World Electric Vehicle Journal

MDPI

Article

Improving Reinforcement Learning with Expert Demonstrations and Vision Transformers for Autonomous Vehicle Control

Badr Ben Elallid [1,*], Nabil Benamar [1,2], Miloud Bagaa [3], Sousso Kelouwani [3] and Nabil Mrani [1]

[1] Computer Sciences at the School of Technology, Moulay Ismail University of Meknes, Meknes 50050, Morocco; n.benamar@umi.ac.ma (N.B.); n.mrani@umi.ac.ma (N.M.)
[2] Computer Science, Al Akhawayn University in Ifrane, Ifrane 53000, Morocco
[3] Department of Electrical and Computer Engineering, Université du Québec à Trois-Rivières, Québec City, QC G9A 5H7, Canada; miloud.bagaa@uqtr.ca (M.B.); sousso.kelouwani@uqtr.ca (S.K.)
[*] Correspondence: badr.benelallid@edu.umi.ac.ma

Abstract: While IL has been successfully applied in RL-based approaches for autonomous driving, significant challenges, such as limited data for RL and poor generalization in IL, still need further investigation. To overcome these limitations, we propose in this paper a novel approach that effectively combines IL with DRL by incorporating expert demonstration data to control AV in roundabout and right-turn intersection scenarios. Instead of employing CNNs, we integrate a ViT into the perception module of the SAC algorithm to extract key features from environmental images. The ViT algorithm excels in identifying relationships across different parts of an image, thereby enhancing environmental understanding, which leads to more accurate and precise decision making. Consequently, our approach not only boosts the performance of the DRL model but also accelerates its convergence, improving the overall efficiency and effectiveness of AVs in roundabouts and right-turn intersections with dense traffic by a achieving high success rate and low collision compared to RL baseline algorithms.

Keywords: autonomous driving; deep reinforcement learning; vision transformer; imitation learning

check for updates

Citation: Elallid, B.B.; Benamar, N.; Bagaa, M.; Kelouwani, S.; Mrani, N. Improving Reinforcement Learning with Expert Demonstrations and Vision Transformers for Autonomous Vehicle Control. *World Electr. Veh. J.* **2024**, *15*, 585. https://doi.org/10.3390/wevj15120585

Joeri Van Mierlo and Michael Fowler

Received: 16 September 2024
Revised: 14 December 2024
Accepted: 17 December 2024
Published: 19 December 2024

1. Introduction

Autonomous vehicles have garnered substantial attention from different stakeholders, namely, governments, industries, academia, and car manufacturers, primarily due to their potential to revolutionize transportation through advancements in AI and computer hardware [1]. The deployment of AVs is aimed at reducing traffic accidents [2,3], enhancing road safety [4,5], and improving mobility in densely populated urban areas [6]. Current vehicle control systems rely on two primary paradigms: IL [7] and RL [8].

IL techniques involve collecting data from expert human drivers and training DL to replicate the expert's actions given specific states [9]. While IL can be effective, it requires vast amounts of data encompassing all possible driving situations [10,11]. On the other hand, RL enables an agent to learn decision making by interacting with its environment, utilizing past experiences to guide future actions [12]. RL has demonstrated impressive results, particularly in autonomous navigation, by employing CNNs to process sensor data from AVs, such as laser scans or visual images [13]. However, RL faces challenges, including data inefficiency and slow convergence, especially in complex scenarios like controlling AVs in dense traffic. Additionally, using CNNs in RL may result in the loss of critical environmental information, leading to suboptimal decision making.

Recently, self-attention-based methods, particularly Transformers, have emerged as the preferred models in natural language processing [14]. The computer vision community has adapted the standard Transformer architecture for image processing, leading to the development of the ViT [15]. The ViT has shown promising results across various fields, including robotic manipulation, due to its ability to handle image inputs effectively [16].

Unlike CNNs, which primarily focus on local features through convolutional filters, a ViT leverages self-attention mechanisms to capture global relationships across the entire image. This capability is particularly advantageous for complex tasks like understanding traffic scenes in autonomous driving. For instance, in scenarios involving multi-lane roundabouts or intersections, the ability to model long-range dependencies enables the ViT to analyze the entire environment holistically. By capturing patterns dispersed throughout the image, the ViT offers a more comprehensive understanding of the scene, which is crucial for accurate decision making in autonomous driving.

To address the limitations of existing approaches, we propose a novel method that combines both IL and RL, utilizing the ViT model in place of the traditional CNN. Our experiments were conducted in a roundabout and right-turn intersection scenarios with high traffic mobility. Initially, we collected data from expert human drivers and trained a policy network to minimize the distance between states and actions. We then integrated a pre-trained model and expert demonstration replay buffer into our RL framework to enhance the learning of real-world driving skills. The results demonstrate that our approach accelerates the learning process and achieves low collisions in dense traffic, outperforming RL baseline algorithms. The contributions of this work are listed as follows:

1. We propose a novel approach that combines RL and IL to control pAVs in roundabouts and right-turn intersection with high mobility.
2. We integrate a demonstration data buffer into the SAC algorithm to learn driving skills from expert human drivers.
3. We replace the CNN with a ViT to extract the most important features from the visual state and combine them with the goal vector, enabling the model to better understand the environment. The results demonstrate significant improvement in learning, with the model achieving a high success rate in various traffic densities within the roundabout and right-turn intersection scenarios, surpassing baseline RL algorithms.

The remainder of this paper is organized as follows. In Section 2, we review the existing literature, analyzing the strengths and weaknesses of each study and underscoring the novelty of our proposed approach. Section 3 provides an overview of RL, IL, and ViTs. In Section 4, we present our ViT-SAC algorithm, detailing the state, action, and reward structures. Section 5 describes our experimental setup, and Section 6 discusses the experimental results. Finally, Section 7 summarizes our findings and conclusions.

2. Related Work

Due to their strong ability to represent high-dimensional states and their superior data efficiency, DRL algorithms are attracting growing interest within the robotics community. Previous research has shown that model-free DRL approaches hold considerable promise for autonomous motion control strategies, leading to significant focus on DRL-based methods.

Huang et al. [17] proposed a novel framework integrating human prior knowledge into DRL for achieving human-like autonomous driving, significantly improving sample efficiency and simplifying reward function design, validated in simulated urban driving scenarios. Perez et al. [18] used DRL, specifically a DQN and DDPG, in AV control using the CARLA simulator, achieving efficient navigation with the DDPG outperforming DQN. Ben et al. [19] presented a DQN approach for guiding AVs through intersections with dense traffic and various road users. In a subsequent study [20], the authors extended their work by training the model under a diverse daytime and weather conditions, using the same intersection scenario.

