Learning Robust Control Policies for End-to-End Autonomous Driving from Data-Driven Simulation

Alexander Amini[1], Igor Gilitschenski[1], Jacob Phillips[1], Julia Moseyko[1], Rohan Banerjee[1], Sertac Karaman[2], Daniela Rus[1]

***Abstract*—In this work, we present a data-driven simulation and training engine capable of learning end-to-end autonomous vehicle control policies using only sparse rewards. By leveraging real, human-collected trajectories through an environment, we render novel training data that allows virtual agents to drive along a continuum of new local trajectories consistent with the road appearance and semantics, each with a different view of the scene. We demonstrate the ability of policies learned within our simulator to generalize to and navigate in previously unseen real-world roads, without access to any human control labels during training. Our results validate the learned policy onboard a full-scale autonomous vehicle, including in previously un-encountered scenarios, such as new roads and novel, complex, near-crash situations. Our methods are scalable, leverage reinforcement learning, and apply broadly to situations requiring effective perception and robust operation in the physical world.**

***Index Terms*—Deep Learning in Robotics and Automation, Autonomous Agents, Real World Reinforcement Learning, Data-Driven Simulation**

I. Introduction

END-TO-END (i.e., perception-to-control) trained neural networks for autonomous vehicles have shown great promise for lane stable driving [1]–[3]. However, they lack methods to learn robust models at scale and require vast amounts of training data that are time consuming and expensive to collect. Learned end-to-end driving policies and modular perception components in a driving pipeline require capturing training data from all necessary edge cases, such as recovery from off-orientation positions or even near collisions. This is not only prohibitively expensive, but also potentially dangerous [4]. Training and evaluating robotic controllers in simulation [5]–[7] has emerged as a potential solution to the need for more data and increased robustness to novel situations, while also avoiding the time, cost, and safety issues of current methods. However, transferring policies learned in simulation into the real-world still remains an open research challenge. In this paper, we present an end-to-end simulation and training engine capable of training real-world reinforcement learning (RL) agents entirely in simulation, without any prior knowledge of human driving or post-training fine-tuning. We demonstrate trained models can then be deployed directly in the real world, on roads and environments not encountered in training. Our engine, termed **VISTA**: *Virtual Image Synthesis and Transformation for Autonomy*, synthesizes a continuum of driving trajectories that are photorealistic and semantically faithful to their respective real world driving conditions (Fig. 1), from a small dataset of human collected driving trajectories. **VISTA** allows a virtual agent to not only observe a stream of sensory data from stable driving (i.e., human collected driving data), but also from a simulated band of new observations from off-orientations on the road. Given visual observations of the environment (i.e., camera images), our system learns a lane-stable control policy over a wide variety of different road and environment types, as opposed to current end-to-end systems [2], [3], [8], [9] which only imitate human behavior. This is a major advancement as there does not currently exist a scalable method for training autonomous vehicle control policies that go beyond imitation learning and can generalize to and navigate in previously unseen road and

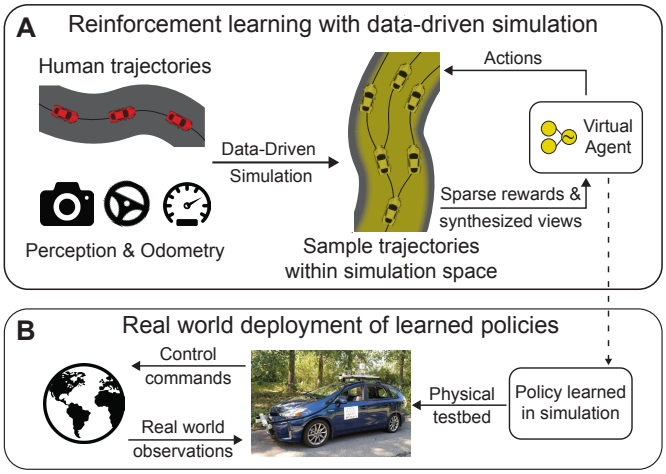

Fig. 1. **Training and deployment of policies from data-driven simulation.** From a single human collected trajectory our data-driven simulator (**VISTA**) synthesizes a space of new possible trajectories for learning virtual agent control policies (A). Preserving photorealism of the real world allows the virtual agent to move beyond imitation learning and instead explore the space using reinforcement learning with only sparse rewards. Learned policies not only transfer directly to the real world (B), but also outperform state-of-the-art end-to-end methods trained using imitation learning.

