highest success rate across all four scenarios, with an average of 99.05%. Compared with PPO, the best-performing baseline, GI-SAC improves the average success rate by approximately 22.04 percentage points. In the Difficult scenario, PPO and SAC achieve success rates of 61.15% and 47.40%, respectively, whereas HA* achieves only 10.80%. By contrast, GI-SAC consistently maintains a success rate above 98%, with the DLP scenario reaching 98.70%. These results indicate that geometric-prior-guided policy learning provides stronger closed-loop decision-making capability in difficult parking layouts and scenarios constructed from real-world parking space distributions. Further analysis shows that collision rates are low across all methods, providing limited discrimination, while most failures result from boundary violations or

timeouts. In the bounded environment with static obstacles, unsuccessful episodes may terminate upon leaving the valid region or reaching the interaction limit without a collision. Compared with the baseline methods, GI-SAC consistently suppresses both failure types across all four scenarios and remains particularly stable under complex constraints. For example, in the Complex scenario, RS exhibits out-of-bounds and timeout rates of 46.40% and 52.10%, respectively, whereas GI-SAC reduces these rates to 0% and 0.55%. In the DLP scenario, SAC and PPO exhibit timeout rates of 56.80% and 27.75%, respectively, while GI-SAC limits the timeout rate to 1.25%. These results demonstrate that GI-SAC reduces boundary violations and prolonged ineffective exploration, thereby improving the reliability of parking task completion in complex scenarios.

Table 2. Task completion performance of different methods.

Scenario	Method	SR (%) ↑	CR (%) ↓	OOB (%) ↓	TR (%) ↓
Normal	RS	10.40	0	48.60	41.00
	HA*	80.85	0	9.50	9.65
	SAC	79.60	0	0.20	20.20
	PPO	88.75	0	0	11.25
	GI-SAC	**99.75**	0	**0**	**0.25**
Complex	RS	1.50	0	46.40	52.10
	HA*	60.20	0	21.80	18.00
	SAC	71.40	0	0.20	28.40
	PPO	86.05	0	0.05	13.90
	GI-SAC	**99.45**	0	**0**	**0.55**
Difficult	RS	0.30	0	42.10	57.60
	HA*	10.80	0	50.60	38.60
	SAC	47.40	0	0	52.60
	PPO	61.15	0	0	38.85
	GI-SAC	**98.30**	0	**0**	**1.70**
DLP	RS	0.10	0	4.80	95.10
	HA*	85.60	0	5.60	8.80
	SAC	42.80	0	0.40	56.80
	PPO	72.10	0	0.15	27.75
	GI-SAC	**98.70**	0.05	**0**	**1.25**

Note: Each evaluation run comprises 2000 episodes per method per scenario; within each run, 0.05 percentage points correspond to one episode. GI-SAC results are averaged across three independent training runs. ↑ indicates higher is better; ↓ indicates lower is better. Bold values indicate the best performance in each scenario.

The reward and success rate curves in Figure 3 show that GI-SAC reaches high success rates within fewer training episodes and maintains more stable convergence than the standard reinforcement learning baselines. SAC and PPO improve slowly and exhibit substantial fluctuations during training, whereas GI-SAC rapidly reaches regions of high reward and high success rate during the early training stage and maintains a success rate close to 1.0 after convergence. These results indicate that the HA* coarse trajectory provides global directional guidance during early training, thereby alleviating sparse-reward learning and inefficient exploration in long-horizon parking tasks. Meanwhile, the learned policy performs local closed-loop adjustments based on the current state, compensating for the limited adaptability of rule-based planning methods under complex constraints. Overall, as scenario constraints become more complex, rule-based methods and standard reinforcement learning methods exhibit greater reductions in success rate and become more susceptible to boundary violations and timeouts. By contrast, GI-SAC maintains strong task completion performance in both difficult randomized scenarios and scenarios constructed

from real-world parking-space distributions. These results demonstrate that integrating geometric-prior guidance with a learning-based closed-loop policy improves both task completion capability and adaptability in complex autonomous parking scenarios.

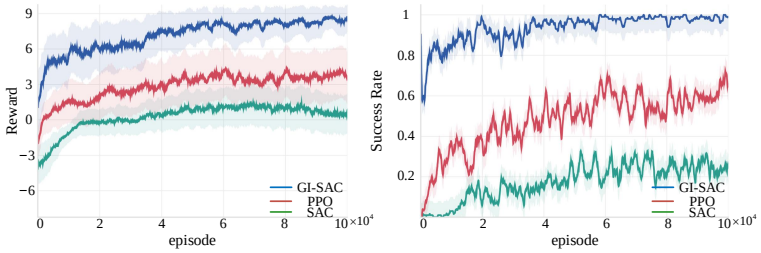


Figure 3. Comparison of training performance across different methods. The solid lines show the smoothed training curves, while the shaded regions represent fluctuations in the original training data.

