2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Learning robust driving policies without online exploration

Daniel Graves[1], Nhat M. Nguyen[1], Kimia Hassanzadeh[1], Jun Jin[1], Jun Luo[1]

***Abstract*— We propose a multi-time-scale predictive representation learning method to efficiently learn robust driving policies in an offline manner that generalize well to novel road geometries, and damaged and distracting lane conditions which are not covered in the offline training data. We show that our proposed representation learning method can be applied easily in an offline (batch) reinforcement learning setting demonstrating the ability to generalize well and efficiently under novel conditions compared to standard batch RL methods. Our proposed method utilizes training data collected entirely offline in the real-world which removes the need of intensive online explorations that impede applying deep reinforcement learning on real-world robot training. Various experiments were conducted in both simulator and real-world scenarios for the purpose of evaluation and analysis of our proposed claims.**

I. Introduction

Learning to drive is a challenging problem that is a long-standing goal in robotics and autonomous driving. In the early days of autonomous driving, a popular approach to staying within a lane was based on lane marking detection [1]. However, a significant challenge with this approach is the lack of robustness to missing, occluded or damaged lane markings [2] where most roads in the US are not marked with reliable lane markings on either side of the road [3]. Modern approaches mitigate some of these issues by constructing high definition maps and developing accurate localization techniques [4], [5], [6], [7]. However, scaling both the map and localization approaches globally in a constantly changing world is still very challenging for autonomous driving and robotic navigation [4].

Recently, there have been a growing number of successes in AI applied to robotics and autonomous driving [8], [9], [10], [11], [12], [3]. These data-driven approaches can be divided into two categories: (1) behavior cloning, and (2) reinforcement learning (RL). Behavior cloning suffers from generalization challenges since valuable negative experiences are rarely collected; in addition they cannot offer performance better than the behavior being cloned [8], [13], [12]. RL on the other-hand is a promising direction for vision-based control [14]; however, RL is usually not practical because it requires extensive online exploration in the environment to find the best policy that maximizes the cumulative reward [11], [15], [16]. Moreover, the success in game environments like Go [17] doesn't always transfer well to success in the real-world where an agent is expected to learn policies that generalize well [18], [16]. A key challenge is that RL overfits to the training environment where learned policies tend to perform poorly on novel situations not seen during training [19], [20], [21], [22]. We aim to address the issues of learning both practical and general driving policies in the real-world by combining a novel representation learning approach with offline RL without any online exploration.

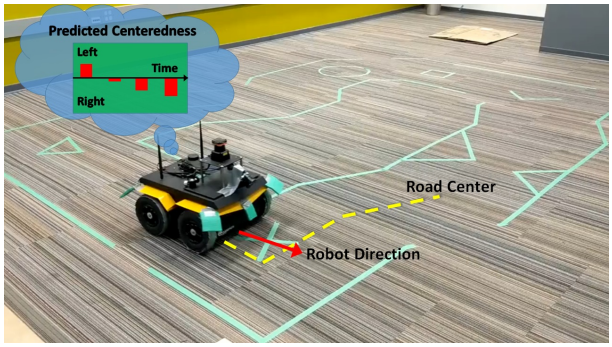


Fig. 1: Lane keeping of a Jackal Robot using vision-based counterfactual predictions of the future lane centeredness over multiple time scales to represent the state of the agent in RL

Offline RL, or learning RL policies from data without exploration [23], could potentially address many of the practicality issues with applying standard online RL in the real-world. Unfortunately, deep offline RL struggles to generalize to data not in the training set [24], [25]. Our approach applies novel representation learning based on counterfactual predictions [26], [27], [25], shown in Figure 1, to address the generalization issue. We learn predictions of future lane centeredness and road angle from offline data safely collected in the environment using noisy localization sources during training, eliminating the need for expensive, on-vehicle, high accuracy localization sensors during deployment. State-of-the-art offline RL [23] is then applied using these counterfactual predictions as a low dimensional representation of the state to learn a policy to drive the vehicle. These counterfactual predictions are motivated in psychology [28], [29] where we find predictions aid in an agent's understanding of the world, particularly in driving [30]. Similar works in classical control have shown how anticipation of the future is important for driving at the limits of stability through feed-forward [31] and model predictive control [32]. Our work is motivated by the predictive state hypothesis [33], [34] that claims counterfactual predictions help an agent generalize and adapt quickly to new problems [35].

The significance of our approach is that it demonstrates practical value in autonomous driving and real-world RL without requiring extensive maps, robust localization techniques or robust lane marking and curb detection. We demonstrate that

[1]Noah's Ark Lab, Huawei Technologies Canada {daniel.graves, minh.nhat.nguyen, kima.hassanzadeh, jun.jin1, jun.luo1}@huawei.com

our approach generalizes to never-before seen roads including those with damaged and distracting lane markings. The novel contributions of this work are summarized as follows: (1) an algorithm for learning counterfactual predictions from real-world driving data with behavior distribution estimation, (2) an investigation into the importance of predictive representations for learning good driving policies that generalize well to new roads and damaged lane markings, and (3) the first demonstration of deep RL applied to autonomous driving with real-world data without any online exploration.

