electronics

Article

Multiagent Manuvering with the Use of Reinforcement Learning

Mateusz Orłowski [1,2,*] and Paweł Skruch [1,2]

[1] Aptiv Services Poland S.A., ul. Podgórki Tynieckie 2, 30-399 Cracow, Poland
[2] Department of Automatic Control and Robotics, AGH University of Science and Technology, Adam Mickiewicz Avenue 30/B1, 30-059 Krakow, Poland
* Correspondence: mateusz.orlowski@aptiv.com

Abstract: This paper presents an approach for defining, solving, and implementing dynamic cooperative maneuver problems in autonomous driving applications. The formulation of these problems considers a set of cooperating cars as part of a multiagent system. A reinforcement learning technique is applied to find a suboptimal policy. The key role in the presented approach is a multiagent maneuvering environment that allows for the simulation of car-like agents within an obstacle-constrained space. Each of the agents is tasked with reaching an individual goal, defined as a specific location in space. The policy is determined during the reinforcement learning process to reach a predetermined goal position for each of the simulated cars. In the experiments, three road scenarios—zipper, bottleneck, and crossroads—were used. The trained policy has been successful in solving the cooperation problem in all scenarios and the positive effects of applying shared rewards between agents have been presented and studied. The results obtained in this work provide a window of opportunity for various automotive applications.

Keywords: autonomous vehicles; reinforcement learning multiagent reinforcement learning

Citation: Orłowski, M.; Skruch, P. Multiagent Manuvering with the Use of Reinforcement Learning. *Electronics* **2023**, *12*, 1894. https://doi.org/10.3390/electronics12081894

Academic Editor: José Santa

Received: 9 March 2023
Revised: 12 April 2023
Accepted: 14 April 2023
Published: 17 April 2023

1. Introduction

Autonomous driving (AD) is one of the most significant moves toward introducing truly intelligent systems to our daily lives. With years of development, cars have begun to become more and more autonomous, increasing both the safety and comfort of traveling. Inherently, driving a car means cooperating with other road users. In most cases, priority in such interaction is codified by law and traffic signs or might be realized as a pure response to others' actions (such as in the Adaptive Cruise Control (ACC) case, when the following car controls its speed based on the speed of the lead car). However, there are situations where the cooperation of the agents (cars) is vital to effectively navigate through a given scenario. Those scenarios may lack clearly defined right-of-way, or blindly following imposed prioritization without responding to other agents' needs will yield suboptimal solutions. In those cases, each of the agents has to come up with a strategy that is consistent with the strategies of others. If such an alignment is not found, either some group of agents will not be able to achieve its goals (busy road with nobody letting merging cars in) or all agents will be in a deadlock (two cars in a bottleneck section driving from opposite directions, when neither or both of them decide to pass first).

At the same time, reinforcement learning has proven to be an attractive way of solving complex problems, both in classic single-agent environments [1–3], as well as in multiagent cases [4–7].

This work focuses on multi-agent reinforcement learning for challenging scenarios that require high cooperation from agents. In order to train and later evaluate the reinforcement learning policy, a multi-agent maneuvering environment is being introduced in which each of the simulated cars aims at reaching a predefined goal position. With this baseline, we codify a set of scenarios that resemble real-world problems in which cooperation between multiple road users is a necessity.

Since part of the simulated scenarios is quite novel and specific to this problem formulation, comparison to other methods would not be informative (like in the case of global planning methods) or require intensive implementation efforts (e.g., the system for simultaneous prediction and planning). Existing methods should be considered rather different from the one presented in the paper and cannot be directly compared. Because of that, we focus on comparing different customizations within our research and refer those to the baseline acquired by the simplest version of the system.

By combining a relatively straightforward training approach, policy design, environment dynamics, and carefully crafted scenarios, we achieved satisfactory performance. The resulting policy does not rely on any communication with other road users or infrastructure but operates solely on a scene and other agents' perceptions. Therefore, we state that individual agents' locally derived actions lead to the globally efficient strategy of solving one of the hardest scenarios from a decision-making perspective in the AD domain. In the second part of the research, we also prove that introducing a shared-reward mechanism, adopted from but not connected to autonomous driving research, improves training efficiency and results in better performance in congested traffic scenarios. It also enables a simple way of addressing other agents' objectives in policy shaping, without the need to know them during execution time explicitly. We suggest that the proposed approach is well-suited to the planning system that could be used in multiple autonomous driving applications that involve planning and cooperating with other road users.

2. Related Work

The primary focus of the current study is the complexity and interactive character of dynamic maneuvering in traffic, which has already been studied previously. The authors of [8] summarized the most common approaches and provided a taxonomy for strategy-determining methods for intersection scenarios. However, this division might be extended to non-intersection cases as well. The first class of methods involves cooperative driving strategies, in which planning is made in the central coordination unit (by V2I—vehicle-to-infrastructure communication) or is distributed among the cars that do communicate with each other (V2V—vehicle-to-vehicle communication). In the prior, planning and strategizing can be carried out on the level of individual agents' interactions [9,10] or by grouping them into platoons [11–13]. The biggest benefit of such an approach is the ability to define a globally optimal strategy by accessing all information and controlling all agents. In the case of distributed strategy, agents do communicate with each other to establish a suboptimal solution but do not have to use a centralized coordination unit. Cooperative game theory has been successfully used in that setting, aiming to tackle strategic interaction between agents [14–16]. Applications of model-predictive control to multi-vehicle traffic optimization methods have been explored as well [17,18].

While cooperative methods do offer obvious benefits, including the ability to define a globally optimal strategy with the centralized unit or directly signalizing intentions and aligning on interactions with the use of communication channels, those methods have serious limitations as well. Centralized cooperative methods require additional hardware to be mounted in the place of interest; therefore, they are limited to specific locations, while traffic negotiation might be needed in other places as well (imagine bottleneck scenario as an example). Additionally, all the above-mentioned methods heavily rely on communication and assume that all traffic participants are capable of such communication, which are impossible assumptions to prove with human-driven cars. Therefore, the second family of methods, concerning individual driving strategies, is the main point of interest of early adopters in the industry and research.