To further evaluate the parking completion efficiency of the learning-based methods, Table 3 reports the mean interaction steps over successful episodes and over all test episodes. The results show that GI-SAC achieves the lowest values for both metrics across all four scenarios. Specifically, the mean interaction steps of GI-SAC over successful episodes range from 15.79 to 30.75, while those over all test episodes range from 16.25 to 33.63. In the Difficult and DLP scenarios, SAC requires an average of 88.61 and 87.84 interaction steps over successful episodes, respectively, while PPO requires 95.37 and 101.91 steps. By contrast, GI-SAC requires only 30.75 and 25.70 steps in the corresponding scenarios. These results demonstrate that GI-SAC completes parking tasks with fewer environment interaction steps than SAC and PPO, indicating higher parking completion efficiency.

Table 3. Parking efficiency of learning-based methods in four scenarios.

Scenario	Method	$L_{\text{success}} \pm \sigma_{\text{success}} \downarrow$	$L_{\text{avg}} \downarrow$
Normal	SAC	70.59 ± 47.26	96.630
	PPO	61.85 ± 37.56	77.390
	GI-SAC	**15.79 ± 11.88**	**16.252**
Complex	SAC	74.56 ± 48.12	110.240
	PPO	67.10 ± 40.00	85.570
	GI-SAC	**18.15 ± 11.09**	**19.149**
Difficult	SAC	88.61 ± 50.33	147.200
	PPO	95.37 ± 47.25	136.020
	GI-SAC	**30.75 ± 25.26**	**33.625**
DLP	SAC	87.84 ± 48.50	151.370
	PPO	101.91 ± 44.88	129.140
	GI-SAC	**25.70 ± 15.21**	**27.861**

Note: ↓ indicates lower is better. Bold values indicate the best performance in each scenario.

4.2.2. Statistical Reliability and Training Progression

To assess the statistical reliability of GI-SAC, three independent training runs were conducted using different random seeds. Each trained policy was evaluated on 2000 parking tasks per scenario. As shown in Table 4, the standard deviations of the success rates range from 0.09 to 0.55 percentage points across the four scenarios, indicating relatively small performance variations across independent training runs.

Table 4. Success rates of GI-SAC under three different random seeds.

Scenario	Seed 1 (%)	Seed 2 (%)	Seed 3 (%)	Mean ± SD (%)
Normal	99.85	99.50	99.90	99.75 ± 0.22
Complex	99.35	99.50	99.50	99.45 ± 0.09
Difficult	97.75	98.85	98.30	98.30 ± 0.55
DLP	98.60	98.45	99.05	98.70 ± 0.31

Note: SD denotes the sample standard deviation across three independent training runs, expressed in percentage points.

To examine performance variations during training, we evaluated GI-SAC checkpoints saved at 30,000, 60,000, and 100,000 episodes from one of the three independent training runs. As shown in Table 5, the success rates increase across all four scenarios as training progresses. Specifically, in the Difficult and DLP scenarios, the success rates increase from 26.0% and 36.40% at 30,000 episodes to 59.15% and 40.40% at 60,000 episodes, reaching 98.85% and 98.45% at 100,000 episodes, respectively.

Table 5. Success rates of GI-SAC at different training stages.

Episodes / Scenario	30,000 (%)	60,000 (%)	100,000 (%)
Normal	60.40	77.05	99.50
Complex	47.20	71.50	99.50
Difficult	26.0	59.15	98.85
DLP	36.40	40.40	98.45

4.2.3. Parking Performance with Distant Initial Positions

To evaluate parking performance from more distant initial positions, the initial vehicle-to-target distance is extended from 5–20 m to 25–40 m. A total of 300 tasks are randomly generated across the Normal, Complex, and Difficult scenarios. As shown in Table 6, GI-SAC achieves an overall success rate of 80.3%, exceeding SAC and PPO by 46.0 and 19.0 percentage points, respectively. These results demonstrate the effectiveness of GI-SAC in parking from more distant initial positions, consistent with its integration of global trajectory guidance and parking-intent-driven feature representation.