Manuscript received: September, 10, 2019; Accepted January, 13, 2020; date of current version January 30, 2020. This letter was recommended for publication by Associate Editor E. E. Aksoy and Editor T. Asfour upon evaluation of the reviewers comments. This work was supported in part by National Science Foundation (NSF), in part by Toyota Research Institute (TRI) and in part by NVIDIA Corporation. *Corresponding author: Alexander Amini*

[1] Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139 {amini, igilitschenski, jdp99, jmoseyko, rohanb, rus}@mit.edu

[2] Laboratory for Information and Decision Systems, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139 {sertac}@mit.edu

Digital Object Identifier 10.1109/LRA.2020.2966414

complex, near-crash situations.

By synthesizing training data for a broad range of vehicle positions and orientations from real driving data, the engine is capable of generating a continuum of novel trajectories consistent with that road and learning policies that transfer to other roads. This variety ensures agent policies learned in our simulator benefit from autonomous exploration of the feasible driving space, including scenarios in which the agent can recover from near-crash off-orientation positions. Such positions are a common edge-case in autonomous driving and are difficult and dangerous to collect training data for in the real-world. We experimentally validate that, by experiencing such edge cases within our synthesized environment during training, these agents exhibit greater robustness in the real-world and recover approximately two times more frequently compared to state-of-the-art imitation learning algorithms.

In summary, the key contributions of this paper can be summarized as:

1) **VISTA**, a photorealistic, scalable, data-driven simulator for synthesizing a continuum of new perceptual inputs locally around an existing dataset of stable human collected driving data;
2) An end-to-end learning pipeline for training autonomous lane-stable controllers using only visual inputs and sparse reward signals, without explicit supervision using ground truth human control labels; and
3) Experimental validation that agents trained in **VISTA** can be deployed directly in the real-world and achieve more robust recovery compared to previous state-of-the-art imitation learning models.

To the best of our knowledge, this work is the first published report of a full-scale autonomous vehicle trained entirely in simulation using only reinforcement learning, that is capable of being deployed onto real roads and recovering from complex, near crash driving scenarios.

II. Related Work

Training agents in simulation capable of robust generalization when deployed in the real world is a long-standing goal in many areas of robotics [9]–[12]. Several works have demonstrated transferable policy learning using domain randomization [13] or stochastic augmentation techniques [14] on smaller mobile robots. In autonomous driving, end-to-end trained controllers learn from raw perception data, as opposed to maps [15] or other object representations [16]–[18]. Previous works have explored learning with expert information for lane following [1], [2], [19], [20], full point-to-point navigation [3], [8], [21], and shared human-robot control [22], [23], as well as in the context of RL by allowing the vehicle to repeatedly drive off the road [4]. However, when trained using state-of-the-art model-based simulation engines, these techniques are unable to be directly deployed in real-world driving conditions.

Performing style transformation, such as adding realistic textures to synthetic images with deep generative models, has been used to deploy learned policies from model-based simulation engines into the real world [9], [24]. While these approaches can successfully transfer low-level details such as textures or sensory noise, these approaches are unable to transfer higher-level semantic complexities (such as vehicle or pedestrian behaviors) present in the real-world that are also required to train robust autonomous controllers. Data-driven engines like *Gibson* [25] and *FlightGoggles* [26] render photorealistic environments using photogrammetry, but such closed-world models are not scalable to the vast exploration space of all roads and driving scenarios needed to train for real world autonomous driving. Other simulators [27] face scalability constraints as they require ground truth semantic segmentation and depth from expensive LIDAR sensors during collection.

The novelty of our approach is in leveraging sparsely-sampled trajectories from human drivers to synthesize training data sufficient for learning end-to-end RL policies robust enough to transfer to previously unseen real-world roads and to recover from complex, near crash scenarios.