II. Related Works

Deep learning approaches to driving: There have been many attempts to apply deep learning to driving including deep RL and imitation learning [11]; however generalization is a key challenge. ChaufferNet [36] used a combination of imitation learning and predictive models to synthesize the worst case scenarios but more work is needed to improve the policy to achieve performance competitive with modern motion planners. Another approach trained the agent entirely in the simulator where transfer to the real-world could be challenging to achieve [11]. DeepDriving [37] learned affordance predictions of road angle from an image for multi-lane driving in simulation using offline data collected by human drivers. However, in contrast with our proposed method, DeepDriving used heuristics and rules to control the vehicle instead of learning a policy with RL. Moreover, DeepDriving learned predictions of the current lane centeredness and current road angle rather than long-term counterfactual predictions of the future.

Offline RL in real-world robot training: There are many prior arts in offline (batch) RL [38], [24]. However, most prior arts in offline RL have challenges learning good policies in the deep setting [24]. The current state of the art in offline RL is batch constrained Q-learning (BCQ) [23], [24] where success is demonstrated in simulation environments such as Atari but the results still perform badly in comparison to online learning. The greatest challenge with offline RL is the difficulty in covering the state-action space of the environment resulting in holes in the training data where extrapolation is necessary. [39] applied a novel offline RL approach to playing soccer with a real-world robot by exploiting the episodic nature of the problem. Our work overcomes these challenges and is, to the best of our knowledge, the first successful real-world robotic application of batch RL with deep learning.

Counterfactual prediction learning: Learning counterfactual predictions as representation of the state of the agent has been proposed before in the real-world [40], [41]. Other approaches demonstrate counterfactual predictions but don't provide a way to use them [26], [42], [27]. While experiments with counterfactual predictions show a lot of promise for improving learning and generalization, most experiments are in simple tabular domains [33], [34], [35]. Auxiliary tasks and similar prediction problems have been applied to deep RL task in simulation but assume the policy is the same as the policy being learned and thus are not counterfactual predictions [43], [44], [29], [45].

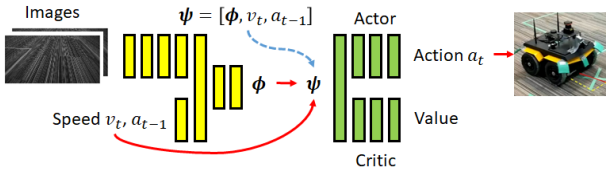


Fig. 2: Overall architecture of the RL system involved learns a predictive representation $\boldsymbol{\psi}$ to represent the state of the agent. Camera is the only environment sensor at test time.

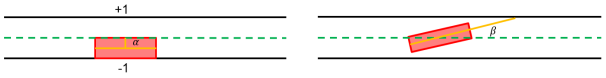
III. Predictive Control for Autonomous Driving

Let us consider the usual setting of an MDP described by a set of states S, a set of actions A, and transition dynamics with probability $P(s'|s,a)$ of transitioning to next state s' after taking action a from state s, and a reward r. The objective of an MDP is to learn a policy π that maximizes the future discounted sum of rewards in a given state. Obtaining the state of the agent in an MDP environment is not trivial especially with deep RL where the policy is changing because the target is moving [46]. Our approach is to learn an intermediate representation mapping sensor readings s to a limited number of counterfactual predictions ϕ as a representation of the state for deep RL. This has the advantage of pushing the heavy burden of deep feature representation learning in RL to the easier problem of prediction learning [47], [48], [49], [43].

The overall architecture of the system is depicted in Figure 2. The proposal is to represent the state of the agent as a vector $\boldsymbol{\psi}$ which is the concatenation of a limited number of the predictions $\boldsymbol{\phi}$, the current speed of the vehicle v_t and the previous action taken a_{t-1}. The predictions $\boldsymbol{\phi}$ are counterfactual predictions, also called general value functions [26]. The previous action taken is needed due to the nature of the predictions which are relative to the last action.

Learning a policy $\pi(\boldsymbol{\psi})$ could provide substantial benefits over learning π from image observations: (1) improving learning performance and speed, (2) enabling batch RL from offline data, and (3) improving generalization of the driving policy. Our approach is to learn a value function $Q(s,a)$ and a deterministic policy $\pi(\boldsymbol{\psi})$ that maximizes that value function using batch constrained Q-learning (BCQ) [23]. While the networks can be modelled as one computational graph, the gradients from the policy and value function network are not back-propagated through the prediction network to decouple the representation learning when learning from the offline data. Thus, training happens in two phases: (1) learning the prediction network, (2) learning the policy and value function.

During the first phase of training, a low-accuracy localization algorithm, based on 2D lidar scan matching, produces the lane centeredness α and relative road angle β of the vehicle, depicted in Figure 3, that are used to train the prediction network. The prediction network is a single network that predicts the lane centeredness and relative road angle over multiple temporal horizons depicted in Figure 4: these



(a) Lane centeredness α (b) Road angle β

Fig. 3: An illustration of (a) lane centeredness position α, and (b) the road angle β which is the angle between the direction of the vehicle and the direction of the road.