Much more like human drivers, individual driving strategies only rely on world perception and are able to make appropriate decisions in a standalone fashion, aiming to be consistent with other road users' strategies. Although those methods lose access to perfect information and, by definition, struggle to provide a globally optimal strategy, they support the coordination of non-automated traffic participants and put no hard

requirements on wireless communication. Those methods need to involve extended scene perception, including complex road structure representation and object-to-lane assignment, as well as intention and trajectory prediction, which are often executed simultaneously with the planning algorithm. The most classical methods for planning involve the utilization of Finite State Machines (FSMs) that divide the state of the vehicle into a discrete set of categories, which refers to the specific control strategy and arbitrates transitions between those states. Successful use of Hierarchical State Machines, extending the FSM concept by introducing more layers of classes, has been successfully used in the first push toward automated vehicles coordination, the DARPA Urban challenge [19,20]. However, state machine methods are only suitable for simple cases, as artificially created rules and states struggle to cover the vastness of all possible scenarios [21].

While addressing the issue of how to cover multiple scenarios, the most obvious choice is data-driven methods. In [22], agent control in urban scenarios has been realized by imitation learning with the augmentation of expert data samples. The neural-network-based models become state-of-the-art methods for behavior and trajectory prediction tasks, which are an essential part of planning motion in dynamic environments [23–25]. As trajectory planning can be characterized as a closed-loop control problem, reinforcement learning has been successfully used in the autonomous driving domain as well [26–29].

The extension of reinforcement learning to the multi-agent setting is a very active research field, which, at the same time, has been proven to be a very challenging one. Optimizing agents' policies with the use of machine learning by utilizing communication channels between agents was investigated in [30,31]. The application of Monte Carlo Tree Search methods paired with reinforcement learning was used in [4,32], where agents competed in games of Go, chess, and Shogi with a self-play mechanism, resulting in a state-of-the-art performance in each of those games. In [6], an extensively scaled-up version of the Proximal Policy Optimization algorithm [2] was used to play a multi-character strategy game, Dota 2. OpenAI Five introduced a hyperparameter called team spirit, which is responsible for weighting individual characters' rewards versus the average rewards of the team members. At the same time, training followed a self-play approach, where agents competed with a pool of old versions of themselves. In [7], two teams played hide-and-seek. The main discovery of the training was the emergence of a series of strategies and counter-strategies by both teams as the training progressed, creating complex tasks from relatively simple game dynamics. A method utilizing the actor–critic approach, in which the critic had access to all agents' observations during training, was proposed in [33]. As the actor network only consisted of local observation, the method did not require any communication between agents during the inference. An interesting observation was made in [34], where the issue of cooperation with the suboptimal human agent was highlighted. The authors showed that in cooperative environments, agents trained through self-play paired with human agents presented significantly worse performance and proposed countermeasures, resulting in better coordination and more robust policies.

In recent years, there have been many applications of multiagent reinforcement learning (MARL) to different autonomous driving problems. In [35], the author introduced a platform that allowed highly customizable learning of multiagent autonomous driving systems and, with it, trained agents to navigate in a partially observable, stop-sign-controlled, three-way urban intersection environment with raw (camera) sensor observations. The merging onto the highway scenario was studied in [36], introducing high-level safety supervisors that relied on communication between vehicles and infrastructure (V2I). The problem of cooperation with human drivers was studied in [37,38], where the introduction of a notion of altruism in the reward function improved traffic flow and safety. Using the semantic action space, the automotive supplier MobilEye used reinforcement learning to plan autonomous car behavior on both highways and in urban scenarios [39,40]. The introduction of MARL-based adversarial agents, which aims to find the potential failure modes of a given autonomous driving system, took place in [41]. The application of MARL methods to adaptive traffic signal control (ATSC) was presented in [42], where the issue

of a massive action space for a large-scale centralized ATSC system was overcome by distributing control to individual local agents.

For further reference, the authors of [43] presented a survey of the application of multiagent reinforcement learning to autonomous driving problems.

3. Multiagent Maneuvering

The multiagent maneuvering environment allows simulating car-like agents within an obstacle-constrained space. Each of the agents is tasked with reaching an individual goal, which is defined as a specific location in space. Knowledge about the location of a goal is only provided to the agent, who is supposed to reach it, and explicit information is not provided to other agents. Additionally, the space is populated with obstacles in the form of polygons. Agent-to-agent and agent-to-obstacle collisions are detected and result in the termination of the episode for the involved agents and their removal from the scene. Figure 1 presents the environment–agent interaction process.

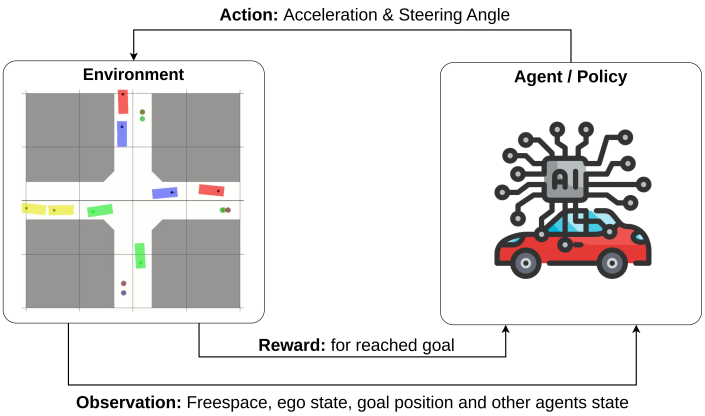

Figure 1. Scheme of agent–environment interaction used in our problem statement.

3.1. Environment Observation

To encode the state of the environment, we took inspiration from [7], where a similar multiagent setup was used for a game of hide-and-seek. First, the freespace around each of the agents was encoded with the use of distance measurements at different azimuths. We used 50 rays starting in the middle of the rear axis and pointing in uniformly distributed directions around the car (see Figure 2). To provide context information about the state of the ego itself, we provided information about the current speed, yaw rate, and relative position of the goal on the x and y axes. To inform each of the agents about the other agents, a list of encodings relating to each agent was added. Each such encoding included information about the relative position and the relative speed of the given agent, as well as a doubled ego observation for context. Additionally, we included a mask, allowing us to identify which of the other agents' embeddings were valid.

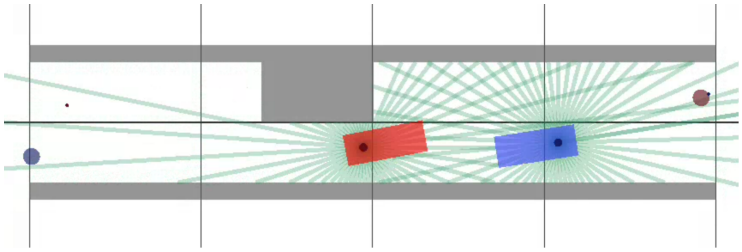

Figure 2. The multiagent maneuvering environment with an example scenario of a bottleneck and two cars trying to negotiate safe passage through it. The color-coded goals of the respective agents are shown in the form of circles, while the green rays are the visualization of the freespace observation.