In order to improve the control of the AV, Liu et al. [21] proposed an RL-based control method that generates both steering and acceleration commands simultaneously using the SAC algorithm and integrates ResNet and DenseNet for improving feature transmission between layers, which helps in capturing and utilizing high-level features. In [22], the authors used a CNN to extract the main feature in the SAC algorithm to control the AV to enter roundabouts.

ViT-based architecture has demonstrated its dominance in enhancing scene representation and analysis, driving progress in computer vision and robotics. Driving in complex urban environments requires complex decision making, which can be enhanced using ViT networks. In [23], a ViT was utilized to process birds-eye-view images and improve scene understanding and decision making by leveraging its attention mechanism, leading to faster and more effective learning compared to traditional ConvNet-based methods.

In [24], a scene representation Transformer was presented to boost RL for autonomous driving by capturing complex scene interactions and predicting future scenarios. This model efficiently learns from data and makes better decisions by focusing on relevant information.

Huang et al. [25] proposed an approach called the Goal-Guided Transformer to navigate an AV toward various destinations. Their model incorporates a ViT within the perception module and is pre-trained using expert demonstrations to accelerate the training of the RL model

However, several previous works have not utilized expert demonstrations to accelerate the training of an RL model or to leverage expert driving skills. Additionally, some studies focused on navigating the AV without accounting for dynamic obstacles.

In this study, we selected the ViT instead of a CNN because of its superior ability to capture global relationships in images through self-attention mechanisms, which is particularly beneficial for complex tasks like understanding traffic scenes in autonomous driving. Unlike CNNs, which focus primarily on local features through convolutional filters, the ViT can model long-range dependencies across an entire image, making it more effective for analyzing complex environments such as multi-lane roundabouts or intersections. This ability to capture global context gives the ViT an edge in identifying patterns that are spread out across the image, which is critical in autonomous driving scenarios, where understanding the entire scene rather than just localized objects is essential for accurate decision making.

In light of the existing literature, the preceding works have focused on using the CNN model to capture the main feature of the environment, and some of them used a ViT to control the AV in a simple scenario without taking into account other road users. To our best knowledge, no current work combines a ViT with RL techniques to control AVs to enter roundabouts and turn right in the intersection while existing with other road users, such as other vehicles, cyclists, and motorcycles.

3. Preliminaries

3.1. Reinforcement Learning

RL aims to develop an effective strategy (known as a policy) by engaging with the controlled environment. This interaction can be modeled as a discrete-time MDP, which is characterized by a set of elements: $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$. At each time step t, the RL agent receives information about the current state of the environment, represented by the state variable s_t, from the set of all possible states $\mathcal{S}$. In response, the agent takes a control action a_t from the set of available actions $\mathcal{A}$. Following the agent's action, the environment provides a reward signal $r_t \in \mathcal{R}$. The environment then changes its state according to the state transition dynamics $\mathcal{P}$, moving from the current state s_t to a new state s_{t+1} [26].

In this context, the objective is to identify a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$ that, at any time step t, selects an action $a_t \sim \pi(\cdot|s_t)$ to maximize the discounted cumulative reward $\sum_{t=0}^{\infty} \gamma^t r_t$, where $\gamma \in (0,1]$ is the discount factor.

RL is a model-free approach that learns optimal policies directly from interaction with the environment without requiring a probabilistic model. This makes RL highly adaptable to unfamiliar and uncertain environments, a critical advantage in autonomous driving. RL typically consists of two phases: exploration to discover the environment and exploitation to leverage learned knowledge for optimal behavior.

DRL algorithms can be categorized into discrete (e.g., DQN double DQN [27]) and continuous (e.g., A3C, DDPG [28], PPO [29]) types. Many continuous algorithms use the actor–critic framework, where an actor network predicts actions based on the current state, and a critic network evaluates the quality of those actions. A DDPG is a prominent continuous DRL algorithm that combines the elements of a DQN and policy gradients. It learns a deterministic policy using off-policy data and the Bellman equation [30].

3.2. Imitation Learning

IL is based on a behavior cloning technique that tries to train the policy $\pi(a|s;\theta)$ to mimic the expert's policy $\pi^E(a|s)$ using a supervised learning method [31]. Based on demonstration data, which are collected from an expert human driver as a set of image state-action pairs, $\mathcal{D}^E = \{\langle s_i^E, a_i^E \rangle\}_{i=1}^{N}$, where N is the number of samples. Therefore, the objective of supervised learning is to reduce the distance between action a_i and function approximation of neural networks $\mathcal{F}(s_i, \theta)$:

$$arg \min_{\theta} \sum_{i=1}^{N} \mathcal{L}\left(\mathcal{F}(s_i, \theta), a_i^E\right) \tag{1}$$

where $\mathcal{L}$ represents the loss function, and a^E is an action from the expert human driver.

3.3. Vision Transformer

The Transformer architecture, introduced by Vaswani et al [14], has become a dominant model in NLP. Building on the effectiveness of self-attention-based deep neural networks, Dosovitskiy et al. [15] introduced the ViT for image classification. This model processes images by dividing them into patches and treating each embedded patch similarly to how words are handled in NLP. Self-attention mechanisms are employed to capture relationships between these embedded patches. The ViT takes the input image $X \in \mathbb{R}^{H \times W \times C}$ and reshapes it into of sequence of $(X_1, X_2, \ldots, X_n)$, where H is height; W, width; and C, the number of channels of the image X. $X_n \in \mathbb{R}^{N \times (P^2 \times C)}$ denotes flattened patches, where (P, P) is the size of each patch, and $N = \frac{H \cdot W}{P^2}$ is the number of patches and the input sequence length.