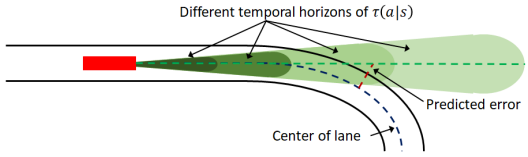


Fig. 4: An illustration of the multiple temporal horizons of the predictions ϕ.

are predictions of the future lane centeredness and relative road angle rather than the current estimates returned by the localization algorithm. They are chosen because they represent both present and future lane centeredness information needed to steer [30]. These predictions are discounted sums of future lane centeredness and relative road angle respectively that are learned with GVFs [26]:

$$\boldsymbol{\phi}(s) = \mathbb{E}_\tau[\sum_{i=0}^{\infty} \gamma^i \boldsymbol{c}_{t+i+1} | s_t = s] \quad (1)$$

where $\boldsymbol{c}_{t+i+1}$ is the cumulant vector consisting of the current lane centeredness α and current relative road angle β. It is important to understand that $\boldsymbol{\phi}(s)$ predicts the sum of all future lane centeredness and road angle values collected under some policy τ. The policy τ is counterfactual in the sense that it is different from the behavior policy μ used to collect the data and the learned policy π. Formally, the policy $\tau(a_t|s_t, a_{t_1}) = \mathcal{N}(a_{t-1}, \Sigma)$ where $\Sigma = 0.0025I$ is a diagonal covariance matrix. The meaning of this policy is to "keep doing what you are doing", similar to the one used in [32] for making counterfactual predictions. Therefore, $\boldsymbol{\phi}(s)$ predicts the discounted sum of future lane centeredness and road angle if the vehicle takes similar actions to its last action. Moreover, $\boldsymbol{\phi}(s)$ can be interpreted as predictions of the deviation from the desired lane centeredness and road angle. Counterfactual predictions can be thought of as anticipated future "errors" that allow controllers to take corrective actions before the errors occur. The discount factor γ controls the temporal horizon of the prediction. It is critical to learn $\boldsymbol{\phi}(s)$ for different values of γ in order to control both steering and speed. The details for learning $\boldsymbol{\phi}(s)$ are provided in the next section.

During the second stage of training, the localization algorithm is no longer needed; it was used to provide the labels for training the predictive representation in the first stage. Instead, the counterfactual predictions $\boldsymbol{\phi}$ are concatenated with the vehicle speed v_t and last action a_{t-1} to form a predictive representation $\boldsymbol{\psi}$. The RL agent receives $\boldsymbol{\psi}$ as the state of the agent in the environment which is used to predict the value and produce the next action a_t as depicted in Figure 2. In our offline learning approach, we used the state of the art batch RL BCQ [23][24] to train the policy.

Note that the same architecture can also be applied online where the counterfactual prediction, policy and value networks are all learned online simultaneously with deep deterministic policy gradient (DDPG) [50] but the details are left in the appendix.

IV. Predictive Learning

The counterfactual predictions given in Equation (1) are general value functions (GVFs) [26] that are learned with a novel combination of different approaches including (1) off-policy, or counterfactual, prediction learning with importance resampling [47], and (2) behavior estimation with the density ratio trick [51].

A. Counterfactual Predictions

To ask a counterfactual predictive question, we use the GVF framework, where one must define a cumulant $c_t = c(s_t, a_t, s_{t+1})$, a.k.a. pseudo-reward, a policy distribution $\tau(a|s)$ and continuation function $\gamma_t = \gamma(s_t, a_t, s_{t+1})$. The answer to the predictive question is the expectation of the return ϕ_t, when following policy τ, defined by

$$\phi^\tau(s) = \mathbb{E}_\tau[\sum_{k=0}^{\infty} (\prod_{j=0}^{k-1} \gamma_{t+j+1}) c_{t+k+1} | s_t = s, a_t = a] \quad (2)$$

where the cumulant is c_t and $0 \leq \gamma_t \leq 1$ [26]. This is the more general form for learning a prediction than the one given in Equation (1) where the only difference is that γ is replaced by a continuation function which allows for predictions that predict the sum of cumulants until an episodic event occurs such as going out of lane. The agent usually collects experience under a different behavior policy $\mu(a|s)$. When τ is different from both the behavior policy μ and the policy being learned π, the predictive question is a counterfactual prediction[1]. Cumulants are often scaled by a factor of $1 - \gamma$ when γ is a constant in non-episodic predictions. The counterfactual prediction $\phi^\tau(s)$ is a general value function (GVF) is approximated by a deep neural network parameterized by θ to learn (2). The parameters θ are optimized with gradient descent minimizing the following loss function

$$L(\theta) = \mathbb{E}_\mu[\rho \delta^2] \quad (3)$$

where $\delta = \phi^\tau(s; \theta) - y$ is the TD error and $\rho = \frac{\tau(a|s)}{\mu(a|s)}$ is the importance sampling ratio to correct for the difference between the policy distribution τ and behavior distribution μ. Note that only the behavior policy distribution is corrected; but the expectation is still over the state visitation distribution under the policy μ. In practice, this is usually not an issue

[1]Some literature call this an off-policy prediction

[47]. The target y is produced by bootstrapping a prediction of the value of the next state [52] under policy τ

$$y = \mathbb{E}_{s_{t+1}\sim P}[c_{t+1} + \gamma\phi^{\tau}(s_{t+1};\hat{\theta})|s_t = s, a_t = a] \quad (4)$$

where y is a bootstrapped prediction using the most recent parameters $\hat{\theta}$ that are assumed constant in the gradient computation. Learning a counterfactual prediction with a fixed policy τ tends to be very stable when minimizing $L(\theta)$ using gradient descent approaches and therefore doesn't require target networks originally used in [46] to stabilize DQN.