3.2. Motion Modeling and Action Space

The agents were simulated as rectangular objects similar to cars. Their action space was a discrete set, created as a combination of acceleration and wheel angle values discretized, respectively.

As we aimed to simulate the interaction between maneuvering cars in low-speed scenarios, we had to simulate and track their dynamic state, assuming that it would be one of the deciding factors in maneuver selection and negotiation with others. For this purpose, we implemented a standard bicycle model, allowing us to control each car with acceleration and steering commands.

The state of the motion model consisted of the position of the center of the rear axle in the global coordinate system, x, y, with orientation angle ψ. To track the dynamic state of the car, the state also consisted of v, a variable that represented the magnitude of the velocity and the yaw rate ω. The angle of slipping was not modeled. The control was defined as a, representing the acceleration command, and the angle of the wheel δ. The model was parameterized with the wheelbase (distance between axles) of the car, L. The Δt symbol represented the amount of time to be simulated. A graphical representation of the bicycle model is shown in Figure 3, while the state dynamics were represented by the following equations:

$$x_{k+1} = x_k + r(\sin(\psi_k + \omega_{k+1}\Delta t) - \sin\psi_k)\,, \tag{1}$$

$$y_{k+1} = y_k + r(\cos\psi_k - \cos(\psi_k + \omega_{k+1}\Delta t))\,, \tag{2}$$

$$\psi_{k+1} = \psi_k + \omega_{k+1}\Delta t\,, \tag{3}$$

$$v_{k+1} = v_k + a_k\Delta t\,, \tag{4}$$

$$\omega_{k+1} = \frac{v_k + \frac{a_k\Delta t}{2}}{r}\,, \tag{5}$$

where the turning radius, r, is derived as:

$$r = \frac{L}{\tan\delta_k}\,. \tag{6}$$

In the case of straight motion ($\delta = 0$) and without a proper definition of the turn radius r (as $r \to \infty$), the equations are simplified straightforwardly. Additionally, the velocity of the car was constrained to a predefined range, which had direct implications on the distance traveled in each simulation step and therefore on most of the model.

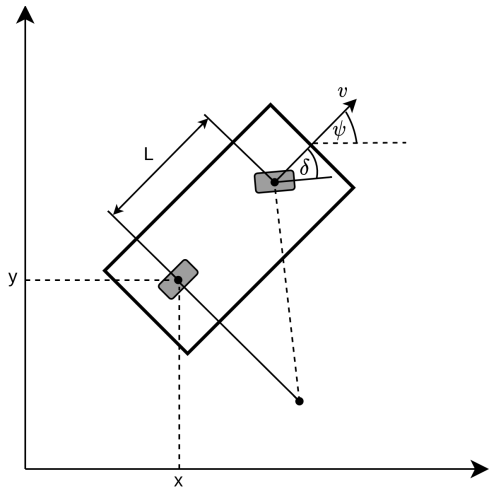

Figure 3. The graphical representation of the kinematic motion model used for the simulation.

3.3. Scenarios

The above elements defined the dynamics of the environment itself, along with the interface through which the agents interacted with it. Treating those parts as a frame, a concrete scenario was then defined as a specific combination of the agents' initial positions, their goals, and the configuration of obstacles.

Within our experiments, we aimed at simulating situations that would require agent cooperation and would introduce an element of competition.

Bottleneck

In the bottleneck scenario (Figure 4), a narrow part of the road prevented the agents from freely passing each other and required one agent to wait until the other passed. Two agents on opposite sides with goals on the other end of the road were simulated. The road was 40 m in length and 7 m in width, and the narrowing was 3.5 m wide. The simulated bottleneck section of the road was randomized both in terms of its location and dimensions, as well as its specific structure. Cases such as no bottleneck at all, a singular bottleneck on one side, a double bottleneck with freespace between obstacles, and a symmetric case with the free middle of the road were simulated.

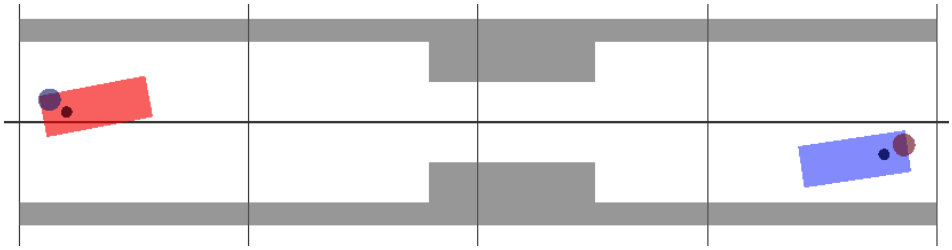

Figure 4. The bottleneck scenario with a centrally placed bottleneck.

Zipper

In the zipper scenario (Figure 5), agents must cooperate to merge from two lanes into one. Six agents were being simulated, in two lanes, respectively, with the singular location of the goal located within the narrowing, which was placed at one end of the road. The single-lane section was simulated on the left, center, and right sides of the road with the option of not narrowing.

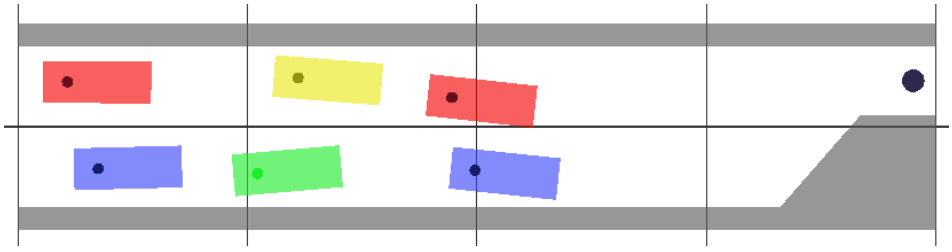

Figure 5. The zipper scenario with the narrowing located on the left side of the road. All the agents share the same goal marked as a dark blue dot on the right side of the image.

Crossroad

In the crossroad scenario (Figure 6), up to ten agents were spawned on four different connecting roads of the cruciform intersection. The goal of each agent was randomly selected at the end of the connecting road. No arbitration regarding priority on the road was introduced.

