Exploring Applications of Deep Reinforcement Learning for Real-world Autonomous Driving Systems

Victor Talpaert[1,2], Ibrahim Sobh[3], B. Ravi Kiran[2], Patrick Mannion[4],
Senthil Yogamani[5], Ahmad El-Sallab[3] and Patrick Perez[6]

[1]*U2IS, ENSTA ParisTech, Palaiseau, France*
[2]*AKKA Technologies, Guyancourt, France*
[3]*Valeo Egypt, Cairo*
[4]*Galway-Mayo Institute of Technology, Ireland*
[5]*Valeo Vision Systems, Ireland*
[6]*Valeo.ai, France*

Keywords: Autonomous Driving, Deep Reinforcement Learning, Visual Perception.

Abstract: Deep Reinforcement Learning (DRL) has become increasingly powerful in recent years, with notable achievements such as Deepmind's AlphaGo. It has been successfully deployed in commercial vehicles like Mobileye's path planning system. However, a vast majority of work on DRL is focused on toy examples in controlled synthetic car simulator environments such as TORCS and CARLA. In general, DRL is still at its infancy in terms of usability in real-world applications. Our goal in this paper is to encourage real-world deployment of DRL in various autonomous driving (AD) applications. We first provide an overview of the tasks in autonomous driving systems, reinforcement learning algorithms and applications of DRL to AD systems. We then discuss the challenges which must be addressed to enable further progress towards real-world deployment.

1 INTRODUCTION

Autonomous driving (AD) is a challenging application domain for machine learning (ML). Since the task of driving "well" is already open to subjective definitions, it is not easy to specify the correct behavioural outputs for an autonomous driving system, nor is it simple to choose the right input features to learn with. Correct driving behaviour is only loosely defined, as different responses to similar situations may be equally acceptable. ML-based control policies have to overcome the lack of a dense metric evaluating the driving quality over time, and the lack of a strong signal for expert imitation. Supervised learning methods do not learn the dynamics of the environment nor that of the agent (Sutton and Barto, 2018), while reinforcement learning (RL) is formulated to handle sequential decision processes, making it a natural approach for learning AD control policies.

In this article, we aim to outline the underlying principles of DRL for applications in AD. Passive perception which feeds into a control system does not scale to handle complex situations. DRL setting would enable active perception optimized for the specific control task. The rest of the paper is structured as follows. Section 2 provides background on the various modules of AD, an overview of reinforcement learning and a summary of applications of DRL to AD. Section 3 discusses challenges and open problems in applying DRL to AD. Finally, Section 4 summarizes the paper and provides key future directions.

2 BACKGROUND

The software architecture for Autonomous Driving systems comprises of the following high level tasks: Sensing, Perception, Planning and Control. While some (Bojarski et al., 2016) achieve all tasks in one unique module, others do follow the logic of one task - one module (Paden et al., 2016). Behind this separation rests the idea of information transmission from the sensors to the final stage of actuators and motors as illustrated in Figure 1. While an End-to-End approach would enable one generic encapsulated solution, the modular approach provides granularity for this multi-discipline problem, as separating the tasks in modules is the *divide and conquer* engineering ap-
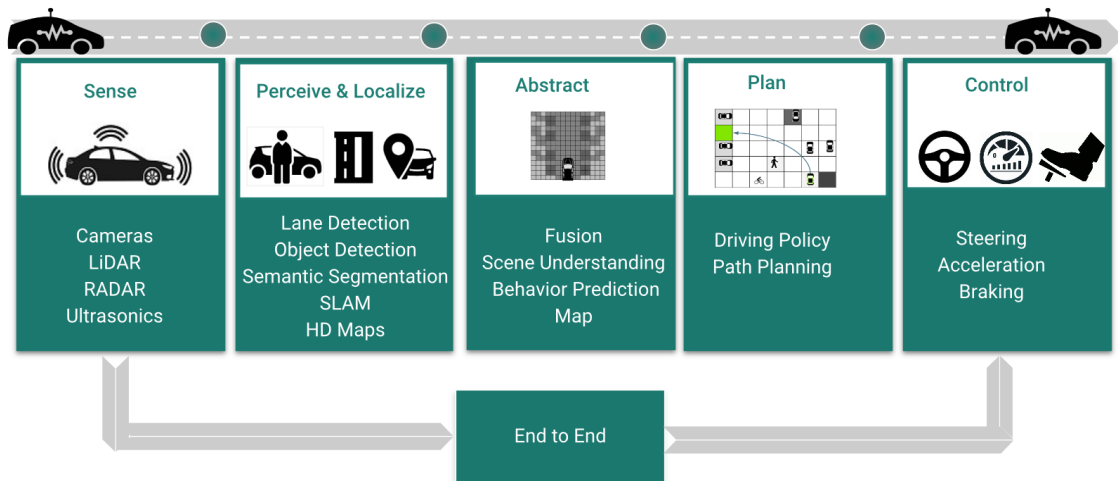
Figure 1: Fixed modules in a modern autonomous driving systems : **Sensor infrastructure** notably include Cameras, Radars, LiDARs (the Laser equivalent of Radars) or GPS-IMUs (GPS and Inertial Measurement Units provide an instantaneous position); their raw data is considered low level. This dynamic data is often processed into higher level descriptions as part of the Perception module. **Perception** estimates positions of: location in lane, cars, pedestrians, traffic lights and other semantic objects among other descriptors. It also provides road occupation over a larger spatial & temporal scale by combining maps. **Mapping** Localised high definition maps (HD maps) provide centrimetric reconstruction of buildings, static obstacles and dynamic objects, which frequently are crowdsourced. **Scene understanding** provide the high-level scene comprehension that includes detection, classification & localisation tasks, feeding the driving policy/planning module. **Path Planning** predicts the future actor trajectories and manoeuvres. A static shortest path from point *A* to point *B* with dynamic traffic information constraints are employed to calculate the path. **Vehicle control** orchestrates the high level orders for motion planning using simple closed loop systems based on sensors from the Perception task.