Therefore, the input of the ViT encoder is obtained by adding position embeddings $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$ to D-dimensional flattened 2D patches:

$$Z_0 = [\mathrm{X}_0; \mathbf{LP}(X_1); \mathbf{LP}(X_2); \cdots; \mathbf{LP}(X_n)] + \mathbf{E}_{\text{pos}} \tag{2}$$

where $\mathbf{LP}$ is a linear projection, and $X_0 \in \mathbb{R}^{1 \times D}$ is an additional learnable embedding known as the class token. By inputting the embedded patches into the standard Transformer encoder, we can obtain multi-head self-attention (MSA) through the SA mechanism.

$$\mathrm{MSA}(Q, K, V) = \mathbf{LP}([\mathbf{Atten}_1(Q, K, V); \mathbf{Atten}_2(Q, K, V); \cdots; \mathbf{Atten}_k(Q, K, V)]) \tag{3}$$

Here, k represents the k-th head, and **Atten** refers to the SA mechanism. As shown in [14], SA is computed using the query Q, keys K, and values V.

$$\mathbf{Atten}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

$$[Q, K, V] = \mathbf{LP}(Z)$$

Here, Z refers a set of embedded patches, and d_k is a scaling factor.

4. Framework: Reinforcement Learning with Vision Transformer

The primary objective of this work is to accelerate the training process of RL, specifically the SAC model, by incorporating a replay buffer filled with data from an expert human driver as shown in Figure 1. This approach enables the RL model to acquire driving skills directly from the expert's experience. Additionally, we replace the traditional CNN with a ViT model, which excels in recognizing relationships across different parts of an image. This enhancement is crucial for AVs as it allows them to better understand their environment and make more informed decisions.

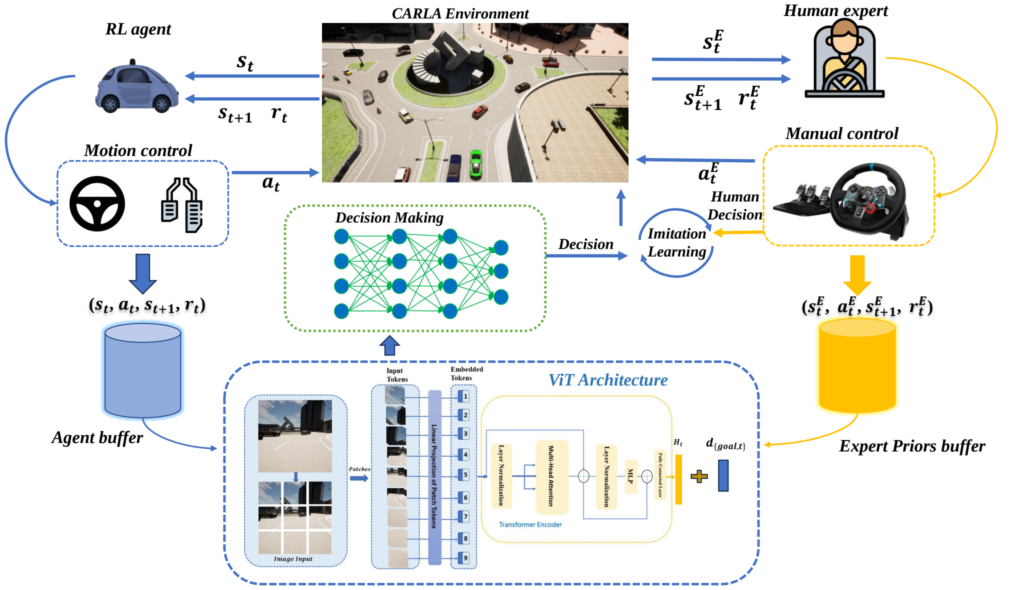


Figure 1. Overview of the ViT SAC with expert demonstration: The input images, s_t, are fed into the ViT to extract relevant latent features, which are then concatenated with the goal vector, $d_{goal,t}$. This combined vector guides the AV toward its target destination. The final vector is passed to the policy network to determine the action a_t.

In the rest of this section, we describe the state, action, and reward used in our approach.

4.1. State Space

Our model's input consists of two main components: visual states in the form of RGB images and a destination vector. Specifically, we use 84 × 84 RGB images captured with a camera, selecting a stack of the four most recent frames. These images are divided into patches, each with a size of 14 × 14 pixels. The patches are then flattened, resulting in embedded tokens. These tokens are fed into a Transformer encoder, which includes layer normalization, multi-head attention, and a Multi-Layer Perceptron (MLP). Finally, we reach a fully connected layer that is a vector with dimension $H_t \in \mathbb{R}^{1\times 256}$, as shown in Figure 2. We can represent this as function $\mathbf{F}_{ViT}$, where

$$\mathbf{F}_{ViT}(s_t) = H_t \tag{5}$$

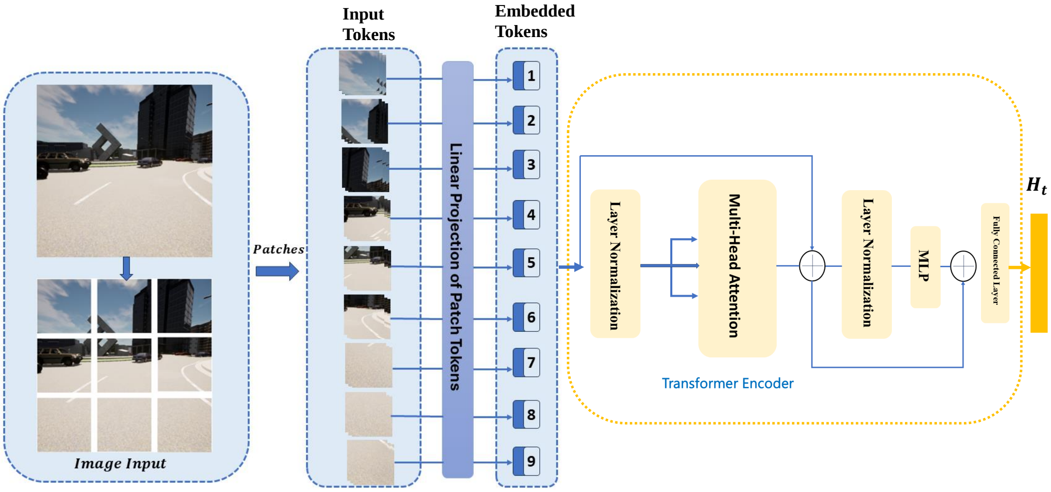


Figure 2. The architecture of the ViT network splits images into patches and feeds them to a Transformer encoder to obtain the main features of the environment.

Here, $s_t \in \mathbb{R}^{4\times84\times84}$ and $H_t \in \mathbb{R}^{1\times256}$.