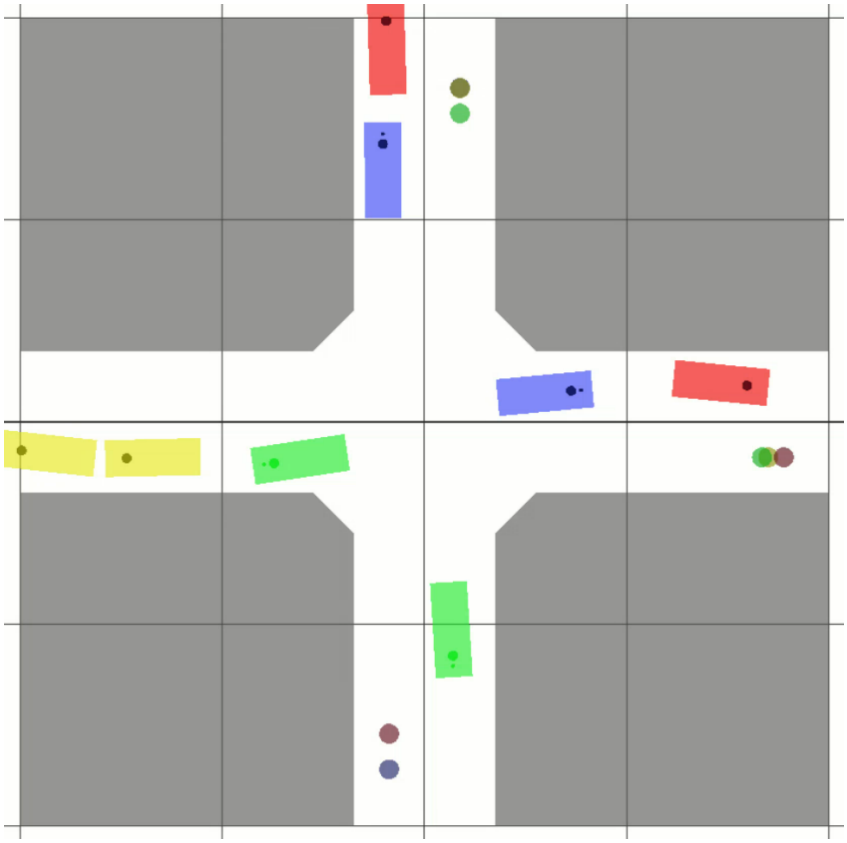

Figure 6. The crossroad scenario, with multiple agents each aiming at a different end goal, which is color-coded.

4. Policy Optimization

In all experiments, agents were trained with the use of a self-play, which improves the data efficiency and allows for strategy synchronization by the agents.

We follow the principle of decentralized execution for experience collection and centralized training. At execution time, every agent acts only on the basis of its local observation by querying the policy for action. The experiences of all agents are gathered throughout the episodes and placed in a centralized buffer, which are used to improve the policy by the trainer.

The policy is optimized with Proximal Policy Optimization [2] and General Advantage Estimation for value estimation [44], using the implementation of the RLlib library [45]. In the implementation of the experiments, we followed some of the tips proposed in [46], such as reward normalization, increasing the size of the batch, and reducing the number of policy optimization steps per batch. Training was executed on the local cluster with the SLRUM scheduling mechanism and in most cases utilized 50 rollout workers, each assigned with a single CPU thread, responsible for experience collection and a single GPU for optimization. Except for those settings, no special adaptation has been made to the PPO algorithm to address the multi-agent aspect of the studied problem. The specific parameters used by the training algorithms have been provided in Table 1.

All agents share the same policy, including the neural network architecture as well as the weights; nevertheless, they observe and act based only on local information. This allows the potential deployment of the policy without the need to implement any communication channels between agents.

Table 1. Parameters of the PPO algorithm used in all experiments.

train batch size	2,000,000
number of sgd iterations	6
gamma	0.995
lambda	0.95
kl coefficient	0.0
clipping parameter	0.1
gradient clipping	2.0
learning rate	5×10^{-5}

The neural network architecture was based on the one proposed in [7] and is presented in Figure 7. Freespace observation, represented as 50 rays encoding the distance to obstacles, is processed with the use of 1D Circular Convolution and later concatenated with the observation of the ego itself. The other agents' data are similarly embedded together with ego to corresponding embeddings. Defined in such a way that the entity embeddings are processed with the use of three masked multi-head attention layers, assuring that the eventual nonvalid objects are not a part of the processing. Later, embedding corresponding to the ego is selected and processed with fully connected layers to acquire discrete distribution over the actions (acceleration and steering angle combinations) and the value function estimation itself.

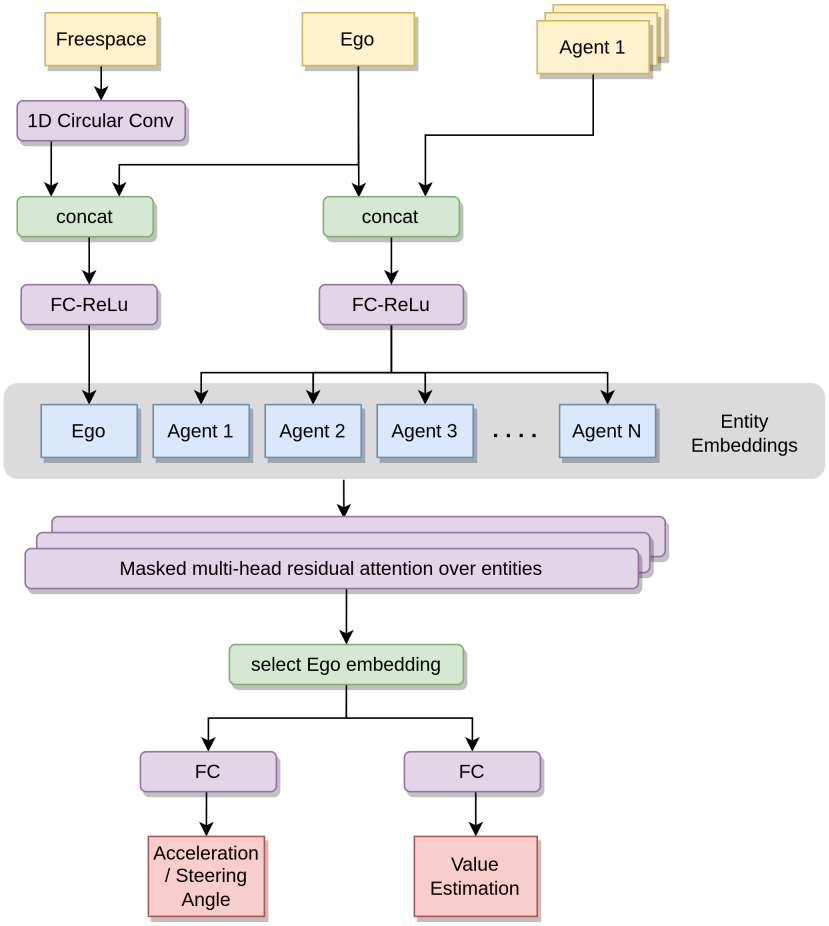

Figure 7. The architecture of the neural network. Inputs to the network are color-coded as yellow. Learnable parameters are presented as purple blocks, while math operations are green. Outputs are red.