The gradient of the loss function (3) is given by

$$\nabla_\theta L(\theta) = \mathbb{E}_\mu[\rho\delta\nabla_\theta\phi^{\tau}(s;\theta)] \quad (5)$$

However, updates with importance sampling ratios are known to have high variance which may negatively impact learning; instead we use the importance resampling technique to reduce the variance of the updates [47]. With importance resampling, a replay buffer D of size N is required and the gradient is estimated from a mini-batch and multiplied with the average importance sampling ratio of the samples in the buffer $\bar{\rho} = \frac{\sum_{i=1}^{N}\rho_i}{N}$.

The gradient with importance resampling is given by

$$\nabla_\theta L(\theta) = \mathbb{E}_{s,a\sim D_\rho}[\bar{\rho}\delta\nabla_\theta\hat{v}^{\tau}(s;\theta)] \quad (6)$$

where D_ρ is a distribution of the transitions in the replay buffer proportional to the importance sampling ratio. The probability for transition $i = 1...N$ is given by $D_i = \frac{\rho_i}{\sum_{j=1}^{N}\rho_j}$ where the importance sampling ratio is $\rho_i = \frac{\tau(a_i|s_i)}{\mu(a_i|s_i)}$. An efficient data structure for the replay buffer is the SumTree used in prioritized experience replay [53].

B. Behavior Estimation

When learning predictions from real-world driving data, one needs to know the behavior policy distribution $\mu(a|s)$; however, in practice this is rarely known. Instead we estimate it using the density ratio trick [51] where the ratio of two probability densities can be expressed as a ratio of discriminator class probabilities that distinguish samples from the two distributions. Let us define an intermediate probability density function $\eta(a|s)$ such as the uniform distribution; this will be compared to the behavior distribution $\mu(a|s)$ which we desire to estimate. The class labels $y = +1$ and $y = -1$ are labels given to samples from $\mu(a|s)$ and $\eta(a|s)$. A discriminator $g(a, s)$ is learned that distinguishes state action pairs from the two distributions using the cross-entropy loss. The ratio of the densities can be computed using only the discriminator $g(a, s)$.

$$\begin{aligned}\frac{\mu(a|s)}{\eta(a|s)} &= \frac{p(a|s, y = +1)}{p(a|s, y = -1)} = \frac{p(y = +1|a, s)/p(y = +1)}{p(y = -1|a, s)/p(y = -1)} \\ &= \frac{p(y = +1|a, s)}{p(y = -1|a, s)} = \frac{g(a, s)}{1 - g(a, s)}\end{aligned} \quad (7)$$

Here we assume that $p(y = +1) = p(y = -1)$. From this result, we can estimate $\mu(a|s)$ with $\hat{\mu}(a|s)$ as follows

$$\hat{\mu}(a|s) = \frac{g(a, s)}{1 - g(a, s)}\eta(a|s) \quad (8)$$

where $\eta(a|s)$ is a known distribution over action conditioned on state. Choosing $\eta(a|s)$ to be the uniform distribution ensures that the discriminator is well trained against all possible actions in a given state; thus good performance is achieved with sufficient coverage of the state space rather than the state-action space. Alternatively, one can estimate the importance sampling ratio without defining an additional distribution η by replacing the distribution η with τ; however, defining η to be a uniform distribution ensures the discriminator is learned effectively across the entire action space. The combined algorithms for training counterfactual predictions with an unknown behavior distribution are given in the Appendix for both the online and offline RL settings.

V. Experiments

Our approach to learning counterfactual predictions for representing the state used in RL to learn a driving policy is applied to two different domains. The first set of experiments is conducted on a Jackal robot in the real-world where we demonstrate the practicality of our approach and its robustness to damaged and distracting lane markings. The second set of experiments is conducted in the TORCS simulator where we conduct an ablation study to understand the effect different counterfactual predictive representations have on performance and comfort. Refer to the Appendix[2] for more details in the experimental setup and training.

A. Jackal Robot

The proposed solution for learning to drive the Jackal robot in the real-world is called GVF-BCQ since it combines our novel method of learning GVF predictions with BCQ [23]. Two baselines are compared with our method: (1) a classical controller using model predictive control (MPC), and (2) batch-constrained Q-learning that trains end-to-end (E2E-BCQ). The MPC uses a map and 2D laser scanner for localization from pre-existing ROS packages. The E2E-BCQ is the current state-of-the-art in offline deep RL [24]. Comparing to online RL was impractical for safety concerns and the need to recharge the robot's battery every 4 hours.