proach.

2.1 Review of Reinforcement learning

Reinforcement Learning (RL) (Sutton and Barto, 2018) is a family of algorithms which allow agents to learn how to act in different situations. In other words, how to establish a map, or a policy, from situations (states) to actions which maximize a numerical reward signal. RL has been successfully applied to many different fields such as helicopter control (Naik and Mammone, 1992), traffic signal control (Mannion et al., 2016a), electricity generator scheduling (Mannion et al., 2016b), water resource management (Mason et al., 2016), playing relatively simple Atari games (Mnih et al., 2015) and mastering a much more complex game of Go (Silver et al., 2016), simulated continuous control problems (Lillicrap et al., 2015), (Schulman et al., 2015), and controlling robots in real environments (Levine et al., 2016).

RL algorithms may learn estimates of state values, environment models or policies. In real-world applications, simple tabular representations of estimates are not scalable. Each additional feature tracked in the state leads to an exponential growth in the number of estimates that must be stored (Sutton and Barto, 2018).

Deep Neural Networks (DNN) have recently been applied as function approximators for RL agents, allowing agents to generalise knowledge to new unseen situations, along with new algorithms for problems with continuous state and action spaces. Deep RL agents can be value-based: DQN (Mnih et al., 2013), Double DQN (Van Hasselt et al., 2016), policy-based: TRPO (Schulman et al., 2015), PPO (Schulman et al., 2017); or actor-critic: DPG (Silver et al., 2014), DDPG (Lillicrap et al., 2015), A3C (Mnih et al., 2016). Model-based RL agents attempt to build an environment model, e.g. Dyna-Q (Sutton, 1990), Dyna-2 (Silver et al., 2008). Inverse reinforcement learning (IRL) aims to estimate the reward function given examples of agents actions, sensory input to the agent (state), and the model of the environment (Abbeel and Ng, 2004a), (Ziebart et al., 2008), (Finn et al., 2016), (Ho and Ermon, 2016).

2.2 DRL for Autonomous Driving

In this section, we visit the different modules of the autonomous driving system as shown in Figure 1 and describe how they are achieved using classical RL and Deep RL methods. A list of datasets and simulators for AD tasks is presented in Table 1.

Table 1: A collection of datasets and simulators to evaluate AD algorithms.