5. Results

5.1. Egoistic Rewards Training Evaluation

The road scenarios did not have a well-defined beginning and end from an all-agent perspective, as the lifespan of each car in a given scenario might be different. At the same time, one may easily encode the specific task for each car, clearly detecting when the car fulfills a given personal objective (arrives at the destination). Because of that, we proposed a straightforward, purely egoistic, and sparse reward mechanism in which agents were rewarded individually based only on their internal objectives. Each agent received a +1 reward for reaching the goal and was not rewarded otherwise. With this definition, agents were also not directly encouraged to complete episodes as quickly as possible, except for the natural effect of the gamma parameter (discount factor), which did not equal 1.

We separately trained agents in all the mentioned scenarios (zipper, bottleneck, and crossroads) using a trained policy for all agents on the scene. The evolution of the example episodes can be seen in Figure 8.

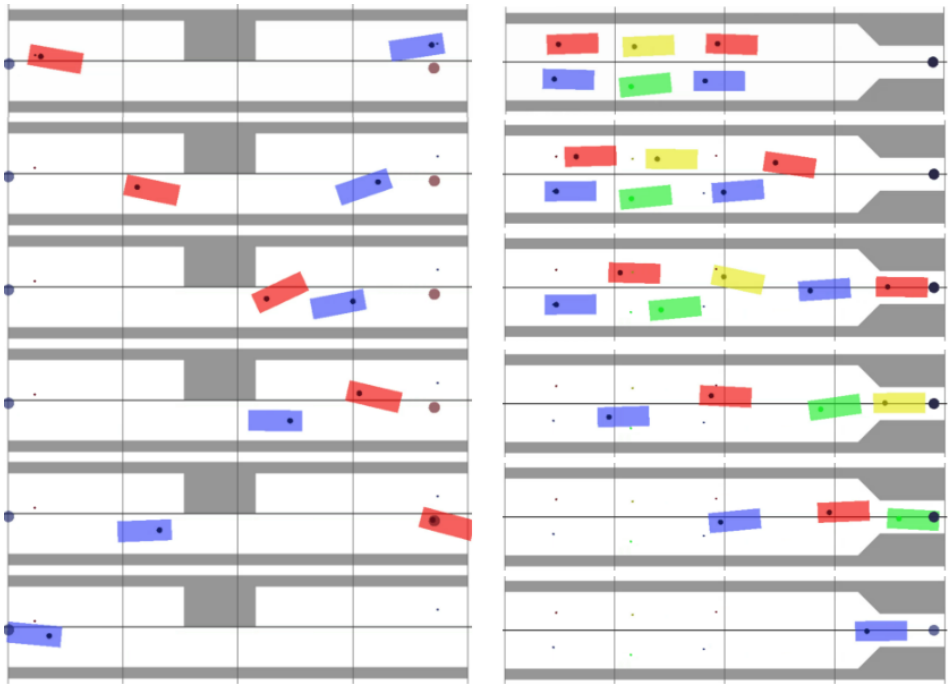

Figure 8. Evolution of the episodes for the bottleneck and zipper scenarios.

Interestingly, all the training resulted in the cooperative behavior of the agents and a good overall performance (see the *Baseline* data in Table 2 for the results of the *Crossroads* scenarios), despite the fact that agents were only rewarded for their own performance. After examining the potential causes of this situation, we arrived at the conclusion that the simulated scenarios were inherently cooperative and did not involve competitive traits. Looking at the training times, the *Bottleneck* scenario was the hardest problem to solve, probably because it's an easy deadlock situation, which is not easy to negotiate and requires a long planning and prediction horizon.

With the proposed scenarios and problem formulation, it was difficult to yield situations in which one of the agents made a move in their own favor or did not jeopardize its own performance by exploiting other agents' performance. There are multiple factors that contribute to this and might differentiate this setup from on-road scenarios. Firstly, both accidents and potential blockages are equally destructive to the performance of all agents involved. Second, the simulation lacks traffic rules that would impose drive-through prioritization; therefore, agents cannot be penalized for breaking them. Last but not least, the reward does not include the direct *faster-the-better* notion. That being said, the conclusion is that the goal for each individual agent is aligned with the good of all other agents.

Table 2. Performance evaluation of the three setups of crossroad environments. In the *Baseline*, the agents were rewarded purely for reaching their goals. In the environment *Timed*, the reward depended on the mean speed in the episode, while the last environment, *Timed with shared reward* recreated the reward mechanism but added the reward-sharing mechanism. The characteristics related to the duration, speed, and acceleration of the episode were calculated only for the agents who achieved the goal successfully.

	Baseline	**Timed**	**Timed with Reward Sharing**
Goal reached (%)	99.5	96.9	97.65
Obstacle collision (%)	0.12	0.43	0.24
Agent collision (%)	0.32	2.73	2.16
Avg episode length	31.08	23.33	22.86
Avg speed (m/s)	1.9	2.566	2.584
Max speed (m/s)	3.41	5.21	5.761
Min speed (m/s)	0.52	1.01	1.032
Static in episode (%)	13.23	6.14	6.34
Avg sum acc (m/s^2)	20.64	18.58	18.27
Std sum acc (m/s^2)	0.76	0.856	0.855

Understanding the limitations of the setup, including the lack of traffic rules, the single black-box policy controlling all agents, as well as the simplicity and narrow scope of the scenarios, it is vital to underline the good performance acquired by the agents in those challenging scenarios with a straightforwardly defined objective. The behavior acquired by the agents seemed to be quite realistic as well and resembled human behavior.