The training data consisted of 6 training roads in both counter clock-wise (CCW) and clock-wise (CW) directions and 3 test roads where each of the 3 test roads had damaged variants. All training data was flipped to simulate travelling in the reverse direction and balance the data set in terms of direction. The training data was collected using a diverse set of drivers including human drivers by remote control and a pure pursuit controller with safe exploration; thus, the training data was not suitable for imitation learning. The test roads were different from the training data: (1) a rectangle-shaped road with rounded outer corners, (2) an oval-shaped road,

[2]Appendix is at https://arxiv.org/abs/2103.08070

TABLE I: Comparison of GVF-BCQ (our method) and E2E-BCQ (baseline) on Rectangle test road with 0.4 m/s target speed in both the CW and CCW directions. GVF-BCQ exceeds performance of E2E-BCQ in all respects with higher overall speed, and far fewer out of lane events. E2E-BCQ was deemed unsafe for further experiments.

Method	Dir.	r/s ↑	Speed ↑	Off-center ↓	Off-angle ↓	Out of Lane ↓
GVF-BCQ	CCW	**2.68**	**0.32**	**0.14**	**0.13**	**0.0%**
E2E-BCQ	CCW	1.26	0.18	0.26	0.24	3.8%
GVF-BCQ	CW	**2.29**	**0.31**	**0.22**	**0.16**	**0.0%**
E2E-BCQ[3]	CW	-0.13	0.17	0.99	0.30	54.2%

TABLE II: Effect of damaged lanes on GVF-BCQ performance in CCW direction with 0.4 m/s target speed where R, O, and C are the Rectangle, Oval and Complex road shapes respectively. GVF-BCQ demonstrates robustness to damaged and distracting lanes.

	Damage	r/s ↑	Off-center ↓	Off-angle ↓	Out of Lane ↓	Speed Jerk ↓	Steer Jerk ↓
R	No	2.68	0.13	0.13	0.0%	0.036	0.23
	Yes	**2.74**	0.14	0.14	0.0%	-0.038	0.23
O	No	**2.40**	**0.28**	0.21	**1.5%**	0.035	0.22
	Yes	2.07	0.33	0.21	7.19%	0.033	0.21
C	No	**2.35**	**0.22**	**0.18**	**0.0%**	**0.034**	**0.23**
	Yes	2.11	0.31	0.24	9.42%	0.044	0.29

and (3) a complex road loop with many turns significantly different from anything observed by the agent during training. In addition, the test roads included variants with damaged lane markings. The reward is given by $r_t = v_t(\cos\beta_t + |\alpha_t|)$ where v_t is the speed of the vehicle in km/h, β_t is the angle between the road direction and the vehicle direction, and α_t is the lane centeredness.

A comparison of the learned approaches is given in Table I where GVF-BCQ approach exceeds the performance of E2E-BCQ in all respects demonstrating better performance at nearly double the speed. Both GVF-BCQ and E2E-BCQ were trained with the same data sets and given 10M updates each for a fair comparison. For GVF-BCQ, the first 5M updates were used for learning the counterfactual predictions and the second 5M updates were used for learning the policy from the predictive representation with BCQ. They both received the same observations consisting of two stacked images, current vehicle speed, and last action and produced desired steering angle and speed.

GVF-BCQ was tested on roads with damaged and distracting lane markings as shown in Table III. The damaged and distracting lane markings for the complex test road loop are shown in Figure 1. These results demonstrate robustness because the training data did not include roads with damaged or distracting lane markings.

GVF-BCQ was also compared to MPC in Table III where GVF-BCQ was found to produce superior performance in nearly all metrics at a high target speed of 0.4 m/s. The MPC performed poorly since it was difficult to tune for 0.4 m/s; performance was more similar at 0.25 m/s speeds where results are in the Appendix. A clear advantage of GVF-BCQ is the stability and smoothness of control achieved at the higher speeds.

[3]E2E-BCQ failed to recover after undershooting the first turn in the clock-wise (CW) direction; it was not safe for testing on the other roads.

TABLE III: Comparison of GVF-BCQ (our method) and MPC (baseline) in CCW direction with 0.4 m/s target speed where R, O, and C are the Rectangle, Oval and Complex road shapes respectively.

	Method	r/s ↑	Off-center ↓	Off-angle ↓	Out of Lane ↓	Speed Jerk ↓	Steer Jerk ↓
R	GVF-BCQ	**2.68**	**0.13**	**0.13**	**0.0%**	**0.036**	**0.23**
	MPC	0.97	0.53	0.19	20.4%	0.083	1.25
O	GVF-BCQ	**2.40**	**0.28**	0.21	**1.45%**	**0.035**	**0.22**
	MPC	0.89/s	0.53	**0.20**	22.7%	0.103	1.41
C	GVF-BCQ	**2.35**	**0.22**	**0.18**	**0.0%**	**0.034**	**0.23**
	MPC	0.72	0.64	0.21	38.9%	-0.063	-1.21

B. Ablation Study in TORCS

In order to understand the role of counterfactual predictions in representing the state of the agent, we conduct an ablation study in the TORCS simulator. We compare representations consisting of future predictions at multiple time scale, future predictions at a single time scale and predictions with supervised regression of the current (non-future) lane centeredness and relative road angle. These experiments were conducted with online RL using deep deterministic policy gradient (DDPG) [50] in order to more easily understand the impact of the different state representations on the learning process.