Dataset	Description
Berkeley Driving Dataset (Xu et al., 2017)	Learn driving policy from demonstrations
Baidu's ApolloScape	Multiple sensors & Driver Behaviour Profiles
Honda Driving Dataset (Ramanishka et al., 2018)	4 level annotation (stimulus, action, cause, and attention objects) for driver behavior profile.
Simulator	**Description**
CARLA (Dosovitskiy et al., 2017)	Urban Driving Simulator with Camera, LiDAR, Depth & Semantic segmentation
Racing Simulator TORCS (Wymann et al., 2000)	Testing control policies for vehicles
AIRSIM (Shah et al., 2018)	Resembling CARLA with support for Drones
GAZEBO (Koenig and Howard, 2004)	Multi-robo simulator for planning & control

Vehicle Control: Vehicle control classically has been achieved with predictive control approaches such as Model Predictive Control (MPC) (Paden et al., 2016). A recent review on motion planning and control task can be found by authors (Schwarting et al., 2018). Classical RL methods are used to perform optimal control in stochastic settings the Linear Quadratic Regulator (LQR) in linear regimes and iterative LQR (iLQR) for non-linear regimes are utilized. More recently, random search over the parameters for a policy network can perform as well as LQR (Mania et al., 2018).

One can note recent work on DQN which is used in (Yu et al., 2016) for simulated autonomous vehicle control where different reward functions are examined to produce specific driving behavior. The agent successfully learned the turning actions and navigation without crashing. In (Sallab et al., 2016) DRL system for lane keeping assist is introduced for discrete actions (DQN) and continuous actions (DDAC), where the TORCS car simulator is used and concluded that, as expected, the continuous actions provide smoother trajectories, and the more restricted termination conditions, the slower convergence time to learn. Wayve, a recent startup, has recently demonstrated an application of DRL (DDPG) for AD using a full-sized autonomous vehicle (Kendall et al., 2018). The system was first trained in simulation, before being trained in real time using onboard computers, and was able to learn to follow a lane, successfully completing a real-world trial on a 250 metre section of road.

DQN for Ramp Merging: The AD problem of ramp merging is tackled in (Wang and Chan, 2017), where DRL is applied to find an optimal driving policy using LSTM for producing an internal state containing historical driving information and DQN for Q-function approximation.

Q-function for Lane Change: A Reinforcement Learning approach is proposed in (Wang et al., 2018) to train the vehicle agent to learn an automated lane change in a smooth and efficient behavior, where the coefficients of the Q-function are learned from neural networks.

IRL for Driving Styles: Individual perception of comfort from demonstration is proposed in (Kuderer et al., 2015), where individual driving styles are modeled in terms of a cost function and use feature based inverse reinforcement learning to compute trajectories in vehicle autonomous mode. Using Deep Q-Networks as the refinement step in IRL is proposed in (Sharifzadeh et al., 2016) to extract the rewards. While evaluated in a simulated autonomous driving environment, it is shown that the agent performs a human-like lane change behavior.

Multiple-goal RL for Overtaking: In (Ngai and Yung, 2011) a multiple-goal reinforcement learning (MGRL) framework is used to solve the vehicle overtaking problem. This work is found to be able to take correct actions for overtaking while avoiding collisions and keeping almost steady speed.

Hierarchical Reinforcement Learning (HRL): Contrary conventional or flat RL, HRL refers to the decomposition of complex agent behavior using temporal abstraction, such as the options framework (Barto and Mahadevan, 2003). The problem of sequential decision making for autonomous driving with distinct behaviors is tackled in (Chen et al., 2018). A hierarchical neural network policy is proposed where the network is trained with the Semi-Markov Decision Process (SMDP) though the proposed hierarchical policy gradient method. The method is applied to a traffic light passing scenario, and it is shown that the method is able to select correct decisions providing better performance compared to

a non-hierarchical reinforcement learning approach. An RL-based hierarchical framework for autonomous multi-lane cruising is proposed in (Nosrati et al., 2018) and it is shown that the hierarchical design enables significantly better learning performance than a flat design for both DQN and PPO methods.

Frameworks: A framework for an end-end Deep Reinforcement Learning pipeline for autonomous driving is proposed in (Sallab et al., 2017), where the inputs are the states of the environment and their aggregations over time, and the output is the driving actions. The framework integrates RNNs and attention glimpse network, and tested for lane keep assist algorithm. In this section we reviewed in brief the applications of DRL and classical RL methods to different AD tasks.