5.2. Introduction of Time Incentive and Reward Sharing

Realizing that the initially defined setup manifested highly cooperative traits, we wanted to increase competitiveness. To do so, we have introduced a reward based on time, which would promote arriving at the destination faster (see Equation (7)). Similar to the previous experiments, the reward was sparse and calculated only at the end of the episode and was nonzero only when the agent successfully arrived at the destination. In the case of success, the driving time (t_d) and length of reference route, d_{ref}, have been used to calculate the effective average velocity over the episode, V_e. The reference route in the crossroads scenarios has been straightforwardly defined as driving from the initial position toward the middle of the intersection, followed by driving from that center toward the goal position. To normalize the reward, we divided the average agent speed by a high bound for the average speed expected in such a scenario, V_{ref}, which we assumed to be 5 m/s. The definition of such a form allows keeping the rewards normalized and should promote high average velocities but only in cases when the agent drives effectively towards the destination.

$$r = \frac{V_e}{V_{ref}}, V_e = d_{ref}/t_d \tag{7}$$

To provide an incentive for cooperation among agents, we introduced the reward-sharing mechanism, similar to that of [6]:

$$r_i^f = (1-\tau)r_i + \tau\bar{r}\,, \tag{8}$$

where team spirit τ was used to weigh the reward of the individual agents r_i with the average reward of the team $\bar{r}$ and to calculate the final reward of the agent r_i^f. As the agents were not necessarily ending the episode at the same time, to be able to calculate the shared reward, we delayed the publication of their termination until all the agents terminated. In addition, we treated all agents as belonging to a single team.

Experiments with time-based rewards were performed in crossroad scenarios. The main difference in the conducted experiments is summarized in Table 3. We compared the agent's performance with and without the reward-sharing mechanism, also referring to

baseline training (Section 5.1). For the reward-sharing training, we used team spirit $\tau = 0.5$. The reward progress is presented in Figure 9, suggesting that reward sharing improved the end performance.

Table 3. Main configuration parameters of experiments conducted in crossroad scenarios. In *Baseline* and *Timed* experiments, team spirit is effectively set to 0.

	Reward Concept	**Team Spirit Value**
Baseline	1.0 for reaching reward, 0 otherwise	0.0
Timed	according to Equation (7)	0.0
Timed with reward sharing	according to Equations (7) and (8)	0.5

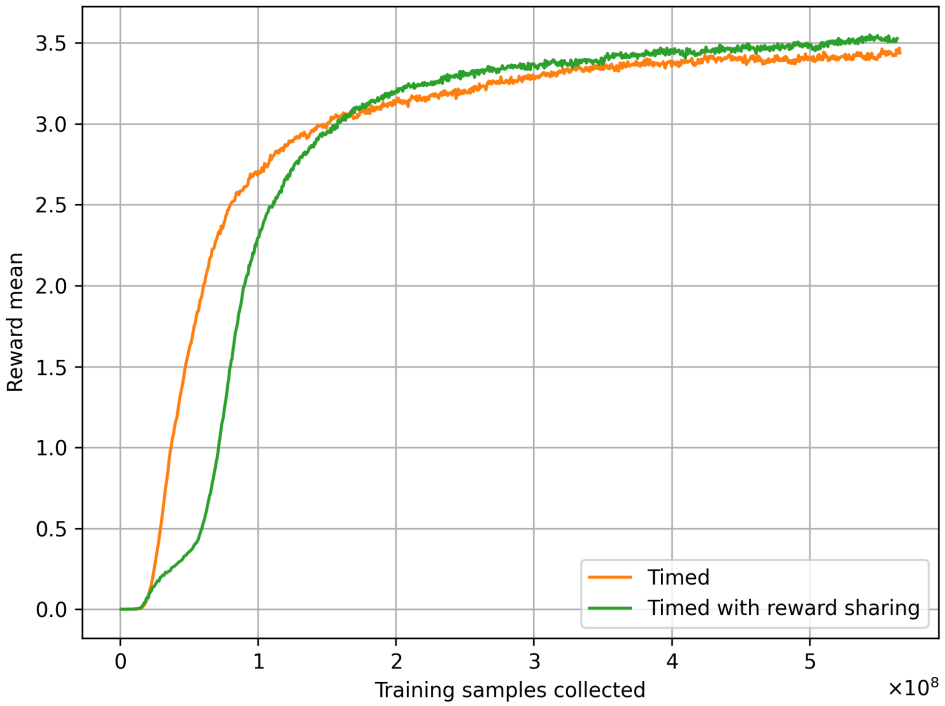

Figure 9. Graph presenting the training progress regarding the mean reward achieved in each training iteration. The reward-sharing mechanism slowed down the training progress in the beginning but resulted in a better performance in the end.

In detail, we performed an evaluation for each of the policies. We evaluated the policies on 10,000 episodes, which resulted in around 56,000 agent trajectories. The evaluation results are shown in Table 2. As expected, *Baseline training* showed a much better performance in achieving the goal and resulted in fewer collisions. At the same time, the *Timed* and *Timed with reward sharing* policies presented much higher average velocities and smooth behavior, with an advantage over the reward-sharing policy (see Figure 10 for histogram velocity). We also evaluated the average velocities and the probability of achieving the target aggregated by the number of agents in the episode (Figure 11). This evaluation showed that the reward-sharing mechanism introduced the most benefits with respect to the mean speed in high-density scenarios, while in less crowded scenarios, it introduced no benefit or even hindered the performance. With respect to the goal-reaching performance, reward sharing helped in all the cases, most probably because of the multiplied effect of collisions.

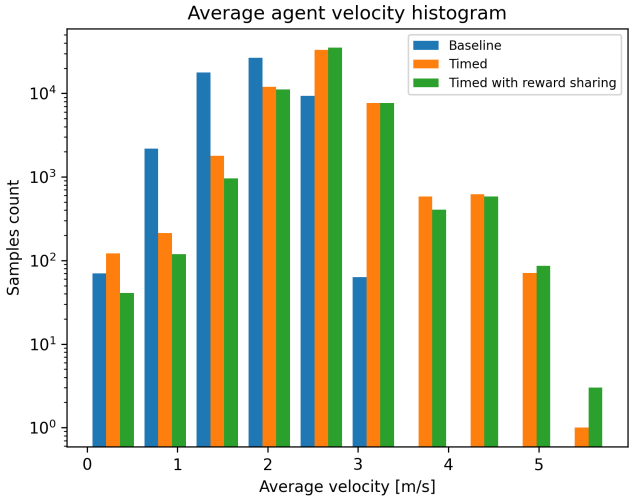

Figure 10. Histogram of the average velocities acquired by the agents (the horizontal axis is logarithmic). The reward-sharing mechanism (green) allowed acquiring many more samples with the highest velocities.