Our method is called GVF-DDPG and uses multiple time scales specified by the values $\gamma = [0.0, 0.5, 0.9, 0.95, 0.97]$. Two variants of our method called GVF-0.95-DDPG and GVF-0.0-DDPG were defined to investigate the impact of different temporal horizons on performance, where $\gamma = 0.95$ and $\gamma = 0.0$ respectively. It is worth pointing out that when $\gamma = 0$, the prediction is myopic meaning that it reduces to a standard supervised regression problem equivalent to the predictions learned in [37]. These methods receive a history of two images, velocity and last action and produce desired steering angle and vehicle speed action commands.

Some additional baselines include a kinematic-based steering approach based on [54] and two variants of DDPG with slightly different state representations. The kinematic-based steering approach is treated as a "ground truth" controller since it has access to perfect localization information to steer the vehicle; unlike our approach, the speed is controlled independently. The variants of DDPG are called (1) DDPG-Image and (2) DDPG-LowDim. DDPG-Image is given a history of two images, velocity and last action while DDPG-LowDim is given a history of two images, velocity, last action, current lane centeredness α and relative road angle β in the observation. Both DDPG-Image and DDPG-LowDim output steering angle and vehicle speed action commands. The performance of DDPG-LowDim serves as an ideal learned

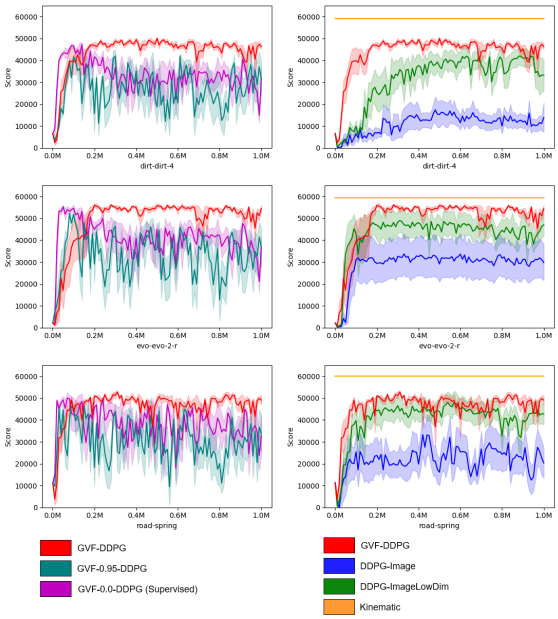


Fig. 5: Ablation study of GVF-DDPG (our method) of test scores (accumulated reward) over different time scale selections (left) and raw image-based state representations (right). Test scores were evaluated every 1000 steps during training for dirt-dirt-4, evo-evo-2 and road-spring which were not part of the training set. Results show our proposed predictive representation with multiple time scales achieves the best performance.

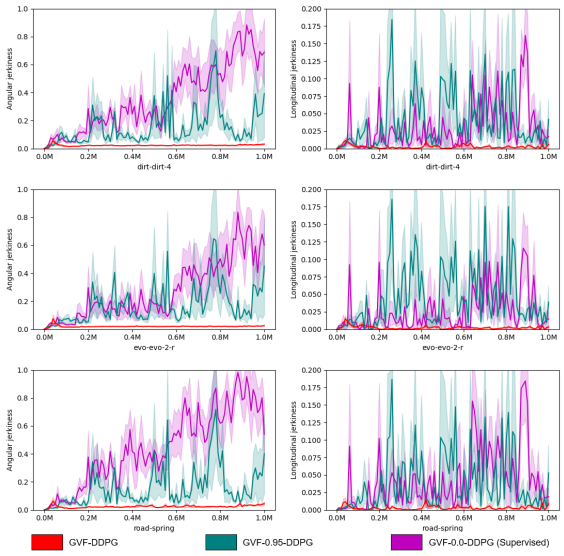


Fig. 6: Ablation study of GVF-DDPG (our method) of jerkiness (lower is better) over different time scale selections. We use angular and longitudinal jerkiness to evaluate the smoothness of the learned policy. The jerkiness is evaluated every 1000 steps during training for dirt-dirt-4, evo-evo-2 and road-spring which were not part of the training set. Results show our proposed multi-time-scale predictions achieves the best performance.

controller since it learns from both images and the perfect localization information.

The learned agents were trained on 85% of 40 tracks available in TORCS. The rest of the tracks were used for testing (6 in total) to measure the generalization performance of the policies. Results are repeated over 5 runs for each method. Only three of the tracks were successfully completed by at least one learned agent and those are reported here. The reward in the TORCS environment is given by $r_t = 0.0002 v_t(\cos\beta_t + |\alpha_t|)$ where v_t is the speed of the vehicle in km/h, β_t is the angle between the road direction and the vehicle direction, and α_t is the current lane centeredness. The policies were evaluated on test roads at regular intervals during training as shown in Figures 5 and 6.

The GVF-0.0-DDPG and GVF-0.95-DDPG variations initially learned very good solutions but then diverged indicating that one prediction may not be enough to control both steering angle and vehicle speed. Despite an unfair advantage provided by DDPG-LowDim with the inclusion of lane centeredness and road angle in the observation vector, GVF-DDPG still outperforms both variants of DDPG on many of the test roads. DDPG-Image was challenging to tune and train due to instability in learning; however, the counterfactual predictions in GVF-DDPG stabilized training for more consistent learning even though they were being learned simultaneously. Only GVF-DDPG with multiple time scale predictions is able to achieve extraordinarily smooth control.