2.3 Predictive Perception

In this section, we review some examples of applications of IOC and IRL to predictive perception tasks. Given a certain instance where the autonomous agent is driving in the scene, the goal of predictive perception algorithms are to predict the trajectories or intention of movement of other actors in the environment. The authors (Djuric et al., 2018), trained a deep convolutional neural network (CNN) to predict short-term vehicle trajectories, while accounting for inherent uncertainty of vehicle motion in road traffic. Deep Stochastic IOC (inverse optimal control) RNN Encoder-Decoder (DESIRE) (Lee et al., 2017) is a framework used to estimate a distribution and not just a simple prediction of an agent's future positions. This is based on the context (intersection, relative position of other agents).

Pedestrian Intention: Pedestrian intents to cross the road, board another vehicle, or is the driver in the parked car going to open the door (Erran and Scheider, 2017) . Authors in (Ziebart et al., 2009) perform maximum entropy inverse optimal control to learn a generic cost function for a robot to avoid pedestrians. Authors (Kitani et al., 2012) used inverse optimal control to predict pedestrian paths by considering scene semantics.

Traffic Negotiation: When in traffic scenarios involving multiple agents, policies learned require agents to negotiate movement in densely populated areas and with continuous movement. MobilEye demonstrated the use of the options framework (Shalev-Shwartz et al., 2016).

3 PRACTICAL CHALLENGES

Deep reinforcement learning is a rapidly growing field. We summarize the frequent challenges such as sample complexity and reward formulation, that could be encountered in the design of such methods.

3.1 Bootstrapping RL with Imitation

Ability learning by imitation is used by humans to teach other humans new skills. Demonstrations usually focus on state space essential areas from the expert's point of view. Learning from Demonstrations (LfD) is significant especially in domains where rewards are sparse. In imitation learning, an agent learns to perform a task from demonstrations without any feedback rewards. The agent learns a policy as a supervised learning process over state-action pairs. However, high quality demonstrations are hard to collect, leading to sub-optimal policies (Atkeson and Schaal, 1997). Accordingly, LfD can be used to initialize the learning agent with a policy inspired by performance of an expert. Then, RL can be conducted to discover a better policy by interacting with the environment. Learning a model of the environment condensing LfD and RL is presented in (Abbeel and Ng, 2005). Measuring the divergence between the current policy and the expert for policy optimization is proposed in (Kang et al., 2018). DQfD (Hester et al., 2017) pre-trains the agent and uses demonstrations by adding them into the replay buffer of the DQN and giving them additional priority. More recently, a training framework that combines LfD and RL for fast learning asynchronous agents is proposed in (Sobh and Darwish, 2018).

3.2 Exploration Issues with Imitation

In some cases, demonstrations from experts are not available or even not covering the state space leading to learning a poor policy. One solution consists in using the Data Aggregation (DAgger) (Ross and Bagnell, 2010) methods where the end-to-end learned policy is run and extracted observation-action pair is again labelled by the expert, and aggregated to the original expert observation-action dataset. Thus iteratively collecting training examples from both reference and trained policies explores more states and solves this lack of exploration. Following work on Search-based Structured Prediction (SEARN) (Ross and Bagnell, 2010), Stochastic Mixing Iterative Learning (SMILE) trains a stochastic stationary policy over several iterations and then makes use of a *geometric* stochastic mixing of the policies trained. In a

standard imitation learning scenario, the demonstrator is required to *cover* sufficient states so as to avoid unseen states during test. This constraint is costly and requires frequent human intervention. Hierarchical imitation learning methods reduce the sample complexity of standard imitation learning by performing data aggregation by organizing the action spaces in a hierarchy (Le et al., 2018).

3.3 Intrinsic Reward Functions

In controlled simulated environments such as games, an explicit reward signal is given to the agent along with its sensor stream. In real-world robotics and autonomous driving deriving, designing a *good* reward functions is essential so that the desired behaviour may be learned. The most common solution has been reward shaping (Ng et al., 1999) and consists in supplying additional rewards to the agent along with that provided by the underlying MDP. Rewards as already pointed earlier in the paper, could be estimated by inverse RL (IRL) (Abbeel and Ng, 2004b), which depends on expert demonstrations.

In the absence of an explicit reward shaping and expert demonstrations, agents can use intrinsic rewards or intrinsic motivation (Chentanez et al., 2005) to evaluate if their actions were good. Authors (Pathak et al., 2017) define curiosity as the error in an agents ability to predict the consequence of its own actions in a visual feature space learned by a self-supervised inverse dynamics model. In (Burda et al., 2018) the agent learns a next state predictor model from its experience, and uses the error of the prediction as an intrinsic reward. This enables that agent to determine what could be a useful behavior even without extrinsic rewards.