III. Data-Driven Simulation

Simulation engines for training robust, end-to-end autonomous vehicle controllers must address the challenges of photorealism, real-world semantic complexities, and scalable exploration of control options, while avoiding the fragility of imitation learning and preventing unsafe conditions during data collection, evaluation, and deployment. Our data-driven simulator, **VISTA**, synthesizes photorealistic and semantically accurate local viewpoints as a virtual agent moves through the environment (Fig. 2). **VISTA** uses a repository of sparsely sampled trajectories collected by human drivers. For each trajectory through a road environment, **VISTA** synthesizes views that allow virtual agents to drive along an infinity of new local trajectories consistent with the road appearance and semantics, each with a different view of the scene.

Upon receiving an observation of the environment at time t, the agent commands a desired steering curvature, κ_t, and velocity, v_t to execute at that instant until the next observation. We denote the time difference between consecutive observations as Δt. **VISTA** maintains an internal state of each agent's position, (x_t, y_t), and angular orientation, θ_t, in a global reference frame. The goal is to compute the new state of the agent at time, $t + \Delta t$, after receiving the commanded steering curvature and velocity. First, **VISTA** computes the changes in state since the last timestep,

$$\begin{aligned} \Delta\theta &= |v_t \cdot \Delta t| \cdot \kappa_t, \\ \Delta\hat{x} &= (1 - \cos(\Delta\theta))/\kappa_t, \\ \Delta\hat{y} &= \sin(\Delta\theta)/\kappa_t. \end{aligned} \tag{1}$$

VISTA updates the global state, taking into account the change in the agent's orientation, by applying a 2D rotational matrix before updating the position in the global frame,

$$\begin{aligned} \theta_{t+\Delta t} &= \theta_t + \Delta\theta, \\ \begin{bmatrix} x_{t+\Delta t} \\ y_{t+\Delta t} \end{bmatrix} &= \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \begin{bmatrix} \cos(\theta_{t+\Delta t}) & -\sin(\theta_{t+\Delta t}) \\ \sin(\theta_{t+\Delta t}) & \cos(\theta_{t+\Delta t}) \end{bmatrix} \begin{bmatrix} \Delta\hat{x} \\ \Delta\hat{y} \end{bmatrix}. \end{aligned} \tag{2}$$

This process is repeated for both the virtual agent who is navigating the environment and the replayed version of the