VI. Conclusions

We present a new approach to learning to drive through a two step process: (1) learn a limited number of counterfactual predictions about future lane centeredness and road angle under a known policy, and (2) learn an RL policy using the counterfactual predictions as a representation of state. Our novel approach is safe and practical because it learns from real-world driving data without online exploration where the behavior distribution of the driving data is unknown. An experimental investigation into the impact of predictive representations on learning good driving policies shows that they generalize well to new roads, damaged lane markings and even distracting lane markings. We find that our approach improves the performance, smoothness and robustness of the driving decisions from images. We conclude that counterfactual predictions at different time scales is crucial to achieve a good driving policy. To the best of our knowledge, this is the first practical demonstration of deep RL applied to autonomous driving on a real vehicle using only real-world data without any online exploration.

Our approach has the potential to be scaled with large volumes of data captured by human drivers of all skill levels; however, more work is needed to understand how well this approach will scale. In addition, a general framework of learning the right counterfactual predictions for real-world problems is needed where online interaction is prohibitively expensive.

REFERENCES

[1] N. Möhler, D. John, and M. Voigtländer, "Lane detection for a situation adaptive lane keeping support system, the safelane system," in *Advanced Microsystems for Automotive Applications 2006*, J. Valldorf and W. Gessner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 485–500.

[2] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 41–54, 2020.

[3] T. Ort, L. Paull, and D. Rus, "Autonomous vehicle navigation in rural environments without detailed prior maps," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2040–2047.

[4] Bing-Fei Wu, Tsu-Tian Lee, Hsin-Han Chang, Jhong-Jie Jiang, Cheng-Nan Lien, Tien-Yu Liao, and Jau-Woei Perng, "Gps navigation based autonomous driving system design for intelligent vehicles," in *IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 3294–3299.

[5] G. Garimella, J. Funke, C. Wang, and M. Kobilarov, "Neural network modeling for steering control of an autonomous vehicle," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2609–2615.

[6] R. Liu, J. Wang, and B. Zhang, "High definition map for automated driving: Overview and analysis," *Journal of Navigation*, vol. 73, no. 2, p. 324–341, 2020.

[7] L. Wang, Y. Zhang, and J. Wang, "Map-based localization method for autonomous vehicles using 3d-lidar," *IFAC*, vol. 50, no. 1, pp. 276 – 281, 2017.

[8] J. Chen, B. Yuan, and M. Tomizuka, "Deep imitation learning for autonomous driving in generic urban scenarios with enhanced safety," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2884–2890, 2019.

[9] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016. [Online]. Available: http://arxiv.org/abs/1604.07316

[10] Z. Chen and X. Huang, "End-to-end learning for lane keeping of self-driving cars," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1856–1860.

[11] A. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 2017, pp. 70–76, 2017.

[12] L. Chi and Y. Mu, "Deep steering: Learning end-to-end driving model from spatial and temporal visual cues," *CoRR*, vol. abs/1708.03798, 2018. [Online]. Available: http://arxiv.org/abs/1810.00123

[13] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. A. Theodorou, and B. Boots, "Imitation learning for agile autonomous driving," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 286–302, 2020.

[14] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3389–3396.

[15] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *CoRR*, vol. abs/2002.00444, 2020. [Online]. Available: http://arxiv.org/abs/2002.00444

[16] G. Dulac-Arnold, D. J. Mankowitz, and T. Hester, "Challenges of real-world reinforcement learning," *International Conference on International Conference on Machine Learning*, vol. abs/1904.12901, 2019. [Online]. Available: http://arxiv.org/abs/1904.12901

[17] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–503, 2016.

[18] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas *et al.*, "Solving rubik's cube with a robot hands," *CoRR*, vol. abs/1910.07113, 2019. [Online]. Available: http://arxiv.org/abs/1910.07113

[19] S. Whiteson, B. Tanner, M. E. Taylor, and P. Stone, "Protecting against evaluation overfitting in empirical reinforcement learning," *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pp. 120–127, 2011.

[20] C. Zhao, O. Sigaud, F. Stulp, and T. M. Hospedales, "Investigating generalisation in continuous deep reinforcement learning," *CoRR*, vol. abs/1902.07015, 2019. [Online]. Available: https://arxiv.org/abs/1902.07015

[21] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," *CoRR*, vol. abs/1709.06560, 2017. [Online]. Available: http://arxiv.org/abs/1709.06560

[22] J. Farebrother, M. C. Machado, and M. Bowling, "Generalization and regularization in DQN," *CoRR*, vol. abs/1810.00123, 2018. [Online]. Available: http://arxiv.org/abs/1810.00123

[23] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," *CoRR*, vol. abs/1812.02900, 2018. [Online]. Available: http://arxiv.org/abs/1812.02900

[24] S. Fujimoto, E. Conti, M. Ghavamzadeh, and J. Pineau, "Benchmarking batch deep reinforcement learning algorithms," *CoRR*, vol. abs/1910.01708, 2019. [Online]. Available: http://arxiv.org/abs/1910.01708

[25] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: tutorial, review and perspectives on open problems," *CoRR*, vol. abs/2005.01643, 2020. [Online]. Available: http://arxiv.org/abs/2005.01643

[26] R. Sutton, J. Modayil, M. Delp, T. Degris, P. Pilarski, A. White, and D. Precup, "Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction," in *International Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '11, vol. 2, 2011, pp. 761–768.