3.4 Bridging the Simulator-reality Gap

Training deep networks requires collecting and annotating a lot of data which is usually costly in terms of time and effort. Using simulation environments enables the collection of large training datasets. However, the simulated data do not have the same data distribution compared to the real data. Accordingly, models trained on simulated environments often fail to generalise on real environments. *Domain adaptation* allows a machine learning model trained on samples from a source domain to generalise to a target domain. **Feature-level domain adaptation** focuses on learning domain-invariant features. In work of (Ganin et al., 2016), the decisions made by deep neural networks are based on features that are both discriminative and invariant to the change of domains. **Pixel**

level domain adaptation focuses on stylizing images from the source domain to make them *similar* to images of the target domain, based on image conditioned generative adversarial networks (GANs). In (Bousmalis et al., 2017b), the model learns a transformation in the pixel space from one domain to the other, in an unsupervised way. GAN is used to adapt simulated images to look like as if drawn from the real domain. Both feature-level and pixel-level domain adaptation combined in (Bousmalis et al., 2017a), where the results indicate that including simulated data can improve the vision-based grasping system, achieving comparable performance with 50 times fewer real-world samples. Another relatively simpler method is introduced in (Peng et al., 2017), by *dynamics randomizing* of the simulator during training, policies are capable of generalising to different dynamics without any training on the real system. *RL with Sim2Real*: A model trained in virtual environment is shown to be workable in real environment (Pan et al., 2017). Virtual images rendered by a simulator environment are first segmented to scene parsing representation and then translated to synthetic realistic images by the proposed image translation network. The proposed network segments the simulated image input, and then generates a synthetic realistic images. Accordingly, the driving policy trained by reinforcement learning can be easily adapted to real environment.

World Models proposed in (Ha and Schmidhuber, 2018b; Ha and Schmidhuber, 2018a) are trained quickly in an unsupervised way, via a Variational AutoEncoder (VAE), to learn a compressed spatial and temporal representation of the environment leading to learning a compact policy. Moreover, the agent can train inside its own dream and transfer the policy back into the actual environment.

3.5 Data Efficient and Fast Adapting RL

Depending on the task being solved, RL require a lot of observations to cover the state space. Efficieny is usually achieved with imitation learning, reward shaping and transfer learning. Readers are directed towards the survey on transfer learning in RL here (Taylor and Stone, 2009). The primary motivation in transfer learning in RL is to reuse previously trained policies/models for a source task, so as to reduce the current target task's training time. Authors in (Liaw et al., 2017) study the policy composition problem where composing previously learned basis policies, e.g., driving in different conditions or over different terrain types, the goal is to be able to reuse them for

a novel task that is a mixture of previously seen dynamics by learning a meta-policy, that maximises the reward. Their results show that learning a new policy for a new task takes longer than a meta-policy learnt using basis policies.

Meta-learning, or learning to learn, is an important approach towards versatile agents that can adapt quickly to news tasks. The idea of having one neural network interact with another one for meta-learning has been applied in (Duan et al., 2016) and (Wang et al., 2016). More recently, the Model-Agnostic Meta-Learning (MAML) is proposed in (Finn et al., 2017), where the meta-learner seeks to find an initialization for the parameters of a neural network, that can be adapted quickly for a new task using only few examples. Continuous adaptation in dynamically changing and adversarial scenarios is presented in (Al-Shedivat et al., 2017) via a simple gradient-based meta-learning algorithm. Additionally, Reptile (Nichol et al., 2018) is mathematically similar to first-order MAML, making it consume less computation and memory than MAML.

3.6 Incorporating Safety in DRL for AD

Deploying an autonomous vehicle after training directly could be dangerous. We review different approaches to incorporate safety into DRL algorithms.

SafeDAgger (Zhang and Cho, 2017) introduces a safety policy that learns to predict the error made by a primary policy trained initially with the supervised learning approach, without querying a reference policy. An additional safe policy takes both the partial observation of a state and a primary policy as inputs, and returns a binary label indicating whether the primary policy is likely to deviate from a reference policy without querying it.