Table 6. Parking success rates with initial vehicle-to-target distances of 25–40 m.

Method	SR (%) ↑
SAC	34.3
PPO	61.3
GI-SAC	**80.3**

Note: Overall success rates are calculated from 300 randomly generated tasks across the Normal, Complex, and Difficult scenarios. ↑ indicates higher is better. Bold values indicate the best performance.

4.2.4. Ablation Study

To analyze the contribution of each module to policy performance, ablation experiments are conducted in the Normal, Difficult, and DLP scenarios, with the results reported in Table 7. The HA* guidance module introduces the coarse-trajectory geometric prior, while the cross-attention mechanism fuses the target intent with environmental features. When neither module is enabled, the model achieves success rates of 79.6%, 47.4% and 42.8% in the three scenarios, respectively, and does not reach a success rate of 90% during training. After introducing HA* guidance, the success rates increase to 99.35%, 96.6% and

91.8%, respectively. The timeout rates decrease overall, and the successful steps are also substantially reduced. These results indicate that geometric-prior guidance improves policy exploration and enhances parking task completion capability. When the cross-attention mechanism is further incorporated into the HA*-guided model, the success rates reach 99.75%, 98.3%, and 98.7% in the three scenarios, respectively. The timeout rates decrease to below 2% in all scenarios, while the successful steps are further reduced to 15.79, 30.75, and 25.70. These results show that cross-attention further improves state feature fusion based on geometric-prior guidance and strengthens decision-making capability in complex scenarios. In particular, in the DLP scenario, the number of training episodes required to first reach a success rate of 90% decreases from 73,030 to 53,303, indicating improved training efficiency.

Table 7. Ablation results of HA* guidance and cross-attention.

Scenario	H	C	SR (%) ↑	TR (%) ↓	E_{90} ↓	Successful Steps ↓
Normal			79.60	20.20	N/A	70.59 ± 47.26
	✓		99.35	0.65	**11,440**	17.01 ± 13.64
	✓	✓	**99.75**	**0.25**	12,662	**15.79 ± 11.88**
Difficult			47.40	52.60	N/A	88.61 ± 50.33
	✓		96.60	3.40	**68,315**	34.36 ± 29.95
	✓	✓	**98.30**	**1.70**	70,586	**30.75 ± 25.26**
DLP			42.80	56.80	N/A	87.84 ± 48.50
	✓		91.80	8.20	73,030	28.77 ± 22.74
	✓	✓	**98.70**	**1.25**	**53,303**	**25.70 ± 15.21**

Note: ✓ indicates that the corresponding module is enabled. H denotes HA* guidance, and C denotes the cross-attention mechanism. Collision and out-of-bounds outcomes are not separately reported in this table; therefore, SR and TR do not necessarily sum to 100%. ↑ indicates higher is better; ↓ indicates lower is better. Bold values indicate the best performance in each scenario.

It should be noted that, in the Normal and Difficult scenarios, the complete model reaches a success rate of 90% later than the model using only HA* guidance. This result indicates that the effect of cross-attention on early convergence is not consistent across scenarios, although it further improves the final policy performance. Overall, HA* guidance facilitates exploration by providing a global trajectory reference and improves parking success rates, while the cross-attention mechanism further enhances feature representation and decision-making performance in complex scenarios.

4.2.5. Action Variation Analysis

To evaluate the effect of the proposed soft temporal action mask under different parking conditions, comparative experiments are conducted across four types of scenarios, and the results are reported in Table 8. Here, s_δ denotes the characteristic scale of inter-step front-wheel steering-angle variations in the soft temporal action mask. The values of s_δ are set to 0.2, 0.3, and 0.4 rad, while the velocity variation scale s_v remains fixed.

The average results across the four scenarios show that the spatiotemporal action mask reduces mean inter-step steering variations during parking. Compared with the configuration using only the spatial action mask, setting $s_\delta = 0.2$ rad reduces $\overline{\Delta\delta}$ from 0.2704 to 0.1608, corresponding to a reduction of approximately 40.5% and indicating improved steering continuity on average.

Table 8. Effects of s_δ on velocity and steering variations.

Configuration	$\overline{\Delta v}$	$\overline{\Delta \delta}$
Spatial	0.2153	0.2704
Spatial + Temporal ($s_\delta = 0.2$)	0.7877	0.1608
Spatial + Temporal ($s_\delta = 0.3$)	0.8697	0.1660
Spatial + Temporal ($s_\delta = 0.4$)	0.9411	0.1792

Note: Spatial and Temporal denote the spatial and soft temporal action masks, respectively; their combination forms the spatiotemporal action mask.