[27] J. Modayil, A. White, and R. S. Sutton, "Multi-timescale nexting in a reinforcement learning robot," in *From Animals to Animats 12*, T. Ziemke, C. Balkenius, and J. Hallam, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 299–309.

[28] A. Clark, "Whatever next? predictive brains, situated agents, and the future of cognitive science," *Behavioral and Brain Science*, vol. 36, no. 3, pp. 181–204, 2013.

[29] E. M. Russek, I. Momennejad, M. M. Botvinick, S. J. Gershman, and N. D. Daw, "Predictive representations can link model-based reinforcement learning to model-free mechanisms," *PLOS Computational Biology*, vol. 13, no. 9, pp. 1–35, 2017.

[30] D. D. Salvucci and R. Gray, "A two-point visual control model of steering," *Perception*, vol. 33, no. 10, pp. 1233–1248, 2004.

[31] N. Kapania and J. Gerdes, "Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling," *Vehicle System Dynamics*, vol. 53, pp. 1–18, 2015.

[32] C. Beal and J. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *Control Systems Technology, IEEE Transactions on*, vol. 21, pp. 1258–1269, 2013.

[33] M. L. Littman and R. S. Sutton, "Predictive representations of state," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., 2002, pp. 1555–1561.

[34] E. J. Rafols, M. B. Ring, R. S. Sutton, and B. Tanner, "Using predictive representations to improve generalization in reinforcement learning," in *International Joint Conference on Artificial Intelligence*, ser. IJCAI'05, 2005, pp. 835–840.

[35] T. Schaul and M. Ring, "Better generalization with forecasts," in *International Joint Conference on Artificial Intelligence*, ser. IJCAI '13, 2013, pp. 1656–1662.

[36] M. Bansal, A. Krizhevsky, and A. S. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," in *Robotics: Science and Systems XV, University of Freiburg, Freiburg im Breisgau, Germany, June 22-26, 2019*, A. Bicchi, H. Kress-Gazit, and S. Hutchinson, Eds., 2019. [Online]. Available: https://doi.org/10.15607/RSS.2019.XV.031

[37] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2722–2730.

[38] P. S. Thomas and E. Brunskill, "Data-efficient off-policy policy evaluation for reinforcement learning," in *International Conference on International Conference on Machine Learning*, ser. ICML'16, vol. 48, 2016, p. 2139–2148.

[39] J. Cunha, R. Serra, N. Lau, L. Lopes, and A. Neves, "Batch reinforcement learning for robotic soccer using the q-batch update-rule," *Journal of Intelligent & Robotic Systems*, vol. 80, pp. 385–399, 2015.

[40] J. Günther, P. M. Pilarski, G. Helfrich, H. Shen, and K. Diepold, "Intelligent laser welding through representation, prediction, and control learning: An architecture with deep neural networks and reinforcement learning," *Mechatronics*, vol. 34, pp. 1 – 11, 2016.

[41] A. L. Edwards, M. R. Dawson, J. S. Hebert, C. Sherstan, R. S. Sutton, K. M. Chan, and P. M. Pilarski, "Application of real-time machine learning to myoelectric prosthesis control: A case series in adaptive switching," *Prosthetics and orthotics international*, vol. 40, no. 5, pp. 573–581, 2016.

[42] A. White, "Developing a predictive approach to knowledge," Ph.D. dissertation, University of Alberta, 2015.

[43] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, "Reinforcement learning with unsupervised auxiliary tasks." *International Conference on Learning Representations*, 2017.

[44] A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. P. van Hasselt, and D. Silver, "Successor features for transfer in reinforcement learning," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, pp. 4055–4065.

[45] H. Van Seijen, M. Fatemi, J. Romoff, R. Laroche, T. Barnes, and J. Tsang, "Hybrid reward architecture for reinforcement learning," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5392–5402.

[46] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602

[47] M. Schlegel, W. Chung, D. Graves, J. Qian, and M. White, "Importance resampling off-policy prediction," in *Neural Information Processing Systems*, ser. NeurIPS'19, 2019.

[48] S. Ghiassian, A. Patterson, M. White, R. S. Sutton, and A. White, "Online off-policy prediction," *CoRR*, vol. abs/1811.02597, 2018. [Online]. Available: http://arxiv.org/abs/1811.02597

[49] D. Graves, K. Rezaee, and S. Scheideman, "Perception as prediction using general value functions in autonomous driving applications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, ser. IROS 2019, 2019.

[50] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International Conference on International Conference on Machine Learning*, ser. ICML'14, vol. 32, 2014, pp. I–387–I–395.

[51] M. Sugiyama, T. Suzuki, and T. Kanamori, "Density ratio estimation: A comprehensive review," *RIMS Kokyuroku*, pp. 10–31, 2010.

[52] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988.

[53] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *International Conference on Learning Representations*, Puerto Rico, 2016.

[54] B. Paden, M. Cáp, S. Z. Yong, D. S. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *CoRR*, vol. abs/1604.07446, 2016. [Online]. Available: http://arxiv.org/abs/1604.07446