In order to guide our AV to reach its destination, we introduce a goal vector $d_{goal} \in \mathbb{R}^{1\times2}$. This goal vector is defined as $[X_{goal} - X_{AV}, Y_{goal} - Y_{AV}]$, where X_{goal} and Y_{goal} represent the X and Y coordinates of the goal position, while X_{AV} and Y_{AV} are the corresponding coordinates of the AV at each time step t. We concatenate the result vector H_t from Equation (5) with goal vector d_{goal}. Therefore, function 5 can be updated as follows:

$$\mathbf{F}_{ViT+goal}(s_t, d_{goal}) = D_t, \quad D_t \in \mathbb{R}^{1\times258} \tag{6}$$

4.2. Action Space

The SAC algorithm, a continuous DRL approach, requires continuous actions. Consequently, at each time step t, the agent chooses action a_t by sending the control signal to the AV, which includes ranges of acceleration $[0,1]$, steering $[-1,1]$, and braking $[0,1]$.

4.3. Reward Function

We carefully designed our reward to ensure that our model converges based on [32]. More specifically, the reward combines between five components as follows:

$$R_t = R_V + R_d + R_c + R_{or} + R_{ol} + R_g \tag{7}$$

For simplicity, we define the current distance to the destination as d_{cu} and the previous distance as d_{pre}. The AV's forward speed is represented by v_{speed}, while indicators for whether the AV is off-road or in another lane are denoted by $M_{offroad}$ and $M_{otherlane}$, respectively. In the event of a collision, a penalty labeled $C_{collision}$ is applied, which adjusts the reward (R_c) accordingly.

Additionally, if the AV moves closer to the goal $(d_{pre} > d_{cu})$, it receives an increased reward (R_d) to encourage progress toward the destination. The AV's speed should not exceed the limit of 3 (derived from 30 divided by 10), which is enforced by R_v.

To ensure that the AV stays in its lane, minor penalties are applied if the AV travels in an unauthorized lane or off-road, represented by R_{or} and R_{ol}. Finally, if the AV successfully reaches its intended destination, it is rewarded with $R_g = 100$. The reward function can be represented as follows:

$$reward = \begin{cases} R_c = -100 \\ R_d = d_{pre} - d_{cu} \\ R_v = \frac{v_{velocity}}{10} & v_{velocity} \in [0, v_{limit}] \\ R_{or} = -0.05 \times M_{offroad} \\ R_{ol} = -0.05 \times M_{otherlane} \\ R_g = 100 \end{cases} \tag{8}$$

Next, using the latent features D_t extracted from ViT-SAC at a given timestep t, the SAC algorithm learns the decision policy $\pi(a_t \mid D_t)$ according to the previously mentioned reward function. A widely adopted technique in the SAC algorithm is the use of double Q-networks to address the problem of overestimation. Consequently, the parameters of the critic network in ViT-SAC are optimized by minimizing the mean Bellman squared error (MBSE) loss function.

$$\mathcal{L}(\theta_i) = \mathbb{E}_{s_t \sim \mathcal{P}, a_t \sim \pi} \left\| Q^{\pi}_{\theta_i}(s_t, a_t) - \left(r_t + \gamma \cdot \hat{Q}^{\pi}\right) \right\|_2 \tag{9}$$

where $\hat{Q}^{\pi}$ is the state-action value of the next step from the double target Q-networks, which is computed as follows:

$$\hat{Q}^{\pi} = \mathbb{E}_{s_{t+1} \sim \mathcal{P}, a_{t+1} \sim \pi} \left(\min_{i=1,2} Q_{\text{target},i}(s_{t+1}, a_{t+1}) - \alpha \log \pi(a_{t+1} | s_{t+1}) \right) \tag{10}$$

Here, α represents a temperature parameter that balances the trade-off between the randomness of the optimal policy and the state-action value. Consequently, the actor network adjusts its parameters by maximizing the soft state-action function:

$$\mathcal{L}(\phi) = \mathbb{E}_{s_t \sim \mathcal{P}, a_t \sim \pi_{\phi}(\cdot | s_t)} \left[\alpha \log \pi_{\phi}(s_t \mid a_t) - \min_{i=1,2} Q^{\pi}(s_t, a_t) \right] \tag{11}$$

To better understand the implementation of our approach, including the expert demonstration buffer, we provide Algorithm 1.

Algorithm 1 Vision Transformer Soft Actor-Critic with Imitation Learning

```
Initialize the parameters for the actor network $\phi$ and the critic network $\theta$.
Set the initial value of $\alpha$.
Define the batch size $N$ and initialize an empty replay buffer $\mathcal{D}^A$ and expert demonstration buffer $\mathcal{D}^E$.
Initialize the target network parameters: $\theta_{\text{target}} \leftarrow \theta$.
Time to update critic networks: $Time_{cri}$
Time to update actor networks: $Time_{act}$
Time to update target networks: $Time_{tar}$
for episode $e = 1$ to $Ep$ do
    Start with an initial image state $s_t \sim$ Env
    Set an initial goal vector $d_{\text{goal}} \sim$ Env
    for each time step $t$ from 1 to $N_{\text{steps}}$ do
        $D_t \leftarrow \mathbf{F}_{ViT}(s_t, d_{\text{goal}})$
        Choose an action $a_t \sim \pi_{\phi}(a_t | D_t)$
        Execute the action in the environment:
        Receive the next observation $s_{t+1}$, reward $r_t$, and updated the goal $d_{(\text{goal},t+1)}$
        Add the tuple to the replay buffer:
        $\mathcal{D} \leftarrow \mathcal{D} \cup (s_t, d_{(\text{goal},t)}, a_t, r_t, s_{t+1}, d_{(\text{goal},t+1)})$
        if $Time_{cri}$ then
            Chose a batch $(s^i_t, d^i_{(\text{goal},t)}, a^i_t, r^i_t, s^i_{t+1}, d^i_{(\text{goal},t+1)})^N_{i=1}$ from $\mathcal{D}^A \cup \mathcal{D}^E$
            Calculate the critic loss $\mathcal{L}(\theta)$ using Equation (9)
```

Algorithm 1 *Cont.*

```
21:            Update the parameters $\theta$
22:        end if
23:        if $Time_{act}$ then
24:            Chose a batch $(s_t^i, d_{(\text{goal},t)}, a_t^i, r_t^i, s_{t+1}^i, d^i_{(\text{goal},t+1)})_{i=1}^N$ from $\mathcal{D}^A \cup \mathcal{D}^E$
25:            Compute the actor loss $\mathcal{L}(\phi)$ using Equation (11)
26:            Update the parameters $\phi$
27:            if automatic entropy adjustment is active then
28:                Update $\alpha$
29:            end if
30:        end if
31:        if $Time_{tar}$ then
32:            Update the target network: $\theta_{\text{target}} \leftarrow \theta$
33:        end if
34:    end for
35: end for
```

5. Experiments

In this section, we describe our simulation, including the simulator used to train and test our approach. We also present a thorough analysis of the results achieved during our experiment.