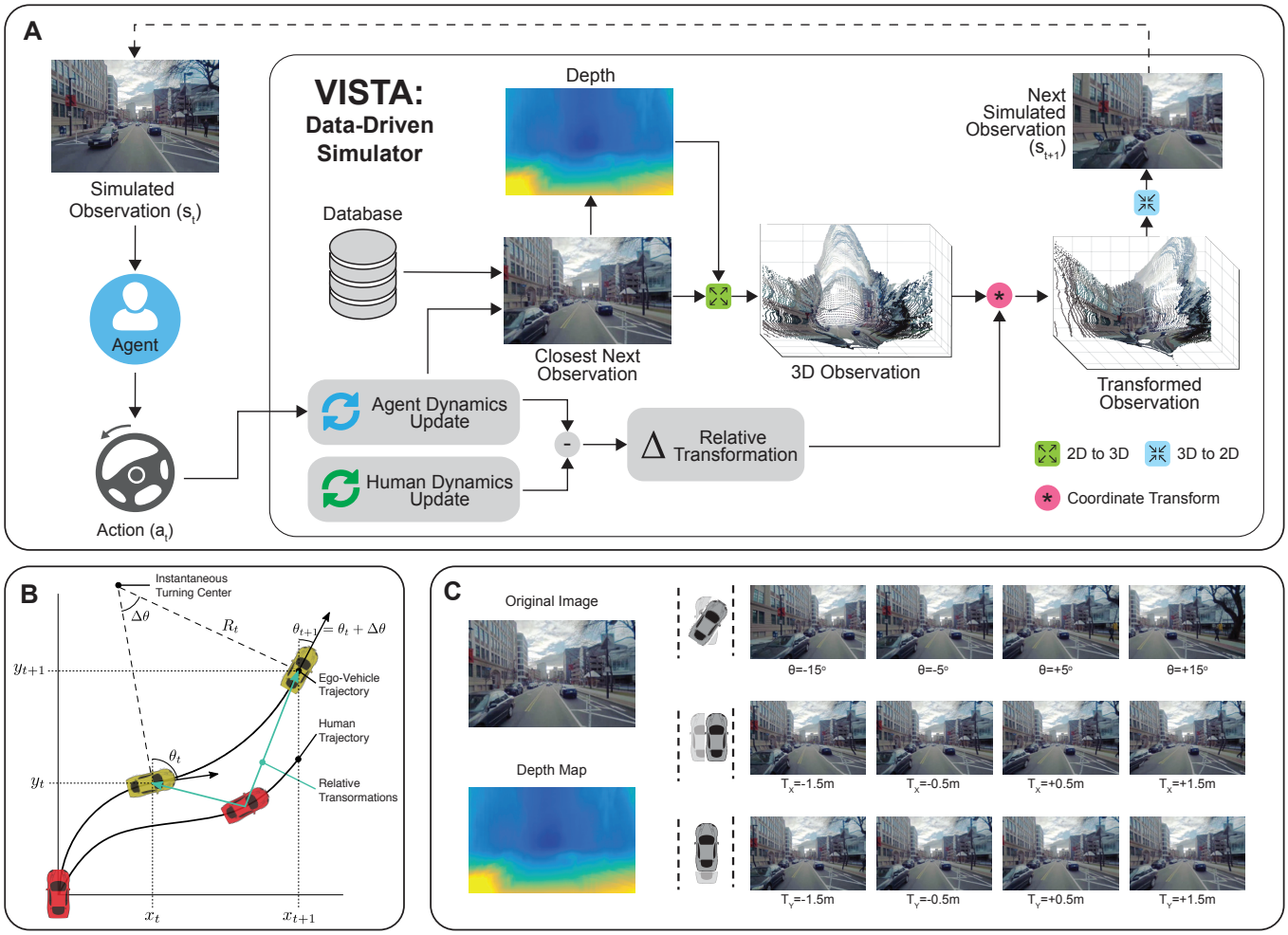

Fig. 2. **Simulating novel viewpoints for learning.** Schematic of an autonomous agent's interaction with the data-driven simulator (A). At time step, t, the agent receives an observation of the environment and commands an action to execute. Motion is simulated in **VISTA** and compared to the human's estimated motion in the real world (B). A new observation is then simulated by transforming a 3D representation of the scene into the virtual agent's viewpoint (C).

human who drove through the environment in the real world. Now in a common coordinate frame, **VISTA** computes the relative displacement by subtracting the two state vectors. Thus, **VISTA** maintains estimates of the lateral, longitudinal, and angular perturbations of the virtual agent with respect to the closest human state at all times (cf. Fig. 2B).

VISTA is scalable as it does not require storing and operating on 3D reconstructions of entire environments or cities. Instead, it considers only the observation collected nearest to the virtual agent's current state. Simulating virtual agents over real road networks spanning thousands of kilometers requires several hundred gigabytes of monocular camera data. Fig. 2C presents view synthesis samples. From the single closest monocular image, a depth map is estimated using a convolutional neural network using self-supervision of stereo cameras [28]. Using the estimated depth map and camera intrinsics, our algorithm projects from the sensor frame into the 3D world frame. After applying a coordinate transformation to account for the relative transformation between virtual agent and human, the algorithm projects back into the sensor frame of the vehicle and returns the result to the agent as its next observation. To allow some movement of the virtual agent within the **VISTA** environment, we project images back into a smaller field-of-view than the collected data (which starts at 120°). Missing pixels are inpainted using a bilinear sampler, although we acknowledge more photorealistic, data-driven approaches [29] that could also be used. **VISTA** is capable of simulating different local rotations ($\pm 15°$) of the agent as well as both lateral and longitudinal translations ($\pm 1.5\text{m}$) along the road. As the free lateral space of a vehicle within its lane is typically less than 1m, **VISTA** can simulate beyond the bounds of lane-stable driving. Note that while we focus on data-driven simulation for lane-stable driving in this work, the presented approach is also applicable to end-to-end navigation [3] learning by stitching together collected trajectories to learn through arbitrary intersection configurations.

A. *End-to-End Learning*

All controllers presented in this paper are learned end-to-end, directly from raw image pixels to actuation. We considered controllers that act based on their current perception without memory or recurrence built in, as suggested in [2],

[16]. Features are extracted from the image using a series of convolutional layers into a lower dimensional feature space, and then through a set of fully connected layers to learn the final control actuation commands. Since all layers are fully differentiable, the model was optimized entirely end-to-end. As in previous work [2], [3], we learn lateral control by predicting the desired curvature of motion. Note that curvature is equal to the inverse turning radius [m^{-1}] and can be converted to steering angle at inference time using a bike model [30], assuming minimal slip.