Multi-agent RL for Comfort Driving and Safety: In (Shalev-Shwartz et al., 2016), autonomous driving is addressed as a multi-agent setting where the host vehicle applies negotiations in different scenarios; where balancing is maintained between unexpected behavior of other drivers and not to be too defensive. The problem is decomposed into a policy for learned desires to enable comfort of driving, and trajectory planning with hard constraints for safety of driving.

DDPG and Safety Based Control: The deep reinforcement learning (DDPG) and safety based control are combined in (Xiong et al., 2016), including artificial potential field method that is widely used for robot path planning. Using TORCS environment,

the DDPG is used first for learning a driving policy in a stable and familiar environment, then policy network and safety-based control are combined to avoid collisions. It was found that combination of DRL and safety-based control performs well in most scenarios.

Negative-Avoidance for Safety: In order to enable DRL to escape local optima, speed up the training process and avoid danger conditions or accidents, Survival-Oriented Reinforcement Learning (SORL) model is proposed in (Ye et al., 2017), where survival is favored over maximizing total reward through modeling the autonomous driving problem as a constrained MDP and introducing Negative-Avoidance Function to learn from previous failure. The SORL model found to be not sensitive to reward function and can use different DRL algorithms like DDPG.

3.7 Other Challenges

Multimodal Sensor Policies. Modern autonomous driving systems constitute of multiple modalities (Sobh et al., 2018), for example Camera RGB, Depth, Lidar and others sensors. Authors in (Liu et al., 2017) propose end-to-end learning of policies that leverages sensor fusion to reduced performance drops in noisy environment and even in the face of partial sensor failure by using Sensor Dropout to reduce sensitivity to any sensor subset.

Reproducibility. State-of-the-art deep RL methods are seldom reproducible. Non-determinism in standard benchmark environments, combined with variance intrinsic to the methods, can make reported results tough to interpret, authors discuss common issues and challenges (Henderson et al., 2018). Therefore, in the future it would be helpful to develop standardized benchmarks for evaluating autonomous vehicle control algorithms, similar to the benchmarks such as KITTI (Geiger et al., 2012) which are already available for AD perception tasks.

4 CONCLUSION

AD systems present a challenging environment for tasks such as perception, prediction and control. DRL is a promising candidate for the future development of AD systems, potentially allowing the required behaviours to be learned first in simulation and further refined on real datasets, instead of being explicitly programmed. In this article, we provide an overview of

AD system components, DRL algorithms and applications of DRL for AD. We discuss the main challenges which must be addressed to enable practical and wide-spread use of DRL in AD applications. Although most of the work surveyed in this paper were conducted on simulated environments, it is encouraging that applications on real vehicles are beginning to appear, e.g. (Kendall et al., 2018). The key challenges in constructing a complete real-world system would require resolving key challenges such as safety in RL, improving data efficiency and finally enabling transfer learning using simulated environments. To add, environment dynamics are better modeled by considering the predictive perception of other actors. We hope that this work inspires future research and development on DRL for AD, leading to increased real-world deployments in AD systems.

REFERENCES

Abbeel, P. and Ng, A. Y. (2004a). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM.

Abbeel, P. and Ng, A. Y. (2004b). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 1–, New York, NY, USA. ACM.

Abbeel, P. and Ng, A. Y. (2005). Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 1–8, Bonn, Germany.

Al-Shedivat, M., Bansal, T., Burda, Y., Sutskever, I., Mordatch, I., and Abbeel, P. (2017). Continuous adaptation via meta-learning in nonstationary and competitive environments. *arXiv preprint arXiv:1710.03641*.

Atkeson, C. G. and Schaal, S. (1997). Robot learning from demonstration. In *Proceedings of the 14th International Conference on Machine Learning*, volume 97, pages 12–20, Nashville, USA.

Barto, A. G. and Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(4):341–379.

Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. (2016). End to end learning for self-driving cars. *CoRR*, abs/1604.07316.

Bousmalis, K., Irpan, A., Wohlhart, P., Bai, Y., Kelcey, M., Kalakrishnan, M., Downs, L., Ibarz, J., Pastor, P., Konolige, K., et al. (2017a). Using simulation and domain adaptation to improve efficiency of deep robotic grasping. *arXiv preprint arXiv:1709.07857*.

Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., and Krishnan, D. (2017b). Unsupervised pixel-level domain adaptation with generative adversarial networks.

In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 7.

Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., and Efros, A. A. (2018). Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*.

Chen, J., Wang, Z., and Tomizuka, M. (2018). Deep hierarchical reinforcement learning for autonomous driving with distinct behaviors. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1239–1244. IEEE.

Chentanez, N., Barto, A. G., and Singh, S. P. (2005). Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288.

Djuric, N., Radosavljevic, V., Cui, H., Nguyen, T., Chou, F.-C., Lin, T.-H., and Schneider, J. (2018). Motion prediction of traffic actors for autonomous driving using deep convolutional networks. *arXiv preprint arXiv:1808.05819*.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. (2016). Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.

Erran, L. and Scheider, J. (2017). Machine learning for autonomous vehicless. In *ICML Workshop on Machine Learning for Autonomous Vehicles 2017,*.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*.

Finn, C., Levine, S., and Abbeel, P. (2016). Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, pages 49–58.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE.

Ha, D. and Schmidhuber, J. (2018a). Recurrent world models facilitate policy evolution. *arXiv preprint arXiv:1809.01999*. Accepted at NIPS 2018.

Ha, D. and Schmidhuber, J. (2018b). World models. *arXiv preprint arXiv:1803.10122*.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. (2018). Deep reinforcement learning that matters. In McIlraith, S. A. and Weinberger, K. Q., editors, *(AAAI-18), the 30th innovative Applications of Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3207–3214. AAAI Press.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A.,

Dulac-Arnold, G., et al. (2017). Deep q-learning from demonstrations. *arXiv preprint arXiv:1704.03732*.

Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573.

Kang, B., Jie, Z., and Feng, J. (2018). Policy optimization with demonstrations. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 2469–2478, Stockholm, Sweden.

Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J.-M., Lam, V.-D., Bewley, A., and Shah, A. (2018). Learning to drive in a day. *arXiv preprint arXiv:1807.00412*.

Kitani, K. M., Ziebart, B. D., Bagnell, J. A., and Hebert, M. (2012). Activity forecasting. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., editors, *Computer Vision – ECCV 2012*, pages 201–214, Berlin, Heidelberg. Springer Berlin Heidelberg.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2149–2154.

Kuderer, M., Gulati, S., and Burgard, W. (2015). Learning driving styles for autonomous vehicles from demonstration. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2641–2646. IEEE.

Le, H. M., Jiang, N., Agarwal, A., Dudík, M., Yue, Y., and Daumé III, H. (2018). Hierarchical imitation and reinforcement learning. *arXiv preprint arXiv:1803.00590*.

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H., and Chandraker, M. (2017). Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–345.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373.

Liaw, R., Krishnan, S., Garg, A., Crankshaw, D., Gonzalez, J. E., and Goldberg, K. (2017). Composing meta-policies for autonomous driving using hierarchical deep reinforcement learning. *arXiv preprint arXiv:1711.01503*.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Liu, G.-H., Siravuru, A., Prabhakar, S., Veloso, M., and Kantor, G. (2017). Learning end-to-end multimodal sensor policies for autonomous navigation. In Levine, S., Vanhoucke, V., and Goldberg, K., editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 249–261. PMLR.

Mania, H., Guy, A., and Recht, B. (2018). Simple random search provides a competitive approach to reinforcement learning. *arXiv preprint arXiv:1803.07055*.

Mannion, P., Duggan, J., and Howley, E. (2016a). An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In McCluskey, L. T., Kotsialos, A., Müller, P. J., Klügl, F., Rana, O., and Schumann, R., editors, *Autonomic Road Transport Support Systems*, pages 47–66. Springer International Publishing.

Mannion, P., Mason, K., Devlin, S., Duggan, J., and Howley, E. (2016b). Multi-objective dynamic dispatch optimisation using multi-agent reinforcement learning. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1345–1346.

Mason, K., Mannion, P., Duggan, J., and Howley, E. (2016). Applying multi-agent reinforcement learning to watershed management. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS 2016)*.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

Naik, D. K. and Mammone, R. (1992). Meta-neural networks that learn by learning. In *Neural Networks, 1992. IJCNN., International Joint Conference on*, volume 1, pages 437–442. IEEE.

Ng, A. Y., Harada, D., and Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287.

Ngai, D. C. K. and Yung, N. H. C. (2011). A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):509–522.