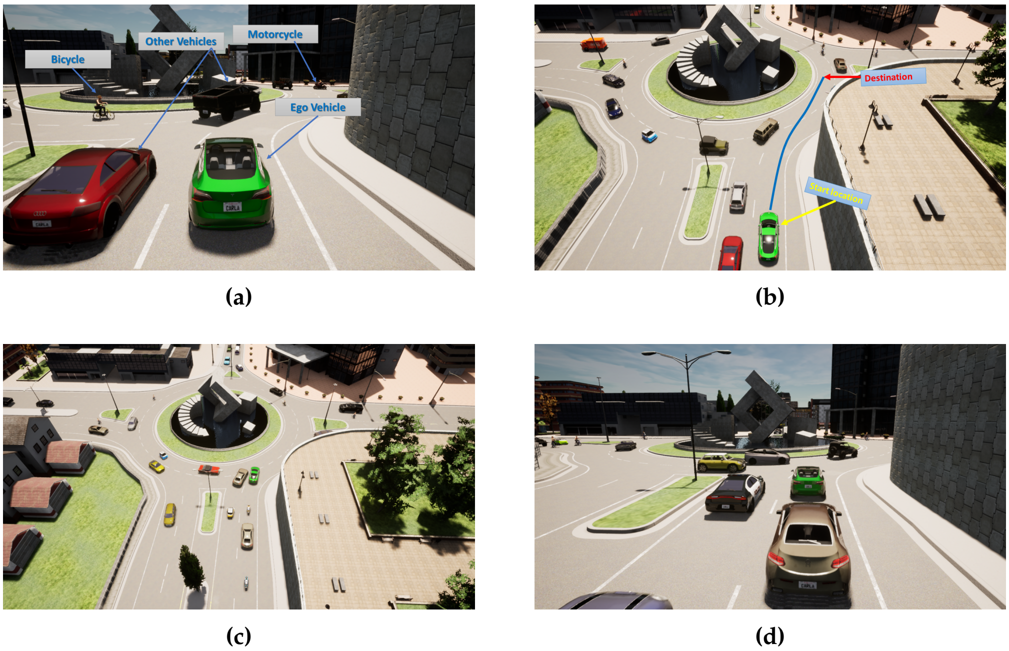
5.1. Expert Demonstration

To mimic the behavior of an expert human driver, we control the AV using our laptop keyboard, guiding it through a roundabout and turning right in the intersection and avoiding collisions with other road users. We collect a total of 13,542 samples of data in the form of tuples $(s_t, d_{(\text{goal},t)}, a_t, r_t, s_{t+1}, d_{(\text{goal},t+1)})$, which are stored in the expert demonstration buffer $\mathcal{D}^E$. Additionally, we selected state-action pairs $\{\langle s_i^E, a_i^E \rangle\}$ for IL, which were split into training and validation sets for training the actor network. Once the model was trained, we integrated the pre-trained model into the actor network of our RL model. This approach reduces the discrepancy between states and actions, thereby accelerating the training and convergence of the RL model.

5.2. Simulation Experiments

The simulation experiments in this work conducted training using a laptop computer equipped with an 11th Gen Intel (R) Core (TM) i7-11800h 2.30 GHz CPU, 16 GB of RAM, and NVIDIA GeForce RTX 3050 laptop GPU hardware. We utilized the CARLA simulator [33] as our simulation environment, where the autonomous vehicle was assigned the task of navigating safely and efficiently through a roundabout and turning right with multiple intersections in an urban setting, as depicted in Figures 3 and 4. While the starting and destination points were consistent, the traffic conditions varied across different training episodes. The ego vehicle was initially spawned randomly within the starting area and had to follow the designated route to the destination, all while avoiding collisions with other vehicles in the dense traffic. The roundabout and the intersection contained various road participants, including cyclists, motorcycles, and other vehicles. For this traffic, we chose 120 vehicles moving in randomly, including motorcycles, cyclists, and four-wheeled vehicles, all controlled by the autopilot of CARLA.

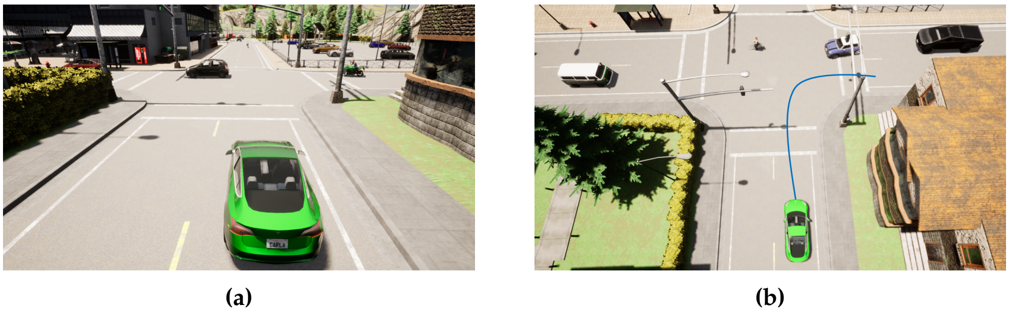
The simulator started by capturing images at a resolution of 800 × 600 pixels. We captured 5 frames per second (fps) during our simulation in order to reduce data overload. The episodes concluded under the following conditions: (1) the AV collides with any object or with any road participants; (2) the AV reaches its goal position; (3) the maximum number of episodes N_{steps} is reached.



(a) **(b)**

(c) **(d)**

Figure 3. (**a**) represents the traffic scenarios used to train and test our model, including various road users. (**b**) illustrates the predefined route of the AV from the starting location to the goal position. (**c**,**d**) present additional complex simulations that the AV must generate in order to navigate its environment effectively. Adapted from Ref. [22].



(a) **(b)**

Figure 4. (**a**) The scenario represents a right-turn intersection with the presence of other vehicles. (**b**) The AV must navigate the right-turn intersection while avoiding collisions.