In contrast, $\overline{\Delta v}$ increases from 0.2153 to 0.7877, indicating greater average inter-step velocity changes with the soft temporal action mask. Therefore, the spatiotemporal action mask does not uniformly reduce variation in all action variables but primarily improves steering continuity. Comparisons across steering variation scales show that increasing s_δ from 0.2 to 0.3 and 0.4 rad increases $\overline{\Delta \delta}$ from 0.1608 to 0.1660 and 0.1792, respectively. The corresponding mean velocity variation increases from 0.7877 to 0.8697 and 0.9411. The temporal mask jointly weights velocity and steering variations rather than imposing separate limits on them. Consequently, after accounting for spatial feasibility, a candidate action with a smaller steering change can retain a relatively high sampling weight despite a larger velocity change, allowing improved steering continuity to coexist with increased velocity variation. With the other parameters fixed, increasing s_δ relaxes the suppression of steering changes and redistributes the relative sampling weights of candidate action pairs, thereby also affecting the selection of their velocity components.

Overall, the spatiotemporal action mask improves average steering continuity by reducing inter-step steering variations. However, this improvement is accompanied by increased velocity variation, indicating a trade-off between steering smoothness and velocity smoothness.

4.3. Qualitative Analysis

Figure 4 illustrates the HA* guidance process in representative parking scenarios, where the coarse reference trajectory provides global directional guidance toward the target parking space. The trajectory visualizations further illustrate the differences between GI-SAC and conventional SAC in representative complex parking scenarios. As shown in Figure 5, conventional SAC is prone to local exploration in complex environments and therefore struggles to complete the target parking maneuver reliably. In Figure 5d, for example, the vehicle gradually deviates from the target parking space, indicating that the conventional policy lacks stable global directional guidance for long-horizon parking tasks. In contrast, GI-SAC generates more complete parking maneuvers across different scenarios. As illustrated in Figure 5h, despite the long initial distance and complex obstacle distribution, the vehicle consistently approaches the target region and performs local pose refinement upon reaching the vicinity of the parking space. These results demonstrate that incorporating geometric priors into a learning-based policy improves both target-approach capability and final pose adjustment in complex parking layouts.

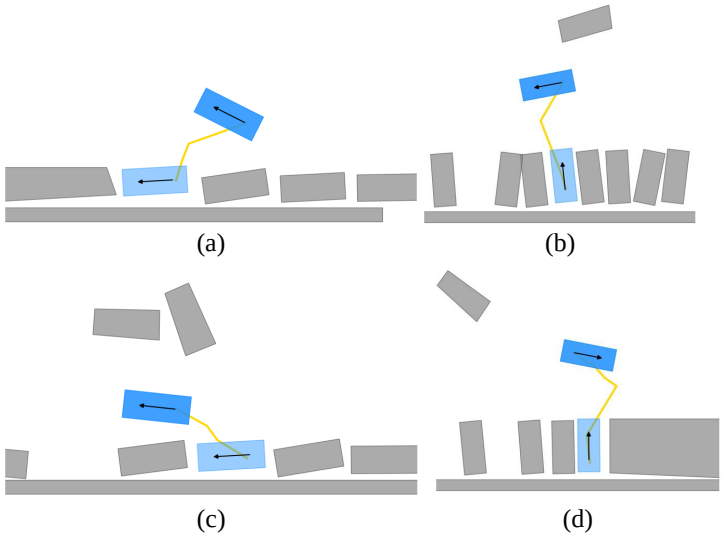


Figure 4. Illustration of HA* guidance in four parking scenarios: (**a**) Scenario 1; (**b**) Scenario 2; (**c**) Scenario 3; (**d**) Scenario 4. Blue rectangles indicate vehicles, gray rectangles represent obstacles, yellow lines show HA* reference paths, and black arrows indicate vehicle headings.