Formally, given a dataset of n observed state-action pairs $(s_t, a_t)_{i=1}^{n}$ from human driving, we aim to learn an autonomous policy parameterized by $\boldsymbol{\theta}$ which estimates $\hat{a}_t = f(s_t; \boldsymbol{\theta})$. In supervised learning, the agent outputs a *deterministic* action by minimizing the empirical error,

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} (f(s_t; \boldsymbol{\theta}) - a_t)^2. \tag{3}$$

However, in the RL setting, the agent has no explicit feedback of the human actuated command, a_t. Instead, it receives a reward r_t for every consecutive action that does not result in an intervention and can evaluate the return, R_t, as the discounted, accumulated reward

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{4}$$

where $\gamma \in (0, 1]$ is a discounting factor. In other words, the return that the agent receives at time t is a discounted distance traveled between t and the time when the vehicle requires an intervention. As opposed to in supervised learning, the agent optimizes a *stochastic* policy over the space of all possible actions: $\pi(a|s_t; \boldsymbol{\theta})$. Since the steering control of autonomous vehicles is a continuous variable, we parameterize the output probability distribution at time t as a Gaussian, (μ_t, σ_t^2). Therefore, the policy gradient, $\nabla_{\boldsymbol{\theta}} \pi(a|s_t; \boldsymbol{\theta})$, of the agent can be computed analytically:

$$\nabla_{\boldsymbol{\theta}} \pi(a|s_t; \boldsymbol{\theta}) = \pi(a|s_t; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \log \left(\pi(a|s_t; \boldsymbol{\theta})\right) \tag{5}$$

Thus, the weights $\boldsymbol{\theta}$ are updating in the direction $\nabla_{\boldsymbol{\theta}} \log \left(\pi(a|s_t; \boldsymbol{\theta})\right) \cdot R_t$ during training [31], [32].

We train RL agents in various simulated environments, where they only receive rewards based on how far they can drive without intervention. Compared to supervised learning, where agents learn to simply imitate the behavior of the human driver, RL in simulation allows agents to learn suitable actions which maximize their total reward in that particular situation. Thus, the agent has no knowledge of how the human drove in that situation. Using only the feedback from interventions in simulation, the agent learns to optimize its own policy and thus to drive longer distances (Alg. 1).

We define a learning episode in **VISTA** as the time the agent starts receiving sensory observations to the moment it exits its lane boundaries. Assuming the original data was collected at approximately the center of the lane, this corresponds to declaring the end of an episode as when the lateral translation of the agent exceeds $\pm 1\mathrm{m}$.

Fig. 3. **Training images from various comparison methods.** Samples drawn from the real-world, **IMIT-AUG** (A) and CARLA (B-C). Domain randomization **DR-AUG** (C) illustrates a single location for comparison.

Algorithm 1 Policy Gradient (PG) training in **VISTA**

```
Initialize θ                                              ▷ NN weights
Initialize D ← 0                                          ▷ Single episode distance
while D < 10km do
    s_t ← VISTA.reset()
    while VISTA.done = False do
        a_t ~ π(s_t; θ)                                   ▷ Sample action
        s_{t+1} ← VISTA.step(a_t)                         ▷ Update state
        r_t ← 0.0 if VISTA.done else 1.0                  ▷ Reward
    end while
    D ← VISTA.episode_distance
    R_t ← Σ_{k=1}^{T} γ^k r_{t+k}                         ▷ Discounted return
    θ ← θ + η Σ_{t=1}^{T} ∇_θ log π(a_t|s_t; θ) R_t       ▷ Update
end while
return θ
```

Upon traversing a road successfully, the agent is transported to a new location in the dataset. Thus, training is not limited to only long roads, but can also occur on multiple shorter roads. An agent is said to sufficiently learn an environment once it successfully drives for 10km without interventions.

IV. Baselines

In this subsection, we discuss the evaluated baselines. The same input data formats (camera placement, field-of-view, and resolution) were used for both IL and RL training. Furthermore, model architectures for all baselines were equivalent with the exception of only the final layer in RL.