5.3. ViT SAC Training

In our model, we employ two networks: a policy network and a critic network. The policy network uses a ViT as the feature extractor. It produces outputs corresponding to the mean and standard deviation of a Gaussian distribution through three fully connected layers with 258, 128, and 32 hidden units, respectively. This results in actions represented as $a_t \sim \mathcal{N}(\mu_\theta(s_t), \sigma_\theta(s_t))$. The critic network shares the same ViT architecture as the policy network. The other parameter of our training model are listed in Table 1.

We compared the SAC and ViT-SAC algorithms in terms of training time, processing time, memory consumption, and training speed over 600 episodes, as shown in Table 2. Notably, ViT-SAC demonstrates faster training and processing times compared to the SAC algorithm. However, the ViT-SAC algorithm consumes more memory than SAC.

Table 1. Parameters used in the simulation.

Notation	Meaning	Value
$v_{\max}$	Velocity limit (m/s)	30
γ	Discount rate	0.99
N_{buffer}	Agent buffer capacity	200,000
N_A	Mini-batch size for agent	32
N_E	Mini-batch size for expert	32
α	Learning rate for actor and critic	3×10^{-4}
$Block$	Number of ViT blocks	2
$Head$	Number of ViT heads	1
N_{steps}	Training steps	1100
$N_{patches}$	Patch size	14 × 14
lr	Learning rate of actor and critic	0.0003

Table 2. Computational efficiency.

Algorithm	Train (h)	Processing Time (s)	Memory Consumption (GB)
ViT-SAC	8.19	0.085	0.04832
SAC	10.82	0.22	0.0243

6. Analysis of Experiment Results

To demonstrate the efficacy of the proposed ViT-SAC model in a roundabout scenario, we compared our approach with the work of [22].

Figure 5 illustrates the average reward obtained by our ViT-SAC model compared to several RL baselines, including SAC, DQN, and PPO in roundabout scenario. Our model demonstrates faster convergence and superior performance throughout the episodes, achieving a high reward value of approximately 96.5, surpassing the other baselines. In contrast, SAC reaches a reward value of 91.9. Notably, the ViT-SAC model stabilizes from episode 1000 to the end, showcasing the efficiency of integrating a replay buffer with expert demonstrations. This integration accelerates the training process and significantly reduces the time required for convergence. Figure 6 illustrates the average reward of the ViT-SAC model compared to the SAC model in an intersection scenario. The ViT-SAC model achieves a significantly higher reward of approximately 96.5, outperforming the SAC model, which attains a reward of 75.9. This result highlights the superior performance of our model, even in the challenging task of navigating a right-turn intersection.

To evaluate the performance of our model, we employed the same metrics as in our previous work [22]. This allowed us to compare the effectiveness of ViT-SAC with other RL baselines, including SAC, DQN, and PPO.

We used the average success rate as the primary metric to fairly evaluate the performance of the proposed framework across different scenarios, each defined by the number of AV present in the environment. Specifically, we chose four density levels for our evaluation: $Veh = 120$, $Veh = 140$, $Veh = 160$, and $Veh = 210$.

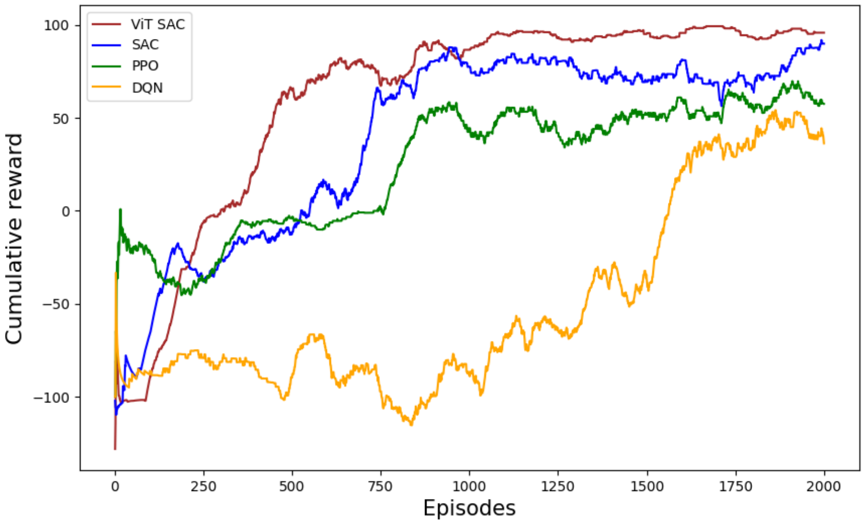


Figure 5. The mean reward of ViT-SAC compared to RL baselines in roundabout. Adapted from Ref. [22].

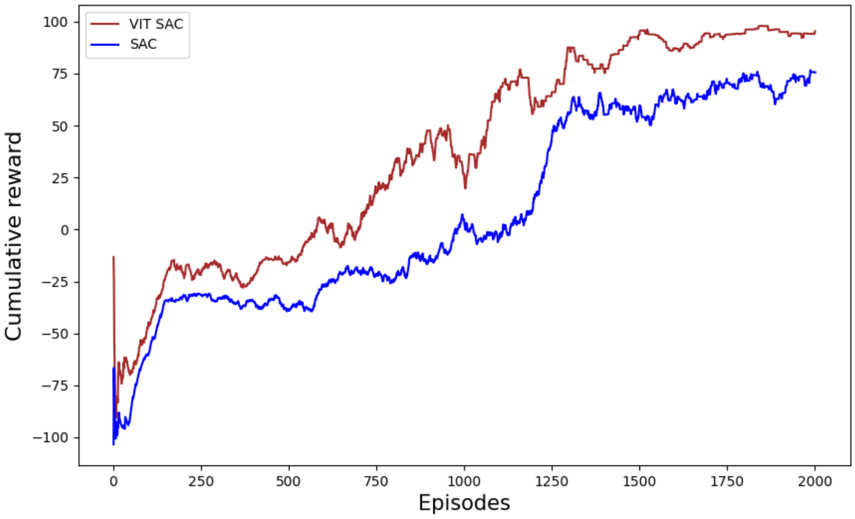


Figure 6. The mean reward of ViT-SAC compared to SAC in right-turn intersection.