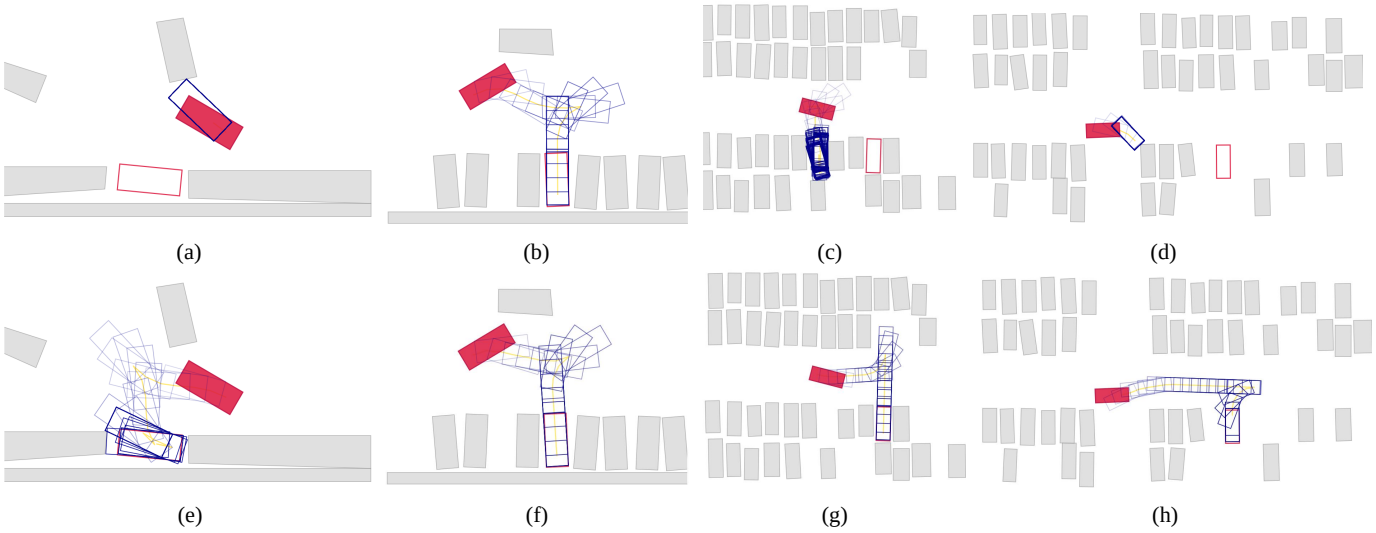


Figure 5. Visual comparison of the parking process between SAC and GI-SAC. (**a**) SAC, Scenario 1; (**b**) SAC, Scenario 2; (**c**) SAC, Scenario 3; (**d**) SAC, Scenario 4; (**e**) GI-SAC, Scenario 1; (**f**) GI-SAC, Scenario 2; (**g**) GI-SAC, Scenario 3; (**h**) GI-SAC, Scenario 4. Red vehicles indicate initial poses, blue outlines show vehicle poses along the trajectories, yellow lines indicate vehicle trajectories, gray rectangles represent obstacles, and red outlined rectangles indicate target parking spaces.

5. Conclusions

In this work, we propose Geometric-Prior-Guided and Intent-Driven Soft Actor-Critic (GI-SAC) to improve policy exploration efficiency, task completion, and steering-command continuity in constrained autonomous parking. The framework integrates a coarse Hybrid A* (HA*) trajectory prior, a guidance reward with a dynamically decaying weight, parking-intent-driven cross-attention, and spatiotemporal action masking. Experiments across four types of parking scenarios demonstrate that GI-SAC achieves an average success rate of 99.05% and consistently outperforms rule-based and learning-based baselines. Ablation studies show that geometric-prior guidance improves learning efficiency and parking success, while cross-attention between parking intent and local environmental features

further improves task completion. Compared with spatial action masking alone, introducing the soft temporal mask with a steering variation scale of 0.2 rad reduces the mean absolute inter-step steering variation by approximately 40.5%, accompanied by increased velocity-command variation. GI-SAC also achieves an overall success rate of 80.3% in the evaluation with initial vehicle-to-target distances of 25–40 m. These results demonstrate that geometric-prior guidance and intent-driven environmental feature fusion improve parking task completion in constrained spaces, while spatiotemporal action masking promotes steering-command continuity.

Future work will incorporate motion prediction into environmental representations and action-feasibility assessment to extend the framework to parking scenarios involving moving vehicles and pedestrians. Coordinated regulation of velocity and steering commands will be investigated to improve longitudinal and lateral control continuity jointly. Trajectory tracking control will also be integrated into high-fidelity simulation and real-vehicle platforms to evaluate the effects of perception noise, vehicle-model mismatch, and actuator delays on parking accuracy and control-command execution.