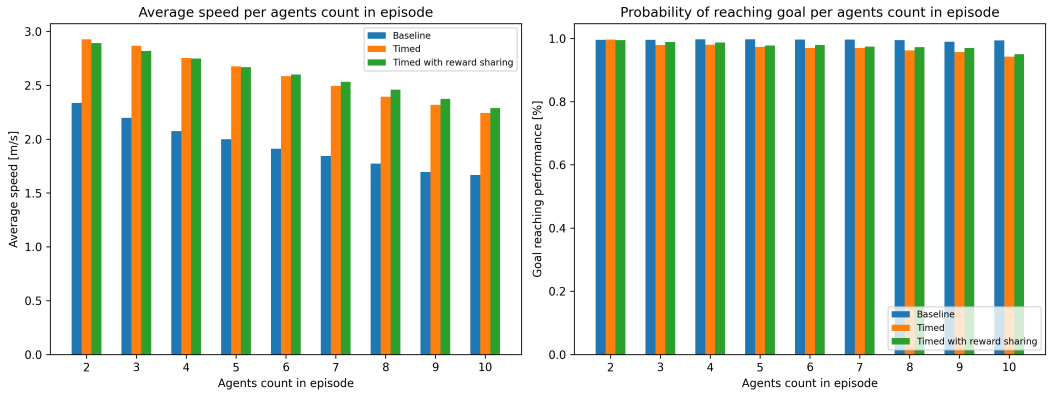

Figure 11. The average speed of the agents (**left**) and the goal reaching performance (**right**) as a function of the number of agents simulated in a given scenario. The obvious conclusion is that the average velocity drops as more cars are present in the scenario. Interestingly, the reward-sharing mechanism (green plot) improves the average velocities in dense scenarios (agent count in the scenario above 5) and slightly reduces it in cases where the number of agents is smaller (with respect to the reward mechanism being turned off). Goal reaching is improved by reward sharing independent of the number of agents; however, the baseline policy (blue) was superior in all cases. Those results suggest that reward sharing plays an important role in environments with denser traffic.

6. Discussion and Further Work

In this work, we demonstrated that with a straightforward problem formulation, it was possible to successfully solve simulated road scenarios involving large amounts of cooperation. Furthermore, simulated road scenarios with implemented reward mechanisms showed that this type of problem formulation yielded highly cooperative scenarios. As collisions have a detrimental effect on all traffic participants, both those at fault and the victims, all agents prioritized avoiding collisions. The acquired behaviors also seemed quite human-like, even though no direct mechanism was used to control them in such a way.

In the timed experiments, we were able to show that the reward-sharing mechanism improved cooperation between agents, yielding better individual results. This effect was especially visible in crowded scenarios, where the coordination of multiple agents played an important role. At the same time, the balance between speed and safety (analyzed in this research as not causing collisions, which is only part of safety considerations) was clearly visible and needs to be carefully addressed.

The possible extensions of this research are numerous. First, the grid search for parameters such as team spirit, gamma, and reward details could be performed with the aim of finding the one that meets the needs of the target. Extending training to a wider and more diverse set of on-road scenarios could result in more robust policies, especially when trying to fit a single policy for all the scenarios. Since agents have been trained with a self-play mechanism in a highly cooperative setting of on-road driving, the strategy of each actor assumes predefined behavior patterns from other agents in the scene and does not assume any out-of-distribution situations. With that in mind, analyzing the policy's robustness and providing countermeasures in the presence of actors not controlled by the same policy would be especially important extensions, taking into consideration the highly cooperative characteristics of on-road driving. Finally, the integration of rule-based constraints or handwritten heuristics into planning, which could encapsulate generally understood traffic rules, would be the most important and simultaneously the most challenging extension of this research, moving current policy closer to production applicability. At the same time, multiagent reinforcement-learning-based methods might be considered a solid element in the development of driving policies for highly automated vehicles.

Author Contributions: Conceptualization, M.O.; methodology, M.O.; software, M.O.; validation, M.O.; formal analysis, M.O. and P.S.; investigation, M.O.; resources, P.S.; data curation, M.O.; writing—original draft preparation, M.O.; writing—review and editing, M.O. and P.S.; visualization, M.O.; supervision, P.S.; project administration, P.S.; funding acquisition, P.S. All authors have read and agreed to the published version of the manuscript.

Funding: Industrial PhD carried out at the AGH University of Science and Technology realized in cooperation with Aptiv Services Poland S.A.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-learning. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016, Phoenix, AZ, USA, 12–17 February 2016; pp. 2094–2100. [CrossRef]
2. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Openai, O.K. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
3. Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. Soft Actor-Critic Algorithms and Applications. *arXiv* **2018**, arXiv:1812.05905.
4. Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv* **2017**, arXiv:1712.01815.
5. Kurach, K.; Raichuk, A.; Stańczyk, P.; Zając, M.; Bachem, O.; Espeholt, L.; Riquelme, C.; Vincent, D.; Michalski, M.; Bousquet, O.; et al. Google Research Football: A Novel Reinforcement Learning Environment. In Proceedings of the AAAI 2020—34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 4501–4510. [CrossRef]
6. Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Dębiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. Dota 2 with Large Scale Deep Reinforcement Learning. *arXiv* **2019**, arXiv:1912.06680.
7. Baker, B.; Kanitscheider, I.; Markov, T.; Wu, Y.; Powell, G.; McGrew, B.; Mordatch, I.; Brain, G. Emergent Tool Use From Multi-Agent Autocurricula. *arXiv* **2019**, arXiv:1909.07528.
8. Wei, L.; Li, Z.; Gong, J.; Gong, C.; Li, J. Autonomous Driving Strategies at Intersections: Scenarios, State-of-the-Art, and Future Outlooks. In Proceedings of the IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, Indianapolis, IN, USA, 19–22 September 2021; pp. 44–51. [CrossRef]
9. Wu, J.; Perronnet, F.; Abbas-Turki, A. Cooperative vehicle-actuator system: A sequencebased framework of cooperative intersections management. *IET Intell. Transp. Syst.* **2014**, *8*, 352–360. [CrossRef]
10. Xu, H.; Zhang, Y.; Li, L.; Li, W. Cooperative Driving at Unsignalized Intersections Using Tree Search. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 4563–4571. [CrossRef]
11. Zhang, Y.J.; Malikopoulos, A.A.; Cassandras, C.G. Optimal Control and Coordination of Connected and Automated Vehicles at Urban Traffic Intersections. In Proceedings of the American Control Conference, Boston, MA, USA, 6–8 July 2016; pp. 6227–6232. [CrossRef]