A. Real-World: Imitation Learning

Using real-world images (Fig. 3A) and control we benchmark models trained with end-to-end imitation learning (**IMIT-AUG**). Augmenting learning with views from synthetic side cameras [2], [20], [33] is the standard approach to increase robustness and teach the model to recover from off-center positions on the roads. We employ the techniques presented in [2], [20] to compute the recovery correction signal that should be trained with given these augmented inputs.

B. Model-Based Simulation: Sim-to-Real

We use the CARLA simulator [34] for evaluating the performance of end-to-end models using sim-to-real transfer learning techniques. As opposed to our data-driven simulator, CARLA, like many other autonomous driving simulators, is model-based. While tremendous effort has been placed into making the CARLA environment (Fig. 3B) as photorealistic as possible, a simulation gap still exists. We found that end-to-end models trained solely in CARLA were unable to transfer to the real-world. Therefore, we evaluated the following two techniques for bridging the sim-to-real gap in CARLA.

Domain Randomization. First, we test the effect of domain randomization (DR) [13] on learning within CARLA. DR attempts to expose the learning agent to many different random variations of the environment, thus increasing its robustness in the real-world. In our experiments, we randomized various properties throughout the CARLA world (Fig. 3C), including the sun position, weather, and hue of each of the semantic classes (i.e. road, lanes, buildings, etc). Like **IMIT-AUG** we also train CARLA DR models with viewpoint augmentation and thus, refer to these models as **DR-AUG**.

Domain Adaptation. We evaluate a model that is trained with both simulated and real images to learn shared control. Since the latent space between the two domains is shared [9], the model can output a control from real images during deployment even though it was only trained with simulated control labels during training. Again, viewpoint augmentation is used when training our sim-to-real baseline, **S2R-AUG**.

C. Expert Human

A human driver (**HUMAN**) drives the designed route as close to the center of the lane as possible, and is used to fairly evaluate and compare against all other learned models.

V. Results

A. Real-World Testbed

Learned controllers were deployed directly onboard a full-scale autonomous vehicle (2015 Toyota Prius V) which we retrofitted for full autonomous control [35]. The primary perception sensor for control is a LI-AR0231-GMSL camera (120 degree field-of-view), operating at 15Hz. Data is serialized with h264 encoding with a resolution of 19201208. At inference time, images are scaled down approximately 3 fold for performance. Also onboard are inertial measurement units (IMUs), wheel encoders, and a global positioning satellite (GPS) sensor for evaluation as well as an NVIDIA PX2 for computing. To standardize all model trials on the test-track, a constant desired speed of the vehicle was set at 20 kph, while the model commanded steering.

The model's generalization performance was evaluated on previously unseen roads. That is, the real-world training set contained none of the same areas as the testing track (spanning over 3km) where the model was evaluated.

Agents were evaluated on all roads in the test environment. The track presents a difficult rural test environment, as it does not have any clearly defined road boundaries or lanes. Cracks, where vegetation frequently grows onto the road, as well as