In addition to the main metric, we incorporated several supplementary metrics during the testing phase, including collision rate, mean duration, mean velocity, mean traveled distance, mean steps, and mean reward:

- **Success Rate (Succ %)**: This metric measures the percentage of episodes in which the AV successfully reaches its destination out of 50 episodes tested.
- **Collision Rate (Coll %)**: This calculates the percentage of episodes in which the AV collides with other vehicles, reflecting critical safety performance.
- **Mean Duration (Mean Dur. (s))**: This measures the average time taken by the AV to reach its destination across 50 episodes.
- **Mean Velocity (Mean Vel.** m/s**)**: This reflects the average speed of the AV during the episodes.
- **Mean Traveled Distance (Mean. Dis** m**)**: This denotes the average distance traveled by the AV from its starting location to its destination.
- **Mean Steps (Mean Steps)**: This represents the average number of steps required for the AV to reach its destination.

Table 3 presents a quantitative comparison of the ViT SAC model against other RL algorithms utilizing CNN models. The results clearly demonstrate the superior performance of the ViT SAC model across all metrics. Notably, ViT SAC achieves a consistently high average reward across all traffic densities in the roundabout and right-turn intersection, even in the challenging fourth density characterized by high mobility.

Table 3. The results of the proposed algorithm are compared with the baseline methods in the roundabout and right-turn intersection. The good results in bold indicate superior performance.

Roundabout Scenario								
Model	**Dens.**	**Mean. Re**	**Mean Dur.**	**Mean Vel.**	**Mean. Dis**	**Mean. Steps**	**Succ.**	**Coll.**
ViT-SAC	$Veh = 120$	**100.00**	**5.95**	**4.21**	**31.40**	**25.74**	**100%**	**0**
	$Veh = 140$	**100.00**	**6.84**	**4.16**	**31.26**	**25.92**	**100%**	**0**
	$Veh = 160$	**100.00**	**6.95**	**4.18**	**31.30**	**25.84**	**100%**	**0**
	$Veh = 210$	**100.00**	**6.73**	**4.19**	**31.53**	**25.94**	**100%**	**0**
SAC [22]	$Veh = 120$	93.01	6.70	3.77	31.58	31.64	98%	2
	$Veh = 140$	92.99	7.48	3.78	31.70	32.91	98%	2
	$Veh = 160$	91.79	8.50	3.64	31.72	35.06	96%	2
	$Veh = 210$	85.07	10.36	3.75	31.63	30.80	94%	6
PPO	$Veh = 120$	62.1	7.01	3.60	32.51	36.01	88%	12
	$Veh = 140$	60.8	8.02	3.68	32.63	37.20	86%	14
	$Veh = 160$	57.2	8.70	3.55	32.72	38.10	83%	17
	$Veh = 210$	50.1	10.5	3.70	32.51	36.20	80%	20
DQN	$Veh = 120$	35.74	7.89	3.58	33.48	47.13	70%	30
	$Veh = 140$	24.98	9.28	3.52	33.35	47.81	64%	36
	$Veh = 160$	29.40	8.63	3.45	33.57	49.0	66%	34
	$Veh = 210$	1.86	10.88	3.63	33.54	47.69	52%	48
Right-Turn Intersection								
Model	**Dens.**	**Mean. Re**	**Mean Dur.**	**Mean Vel.**	**Mean. Dis**	**Mean. Steps**	**Succ.**	**Coll.**
ViT-SAC	$Veh = 120$	**95.66**	**4.97**	**3.43**	**20.29**	**21.48**	**100%**	**0**
	$Veh = 140$	**94.20**	**5.92**	**3.42**	**20.32**	**22.63**	**100%**	**0**
	$Veh = 160$	**94.93**	**6.35**	**3.46**	**20.40**	**21.54**	**96%**	**4%**
	$Veh = 210$	**94.24**	**8.08**	**3.45**	**21.02**	**21.46**	**96%**	**4%**
SAC [22]	$Veh = 120$	**55.01**	**7.24**	**2.58**	**21.08**	**31.78**	**72%**	**14%**
	$Veh = 140$	**56.76**	**9.40**	**2.39**	**21.12**	**35.89**	**72%**	**14%**
	$Veh = 160$	**55.86**	**9.01**	**2.50**	**21.31**	**32.62**	**70%**	**30%**
	$Veh = 210$	**54.20**	**12.50**	**2.57**	**22.01**	**32.04**	**68%**	**32%**

Moreover, the success rate of the ViT SAC model is outstanding, maintaining a 100% success rate across all densities, with a collision rate of zero in the roundabout. This highlights the efficiency and robustness of the proposed model, outperforming other algorithms such as SAC, PPO, and DQN. For right-turn intersections, ViT-SAC achieves a 100% success rate and a collision rate of 0% at lower densities. At higher densities, it maintains a 96% success rate with a 4% collision rate, outperforming SAC, which achieves success rates of 72% and 68%, along with collision rates of 14%, 30%, and 32% at corresponding density levels.

Additionally, the ViT SAC model exhibits exceptional performance in all four test scenarios, indicating its resilience to variations in traffic density. Consequently, the ViT SAC model is well suited for real-world situations where traffic density can fluctuate, ensuring that the autonomous vehicle can maintain safety in varying conditions.

7. Conclusions

This paper introduces a novel approach that combines IL and DRL to enhance vehicle control in AVs navigating roundabout and right-turn intersection scenarios. Specifically, we integrated an expert demonstration buffer into the SAC algorithm to enable the model to learn driving skills from a human expert. Furthermore, we utilize a ViT model instead of a CNN to extract key features from visual input, ensuring the retention of long-range information during driving. These features are then combined with a goal vector to guide the AV toward its destination. We trained and tested our model using the CARLA simulator to demonstrate its performance in an environment closely resembling the real world. The results highlight the effectiveness of our proposed approach, achieving a 100% success rate with zero collisions in the roundabout scenario, as well as in the first two density levels of the right-turn intersection. For the last two density levels of the right-turn intersection, our approach maintains a high performance with a 96% success rate and a 4% collision rate. These outcomes surpass the performance of baseline RL algorithms, including SAC, PPO, and DQN.

For future work, we aim to train and test our model in other scenarios, such as left-turn intersections with dense traffic or other complex scenario.

Author Contributions: Conceptualization and methodology, B.B.E., N.B., M.B., S.K. and N.M.; investigation, N.B. and M.B.; writing—original draft preparation, B.B.E.; supervision, N.B., M.B., S.K. and N.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author(s).