Nichol, A., Achiam, J., and Schulman, J. (2018). On first-order meta-learning algorithms. *CoRR, abs/1803.02999*.

Nosrati, M. S., Abolfathi, E. A., Elmahgiubi, M., Yadmellat, P., Luo, J., Zhang, Y., Yao, H., Zhang, H., and Jamil, A. (2018). Towards practical hierarchical reinforcement learning for multi-lane autonomous driving.

Paden, B., Čáp, M., Yong, S. Z., Yershov, D. S., and Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *CoRR*, abs/1604.07446.

Pan, X., You, Y., Wang, Z., and Lu, C. (2017). Virtual to real reinforcement learning for autonomous driving. *arXiv preprint arXiv:1704.03952*.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. (2017). Sim-to-real transfer of robotic control with dynamics randomization. *arXiv preprint arXiv:1710.06537*.

Ramanishka, V., Chen, Y.-T., Misu, T., and Saenko, K. (2018). Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7699–7707.

Ross, S. and Bagnell, D. (2010). Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668.

Sallab, A. E., Abdou, M., Perot, E., and Yogamani, S. (2016). End-to-end deep reinforcement learning for lane keeping assist. *arXiv preprint arXiv:1612.04340*.

Sallab, A. E., Abdou, M., Perot, E., and Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schwarting, W., Alonso-Mora, J., and Rus, D. (2018). Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*.

Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2018). Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pages 621–635. Springer.

Shalev-Shwartz, S., Shammah, S., and Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*.

Sharifzadeh, S., Chiotellis, I., Triebel, R., and Cremers, D. (2016). Learning to drive using inverse reinforcement learning and deep q-networks. *arXiv preprint arXiv:1612.03653*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *ICML*.

Silver, D., Sutton, R. S., and Müller, M. (2008). Sample-based learning and search with permanent and transient memories. In *Proceedings of the 25th international conference on Machine learning*, pages 968–975. ACM.

Sobh, I., Amin, L., Abdelkarim, S., Elmadawy, K., Saeed, M., Abdeltawab, O., Gamal, M., and El Sallab, A. (2018). End-to-end multi-modal sensors fusion system for urban automated driving. In *NIPS Workshop on Machine Learning for Intelligent Transportation Systems*.

Sobh, I. and Darwish, N. (2018). End-to-end framework for fast learning asynchronous agents. In *Proceedings of the 32nd Conference on Neural Information Processing Systems, Imitation Learning and its Challenges in Robotics workshop*.

Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine Learning Proceedings 1990*, pages 216–224. Elsevier.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Taylor, M. E. and Stone, P. (2009). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685.

Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *AAAI*, volume 16, pages 2094–2100.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. (2016). Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.

Wang, P. and Chan, C.-Y. (2017). Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge. In *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*, pages 1–6. IEEE.

Wang, P., Chan, C.-Y., and de La Fortelle, A. (2018). A reinforcement learning based approach for automated lane change maneuvers. *arXiv preprint arXiv:1804.07871*.

Wymann, B., Espié, E., Guionneau, C., Dimitrakakis, C., Coulom, R., and Sumner, A. (2000). Torcs, the open racing car simulator. *Software available at http://torcs. sourceforge. net*, 4.

Xiong, X., Wang, J., Zhang, F., and Li, K. (2016). Combining deep reinforcement learning and safety based control for autonomous driving. *arXiv preprint arXiv:1612.00147*.

Xu, H., Gao, Y., Yu, F., and Darrell, T. (2017). End-to-end learning of driving models from large-scale video datasets. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3530–3538.

Ye, C., Ma, H., Zhang, X., Zhang, K., and You, S. (2017). Survival-oriented reinforcement learning model: An effcient and robust deep reinforcement learning algorithm for autonomous driving problem. In *International Conference on Image and Graphics*, pages 417–429. Springer.

Yu, A., Palefsky-Smith, R., and Bedi, R. (2016). Deep reinforcement learning for simulated autonomous vehicle control. *Course Project Reports: Winter 2016 (CS231n: Convolutional Neural Networks for Visual Recognition)*, pages 1–7.

Zhang, J. and Cho, K. (2017). Query-efficient imitation learning for end-to-end simulated driving. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA.*, pages 2891–2897.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA.

Ziebart, B. D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J. A., Hebert, M., Dey, A. K., and Srinivasa, S. (2009). Planning-based prediction for pedestrians. In *Proc. IROS*.