Author Contributions: Conceptualization, J.G. and Z.Y.; methodology, J.G.; software, J.G.; validation, J.G., Y.Z., M.F., X.Z., H.Y., L.J. and Y.X.; formal analysis, J.G.; investigation, J.G.; resources, J.G.; data curation, J.G.; writing—original draft preparation, J.G.; writing—review and editing, J.G., Y.Z., M.F., X.Z., H.Y., L.J., Y.X. and Z.Y.; visualization, J.G.; supervision, Z.Y.; project administration, Z.Y.; funding acquisition, Z.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Major Program of Guangxi (Grant/Award Number: AA23062030).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code developed in this study is not publicly available but will be provided upon reasonable request to the corresponding author.

Acknowledgments: The authors would like to thank the anonymous reviewers for their constructive comments.

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

Abbreviations

Abbreviation	Definition
GI-SAC	Geometric-Prior-Guided and Intent-Driven Soft Actor-Critic
HA*	Hybrid A*
PPO	Proximal Policy Optimization
RS	Reeds–Shepp
SAC	Soft Actor-Critic
RL	Reinforcement learning
IL	Imitation learning
MDP	Markov decision process
DLP	Dragon Lake Parking
SR	Success rate
CR	Collision rate
OOB	Out-of-bounds rate
TR	Timeout rate