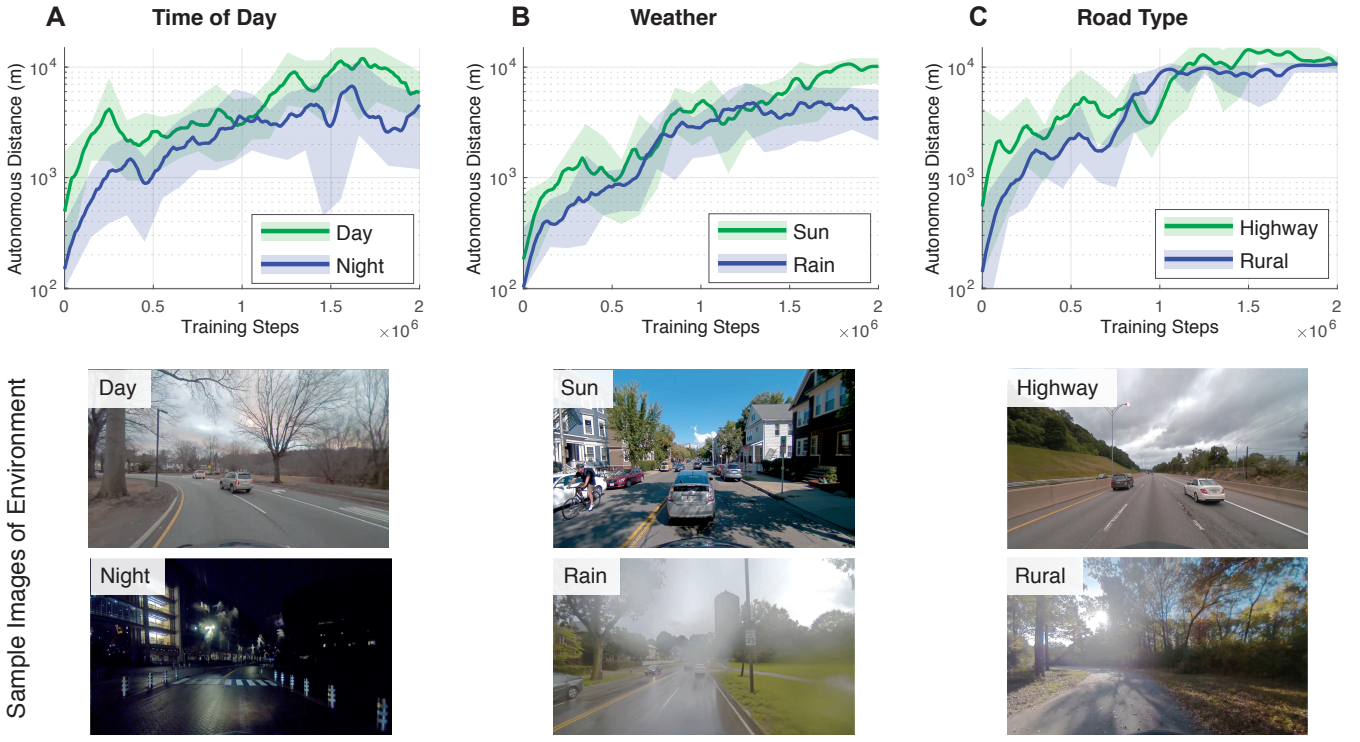

Fig. 4. **Reinforcement learning in simulation.** Autonomous vehicles placed in the simulator with no prior knowledge of human driving or road semantics demonstrate the ability to learn and optimize their own driving policy under various different environment types. Scenarios range from different times of day (A), to weather condition (B), and road types (C).