12. Hu, H.C.; Smith, S.F.; Goldstein, R. Cooperative Schedule-Driven Intersection Control with Connected and Autonomous Vehicles. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 1668–1673. [CrossRef]
13. Li, B.; Zhang, Y.; Zhang, Y.; Jia, N.; Ge, Y. Near-Optimal Online Motion Planning of Connected and Automated Vehicles at a Signal-Free and Lane-Free Intersection. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1432–1437. [CrossRef]
14. Li, N.; Yao, Y.; Kolmanovsky, I.; Atkins, E.; Girard, A.R. Game-Theoretic Modeling of Multi-Vehicle Interactions at Uncontrolled Intersections. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1428–1442. [CrossRef]
15. Chen, X.; Sun, Y.; Ou, Y.; Zheng, X.; Wang, Z.; Li, M. A conflict decision model based on game theory for intelligent vehicles at urban unsignalized intersections. *IEEE Access* **2020**, *8*, 189546–189555. [CrossRef]
16. Hang, P.; Lv, C.; Huang, C.; Xing, Y.; Hu, Z. Cooperative Decision Making of Connected Automated Vehicles at Multi-Lane Merging Zone: A Coalitional Game Approach. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 3829–3841. [CrossRef]
17. Katriniok, A.; Kleibaum, P.; Joševski, M. Distributed Model Predictive Control for Intersection Automation Using a Parallelized Optimization Approach. *IFAC-PapersOnLine* **2017**, *50*, 5940–5946. [CrossRef]
18. Morales Medina, A.I.; Van De Wouw, N.; Nijmeijer, H. Cooperative Intersection Control Based on Virtual Platooning. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 1727–1740. [CrossRef]
19. Fu, L.; Yazici, A.; Ozgüner, U. Route planning for OSU-ACT autonomous vehicle in DARPA Urban Challenge. In Proceedings of the IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 781–786. [CrossRef]
20. Montemerlo, M.; Becker, J.; Bhat, S.; Dahlkamp, H.; Dolgov, D.; Ettinger, S.; Haehnel, D.; Hilden, T.; Hoffmann, G.; Huhnke, B.; et al. Junior: The stanford entry in the urban challenge. *Springer Tracts Adv. Robot.* **2009**, *56*, 91–123. [CrossRef]
21. Lin, X.; Zhang, J.; Shang, J.; Wang, Y.; Yu, H.; Zhang, X. Decision Making through Occluded Intersections for Autonomous Driving. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference, ITSC, Auckland, New Zealand, 27–30 October 2019; pp. 2449–2455. [CrossRef]
22. Bansal, M.; Krizhevsky, A.; Ogale, A. ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst. *arXiv* **2018**, arXiv:1812.03079.
23. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. Nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11618–11628. [CrossRef]
24. Phan-Minh, T.; Grigore, E.C.; Boulton, F.A.; Beijbom, O.; Wolff, E.M. CoverNet: Multimodal Behavior Prediction Using Trajectory Sets. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 14062–14071. [CrossRef]
25. Schafer, M.; Zhao, K.; Buhren, M.; Kummert, A. Context-Aware Scene Prediction Network (CASPNet). In Proceedings of the IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, Macau, China, 8–12 October 2022; pp. 3970–3977. [CrossRef]
26. Chen, J.; Yuan, B.; Tomizuka, M. Model-free Deep Reinforcement Learning for Urban Autonomous Driving. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 2765–2771. [CrossRef]
27. Wang, S.; Jia, D.; Weng, X. Deep Reinforcement Learning for Autonomous Driving. *arXiv* **2018**, arXiv:1811.11329.
28. Li, C.; Czarnecki, K. Urban Driving with Multi-Objective Deep Reinforcement Learning. *arXiv* **2018**, arXiv:1811.08586.
29. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Sallab, A.A.; Yogamani, S.; Perez, P. Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 4909–4926. [CrossRef]
30. Mordatch, I.; Abbeel, P. Emergence of Grounded Compositional Language in Multi-Agent Populations. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, New Orleans, LA, USA, 2–7 February 2018; pp. 1495–1502. [CrossRef]
31. Sukhbaatar, S.; Szlam, A.; Fergus, R. Learning Multiagent Communication with Backpropagation. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 2252–2260. [CrossRef]
32. Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature* **2020**, *588*, 604–609. [CrossRef]
33. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Proceedings of the Advances in Neural Information Processing Systems, December 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 6380–6391. [CrossRef]
34. Carroll, M.; Shah, R.; Ho, M.K.; Griffiths, T.L.; Seshia, S.A.; Abbeel, P.; Dragan, A. On the Utility of Learning about Humans for Human-AI Coordination. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019. [CrossRef]
35. Palanisamy, P. Multi-Agent Connected Autonomous Driving using Deep Reinforcement Learning. In Proceedings of the International Joint Conference on Neural Networks, Budapest, Hungary, 14–19 July 2019. [CrossRef]
36. Chen, D.; Hajidavalloo, M.R.; Li, Z.; Chen, K.; Wang, Y.; Jiang, L.; Wang, Y. Deep Multi-agent Reinforcement Learning for Highway On-Ramp Merging in Mixed Traffic. *arXiv* **2021**, arXiv:2105.05701.

37. Toghi, B.; Valiente, R.; Sadigh, D.; Pedarsani, R.; Fallah, Y.P. Cooperative Autonomous Vehicles that Sympathize with Human Drivers. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 Septembe–1 October 2021; pp. 4517–4524. [CrossRef]
38. Toghi, B.; Valiente, R.; Sadigh, D.; Pedarsani, R.; Fallah, Y.P. Social Coordination and Altruism in Autonomous Driving. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 24791–24804. [CrossRef]
39. Shalev-Shwartz, S.; Shammah, S.; Shashua, A. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. *arXiv* **2016** arXiv:1610.03295.
40. Shalev-Shwartz, S.; Shammah, S.; Shashua, A. On a Formal Model of Safe and Scalable Self-driving Cars. *arXiv* **2017**, arXiv:1708.06374.
41. Wachi, A. Failure-scenario maker for rule-based agent using multi-agent adversarial reinforcement learning and its application to autonomous driving. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 6006–6012. [CrossRef]
42. Chu, T.; Wang, J.; Codeca, L.; Li, Z. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1086–1095. [CrossRef]
43. Dinneweth, J.; Boubezoul, A.; Mandiau, R.; Espié, S. Multi-agent reinforcement learning for autonomous vehicles: A survey. *Auton. Intell. Syst.* **2022**, *2*, 27. [CrossRef]
44. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.I.; Abbeel, P. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016—Conference Track Proceedings, Juan, PR, USA, 2–4 May 2016. [CrossRef]
45. Liang, E.; Liaw, R.; Moritz, P.; Nishihara, R.; Fox, R.; Goldberg, K.; Gonzalez, J.E.; Jordan, M.I.; Stoica, I. RLlib: Abstractions for Distributed Reinforcement Learning. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, 10–15 July 2018; Volume 7, pp. 4768–4780. [CrossRef]
46. Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games. *arXiv* **2021**, arXiv:2103.01955.