References

1. Cao, Y.; Li, B.; Deng, Z. Optimization-Based Automated Parking Trajectory Planning in Unstructured Environments with Efficient Obstacle Query. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 15422–15435. [CrossRef]
2. de Visser, E.J.; Phillips, E.; Tenhundfeld, N.; Donadio, B.; Barentine, C.; Kim, B.; Madison, A.; Ries, A.; Tossell, C.C. Trust in Automated Parking Systems: A Mixed Methods Evaluation. *Transp. Res. Part F Traffic Psychol. Behav.* **2023**, *96*, 185–199. [CrossRef]
3. Li, B.; Fan, L.; Ouyang, Y.; Tang, S.; Wang, X.; Cao, D.; Wang, F.-Y. Online Competition of Trajectory Planning for Automated Parking: Benchmarks, Achievements, Learned Lessons, and Future Perspectives. *IEEE Trans. Intell. Veh.* **2023**, *8*, 16–21. [CrossRef]
4. Teng, S.; Hu, X.; Deng, P.; Li, B.; Li, Y.; Ai, Y.; Yang, D.; Li, L.; Xuanyuan, Z.; Zhu, F.; et al. Motion Planning for Autonomous Driving: The State of the Art and Future Perspectives. *IEEE Trans. Intell. Veh.* **2023**, *8*, 3692–3711. [CrossRef]
5. Li, S.; Wang, J. Parallel Parking Path Planning in Narrow Space Based on a Three-Stage Curve Interpolation Method. *IEEE Access* **2023**, *11*, 93841–93851. [CrossRef]
6. Wang, Z.; Li, P.; Wang, Z.; Li, Z. APG-RRT: Sampling-Based Path Planning Method for Small Autonomous Vehicle in Closed Scenarios. *IEEE Access* **2024**, *12*, 25731–25739. [CrossRef]
7. Liu, T.; Chai, R.; Chai, S.; Xia, Y. Chance-Constrained Trajectory Optimization for Automatic Parking Based on Conservative Approximation. *IEEE Trans. Veh. Technol.* **2024**, *73*, 6259–6269. [CrossRef]
8. Li, C.; Ji, Z.; Chen, Z.; Qin, T.; Yang, M. ParkingE2E: Camera-Based End-to-End Parking Network, From Images to Planning. In Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024; pp. 13206–13212.
9. Zhong, Y.; Gan, S.; Fan, B.; Cheng, J.; Chen, T.; Shi, Q. Advancing Narrow Space Parking with Latent Memory-Based Reinforcement Learning. In Proceedings of the 2025 IEEE Intelligent Vehicles Symposium (IV), Cluj-Napoca, Romania, 22–25 June 2025; pp. 2273–2280.
10. Shi, J.; Li, K.; Piao, C.; Gao, J.; Chen, L. Model-Based Predictive Control and Reinforcement Learning for Planning Vehicle-Parking Trajectories for Vertical Parking Spaces. *Sensors* **2023**, *23*, 7124. [CrossRef] [PubMed]
11. Yuan, Z.; Wang, Z.; Li, X.; Li, L.; Zhang, L. Hierarchical Trajectory Planning for Narrow-Space Automated Parking with Deep Reinforcement Learning: A Federated Learning Scheme. *Sensors* **2023**, *23*, 4087. [CrossRef] [PubMed]
12. Lu, W.; Chen, L.; Wang, Y.; Wei, Y.; Wu, Z.; Xiong, R.; Wang, Y. Demonstration Data-Driven Parameter Adjustment for Trajectory Planning in Highly Constrained Environments. *IEEE Robot. Autom. Lett.* **2024**, *9*, 11641–11648. [CrossRef]
13. Jiang, M.; Li, Y.; Zhang, S.; Chen, S.; Wang, C.; Yang, M. HOPE: A Reinforcement Learning-Based Hybrid Policy Path Planner for Diverse Parking Scenarios. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 6130–6141. [CrossRef]
14. Wang, Z.; Chen, Z.; Jiang, M.; Qin, T.; Yang, M. RL-OGM-Parking: LiDAR OGM-Based Hybrid Reinforcement Learning Planner for Autonomous Parking. In *Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2025; pp. 8420–8426.
15. Tian, D.; Guvenc, L. A Hierarchical NMPC and TD3-Based Framework for Seamless Cruise-to-Park Automated Valet Parking. *Sensors* **2026**, *26*, 3409. [CrossRef] [PubMed]
16. Özturan, M.Y.O.; Dirik, A.E. Suppressing Control Oscillations in Reinforcement Learning via External Action Filtering for Autonomous Driving. In *Proceedings of the 2025 16th International Conference on Electrical and Electronics Engineering (ELECO), Istanbul, Türkiye, 27–29 November 2025*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–5.
17. Dubins, L.E. On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *Am. J. Math.* **1957**, *79*, 497–516. [CrossRef]
18. Reeds, J.; Shepp, L. Optimal Paths for a Car That Goes Both Forwards and Backwards. *Pac. J. Math.* **1990**, *145*, 367–393. [CrossRef]
19. Zhang, J.; Shi, Z.; Yang, X.; Zhao, J. Trajectory Planning and Tracking Control for Autonomous Parallel Parking of a Non-Holonomic Vehicle. *Meas. Control* **2020**, *53*, 1800–1816. [CrossRef]
20. Li, C.; Jiang, H.; Ma, S.; Jiang, S.; Li, Y. Automatic Parking Path Planning and Tracking Control Research for Intelligent Vehicles. *Appl. Sci.* **2020**, *10*, 9100. [CrossRef]
21. Dolgov, D.; Thrun, S.; Montemerlo, M.; Diebel, J. Practical Search Techniques in Path Planning for Autonomous Driving. In Proceedings of the First International Symposium on Search Techniques in Artificial Intelligence and Robotics (STAIR), Chicago, IL, USA, 13–14 July 2008; pp. 32–37.
22. Jin, W.; Li, J.; Lv, X.; Zhang, T. An Optimized Path Planning Approach for Automatic Parking Using Hybrid A* Bidirectional Search. *Appl. Intell.* **2025**, *55*, 201. [CrossRef]
23. Meng, T.; Yang, T.; Huang, J.; Jin, W.; Zhang, W.; Jia, Y.; Wan, K.; Xiao, G.; Yang, D.; Zhong, Z. Improved Hybrid A-Star Algorithm for Path Planning in Autonomous Parking System Based on Multi-Stage Dynamic Optimization. *Int. J. Automot. Technol.* **2023**, *24*, 459–468. [CrossRef]
24. Wang, H.; Lou, S.; Jing, J.; Wang, Y.; Liu, W.; Liu, T. The EBS-A* Algorithm: An Improved A* Algorithm for Path Planning. *PLoS ONE* **2022**, *17*, e0263841. [CrossRef] [PubMed]