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence;
RL	Reinforcement Learning;
DL	Deep Learning;
MDP	Markov Decision Process;
CNN	Convolutional Neural Network;
AV	Autonomous Vehicle;
ViT	Vision Transformer;
DRL	Deep Reinforcement Learning;
DQN	Deep Q-Network;
DDPG	Deep Deterministic Policy Gradient;
PPO	Proximal Policy Optimization;
SAC	Soft Actor–Critic;
NLP	Natural Language Processing;
SA	Self-Attention;
IL	Imitation Learning.

References

1. Elallid, B.B.; Benamar, N.; Hafid, A.S.; Rachidi, T.; Mrani, N. A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving. *J. King Saud-Univ.-Comput. Inf. Sci.* **2022**, *34*, 7366–7390. [CrossRef]
2. Van Brummelen, J.; O'brien, M.; Gruyer, D.; Najjaran, H. Autonomous vehicle perception: The technology of today and tomorrow. *Transp. Res. Part Emerg. Technol.* **2018**, *89*, 384–406. [CrossRef]
3. El Hamdani, S.; Benamar, N. A comprehensive study of intelligent transportation system architectures for road congestion avoidance. In Proceedings of the Ubiquitous Networking: Third International Symposium, UNet 2017, Casablanca, Morocco, 9–12 May 2017; Revised Selected Papers 3; Springer: Berlin/Heidelberg, Germany, 2017; pp. 95–106.
4. Qureshi, K.N.; Abdullah, A.H. A survey on intelligent transportation systems. *Middle-East J. Sci. Res.* **2013**, *15*, 629–642.
5. Zhao, X.; Fang, Y.; Min, H.; Wu, X.; Wang, W.; Teixeira, R. Potential sources of sensor data anomalies for autonomous vehicles: An overview from road vehicle safety perspective. *Expert Syst. Appl.* **2024**, *236*, 121358. [CrossRef]
6. Fadhel, M.A.; Duhaim, A.M.; Saihood, A.; Sewify, A.; Al-Hamadani, M.N.; Albahri, A.; Alzubaidi, L.; Gupta, A.; Mirjalili, S.; Gu, Y. Comprehensive Systematic Review of Information Fusion Methods in Smart Cities and Urban Environments. *Inf. Fusion* **2024**, *107* , 102317. [CrossRef]

7. Le Mero, L.; Yi, D.; Dianati, M.; Mouzakitis, A. A survey on imitation learning techniques for end-to-end autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 14128–14147. [CrossRef]
8. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A.A.; Yogamani, S.; Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 4909–4926. [CrossRef]
9. Zhu, Z.; Zhao, H. A survey of deep RL and IL for autonomous driving policy learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 14043–14065. [CrossRef]
10. Zare, M.; Kebria, P.M.; Khosravi, A.; Nahavandi, S. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Trans. Cybern.* **2024**, *54*, 7173–7186. [CrossRef] [PubMed]
11. Fang, B.; Jia, S.; Guo, D.; Xu, M.; Wen, S.; Sun, F. Survey of imitation learning for robotic manipulation. *Int. J. Intell. Robot. Appl.* **2019**, *3*, 362–369. [CrossRef]
12. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
13. Huang, Y.; Chen, Y. Autonomous driving with deep learning: A survey of state-of-art technologies. *arXiv* **2020**, arXiv:2006.06091.
14. Vaswani, A. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
15. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
16. Li, X.; Ding, H.; Yuan, H.; Zhang, W.; Pang, J.; Cheng, G.; Chen, K.; Liu, Z.; Loy, C.C. Transformer-based visual segmentation: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 10138–10163. [CrossRef] [PubMed]
17. Huang, Z.; Wu, J.; Lv, C. Efficient deep reinforcement learning with imitative expert priors for autonomous driving. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 7391–7403. [CrossRef]
18. Pérez-Gil, Ó.; Barea, R.; López-Guillén, E.; Bergasa, L.M.; Gómez-Huélamo, C.; Gutiérrez, R.; Díaz-Díaz, A. Deep reinforcement learning based control for Autonomous Vehicles in CARLA. *Multimed. Tools Appl.* **2022**, *81*, 3553–3576. [CrossRef]
19. Elallid, B.B.; Bagaa, M.; Benamar, N.; Mrani, N. A reinforcement learning based approach for controlling autonomous vehicles in complex scenarios. In Proceedings of the 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, 19–23 June 2023; IEEE: New York, NY, USA, 2023; pp. 1358–1364.
20. Ben Elallid, B.; Bagaa, M.; Benamar, N.; Mrani, N. A reinforcement learning based autonomous vehicle control in diverse daytime and weather scenarios. *J. Intell. Transp. Syst.* **2024**, 1–14. [CrossRef]
21. Liu, J.; Cui, Y.; Duan, J.; Jiang, Z.; Pan, Z.; Xu, K.; Li, H. Reinforcement learning-based high-speed path following control for autonomous vehicles. *IEEE Trans. Veh. Technol.* **2024**, *73*, 7603–7615. [CrossRef]
22. Elallid, B.B.; Benamar, N.; Bagaa, M.; Hadjadj-Aoul, Y. Enhancing Autonomous Driving Navigation Using Soft Actor-Critic. *Future Internet* **2024**, *16*, 238. [CrossRef]
23. Kargar, E.; Kyrki, V. Vision transformer for learning driving policies in complex and dynamic environments. In Proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 5–9 June 2022; IEEE: New York, NY, USA, 2022; pp. 1558–1564.
24. Liu, H.; Huang, Z.; Mo, X.; Lv, C. Augmenting Reinforcement Learning with Transformer-based Scene Representation Learning for Decision-making of Autonomous Driving. *IEEE Trans. Intell. Veh.* **2024**, *9*, 4405–4421. [CrossRef]
25. Huang, W.; Zhou, Y.; He, X.; Lv, C. Goal-guided transformer-enabled reinforcement learning for efficient autonomous navigation. *IEEE Trans. Intell. Transp. Syst.* **2023**, *25*, 1832–1845. [CrossRef]
26. Li, Y. Deep reinforcement learning: An overview. *arXiv* **2017**, arXiv:1701.07274.
27. Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.; Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In Proceedings of the AAAI conference on artificial intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
28. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
29. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
30. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
31. Hua, J.; Zeng, L.; Li, G.; Ju, Z. Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. *Sensors* **2021**, *21*, 1278. [CrossRef]
32. Palanisamy, P. *Hands-On Intelligent Agents with OpenAI Gym: Your Guide to Developing AI Agents Using Deep Reinforcement Learning*; Packt Publishing: Birmingham, UK, 2018.
33. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 1–16.

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.