25. Li, B.; Wang, K.; Shao, Z. Time-Optimal Maneuver Planning in Automatic Parallel Parking Using a Simultaneous Dynamic Optimization Approach. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3263–3274. [CrossRef]
26. Li, B.; Acarman, T.; Zhang, Y.; Ouyang, Y.; Yaman, C.; Kong, Q.; Zhong, X.; Peng, X. Optimization-Based Trajectory Planning for Autonomous Parking with Irregularly Placed Obstacles: A Lightweight Iterative Framework. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 11970–11981. [CrossRef]
27. Ren, H.; Niu, Y.; Li, Y.; Yang, L.; Gao, H. Automatic Parking Trajectory Planning Based on Warm Start Nonlinear Dynamic Optimization. *Sensors* **2025**, *25*, 112. [CrossRef] [PubMed]
28. Zhang, P.; Zhou, S.; Hu, J.; Zhao, W.; Zheng, J.; Zhang, Z.; Gao, C. Automatic Parking Trajectory Planning in Narrow Spaces Based on Hybrid A* and NMPC. *Sci. Rep.* **2025**, *15*, 1384. [CrossRef] [PubMed]
29. Zhang, X.; Liniger, A.; Sakai, A.; Borrelli, F. Autonomous Parking Using Optimization-Based Collision Avoidance. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018; pp. 4327–4332.
30. Nakanishi, R.; Sakib, N.; Bakibillah, A.S.M.; Kamal, M.A.S.; Yamada, K. Reinforcement Learning-Based Dynamic Parking Allocation for Sustainable and User-Centric Smart Cities. *IET Conf. Proc.* **2025**, *2025*, 200–206. [CrossRef]
31. Sakib, N.; Bakibillah, A.S.M.; Susilawati, S.; Kamal, M.A.S.; Yamada, K. Eco-Friendly Smart Car Parking Management System with Enhanced Sustainability. *Sustainability* **2024**, *16*, 4145. [CrossRef]
32. Anagnostara, I.M.; Tsardoulias, E.; Symeonidis, A.L. Deep Reinforcement Learning and Imitation Learning for Autonomous Parking Simulation. *Electronics* **2025**, *14*, 1992. [CrossRef]
33. Chai, R.; Liu, D.; Liu, T.; Tsourdos, A.; Xia, Y.; Chai, S. Deep Learning-Based Trajectory Planning and Control for Autonomous Ground Vehicle Parking Maneuver. *IEEE Trans. Autom. Sci. Eng.* **2023**, *20*, 1633–1647. [CrossRef]
34. Yang, Y.; Chen, D.; Qin, T.; Mu, X.; Xu, C.; Yang, M. E2E Parking: Autonomous Parking by the End-to-End Neural Network on the CARLA Simulator. In Proceedings of the 2024 IEEE Intelligent Vehicles Symposium (IV), Jeju Island, Republic of Korea, 2–5 June 2024; pp. 2375–2382.
35. Du, H.; Chew, C.-M. TransParking: A Dual-Decoder Transformer Framework with Soft Localization for End-to-End Automatic Parking. *arXiv* **2025**, arXiv:2503.06071.
36. Albilani, M.; Bouzeghoub, A. Dynamic Adjustment of Reward Function for Proximal Policy Optimization with Imitation Learning: Application to Automated Parking Systems. In Proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 5–9 June 2022; pp. 1400–1408.
37. Qiu, Z.; Chen, S.; Shi, J.; Wang, F.; Zheng, N. Human-Like Reverse Parking Using Deep Reinforcement Learning with Attention Mechanism. In Proceedings of the 2024 IEEE Intelligent Vehicles Symposium (IV), Jeju Island, Republic of Korea, 2–5 June 2024; pp. 2553–2560.
38. Xu, X.; Zhao, Y.; Zhang, X.; Song, R.; Zhang, W. HRL-KCAP: Hierarchical Reinforcement Learning with Kinematic Constraints for Adaptive Parking. *IEEE Trans. Ind. Electron.* **2026**, *73*, 5771–5781. [CrossRef]
39. Du, Z.; Miao, Q.; Zong, C. Trajectory Planning for Automated Parking Systems Using Deep Reinforcement Learning. *Int. J. Automot. Technol.* **2020**, *21*, 881–887. [CrossRef]
40. Tang, X.; Yang, Y.; Liu, T.; Lin, X.; Yang, K.; Li, S. Path Planning and Tracking Control for Parking via Soft Actor-Critic Under Non-Ideal Scenarios. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 181–195. [CrossRef]
41. Al-Mousa, A.; Arrabi, A.; Daoud, H. A Reinforcement Learning-Based Reverse-Parking System for Autonomous Vehicles. *IET Intell. Transp. Syst.* **2025**, *19*, e12614. [CrossRef]
42. Li, Y.; Zhang, S.; Jiang, M.; Chen, X.; Yang, J.; Qian, Y.; Wang, C.; Yang, M. Tactics2D: A Highly Modular and Extensible Simulator for Driving Decision-Making. *IEEE Trans. Intell. Veh.* **2024**, *9*, 4840–4844. [CrossRef]
43. *GB/T 47031–2026*; Intelligent and Connected Vehicles—Performance Requirements and Test Methods for Automated Parking System. State Administration for Market Regulation; Standardization Administration of China: Beijing, China, 2026.
44. *JGJ 100–2015*; Code for Design of Parking Garage Building. Ministry of Housing and Urban-Rural Development of the People's Republic of China: Beijing, China, 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.