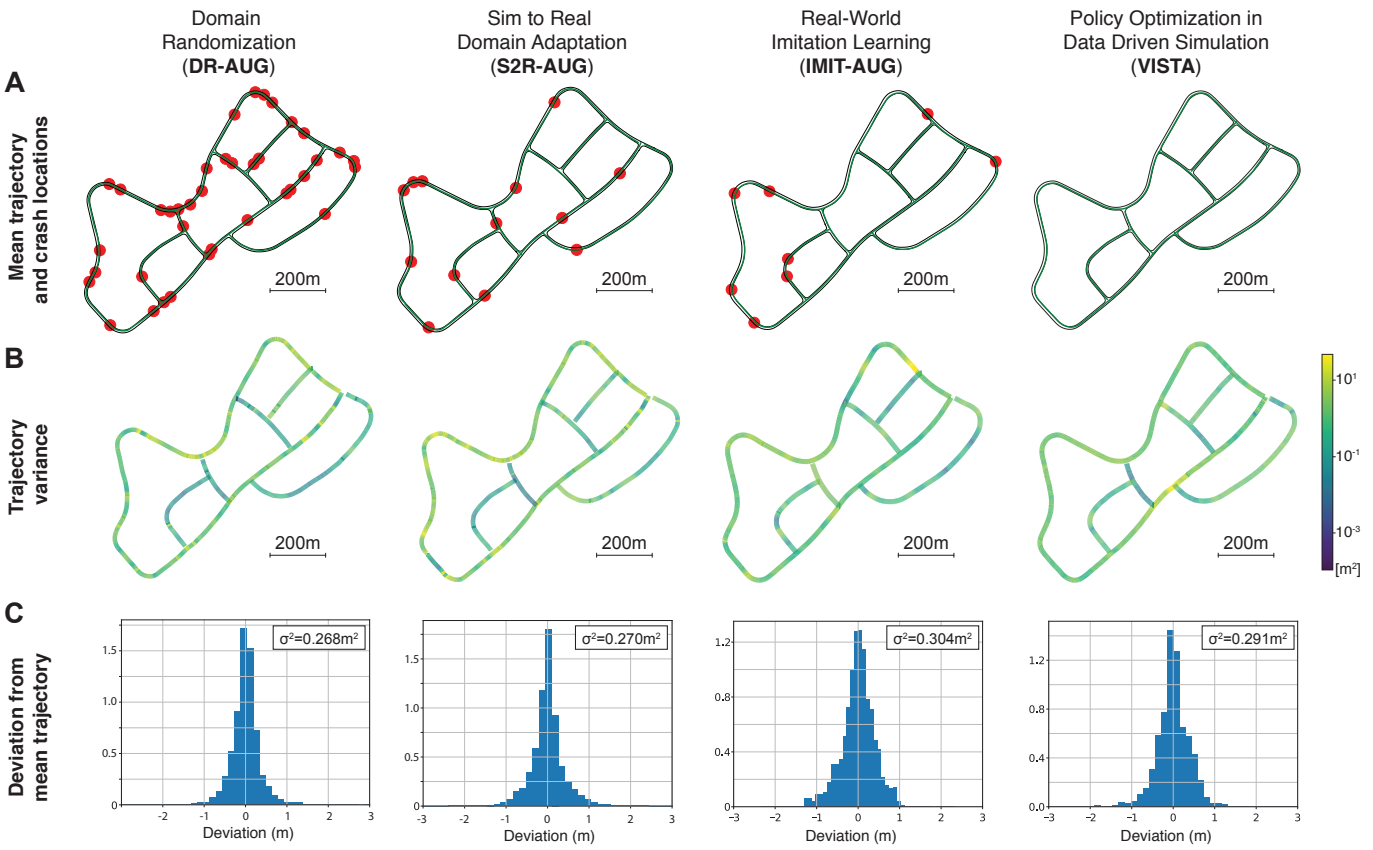

Fig. 5. **Evaluation of end-to-end autonomous driving.** Comparison of simulated domain randomization [13] and adaptation [9] as well as real-world imitation learning [2] to learning within **VISTA** (left-to-right). Each model is tested 3 times at fixed speeds on every road on the test track (A), with interventions marked as red dots. The variance between runs (B) and the distribution of deviations from the mean trajectory (C) illustrate model consistency.

strong shadows cast from surrounding trees, cause classical road detection algorithms to fail.

B. Reinforcement Learning in VISTA

In this section, we present results on learning end-to-end control of autonomous vehicles entirely within **VISTA**, under different weather conditions, times of day, and road types. Each environment collected for this experiment consisted of, on average, one hour of driving data from that scenario.

We started by learning end-to-end policies in different times of day (Fig. 4A) and, as expected, found that agents learned more quickly during the day than at night, where there was often limited visibility of lane markers and other road cues. Next, we considered changes in the weather conditions. Environments were considered "rainy" when there was enough water to coat the road sufficiently for reflections to appear or when falling rain drops were visible in the images. Comparing dry with rainy weather learning, we found only minor differences between their optimization rates (Fig. 4B). This was especially surprising considering the visibility challenges for humans due to large reflections from puddles as well as raindrops covering the camera lens during driving. Finally, we evaluated different road types by comparing learning on highways and rural roads (Fig. 4C). Since highway driving has a tighter distribution of likely steering control commands (i.e., the car is traveling primarily in a nearly straight trajectory), the agent quickly learns to do well in this environment compared to the rural roads, which often have much sharper and more frequent turns. Additionally, many of the rural roads in our database lacked lane markers, thus making the beginning of learning harder since this is a key visual feature for autonomous navigation.

In our experiments, our learned agents iteratively explore and observe their surroundings (e.g. trees, cars, pedestrians, etc.) from novel viewpoints. On average, the learning agent converges to autonomously drive 10km without crashing within 1.5 million training iterations. Thus, when randomly placed in new locations with similar features during training the agent is able to use its learned policy to navigate. While demonstration of learning in simulation is critical for development of autonomous vehicle controllers, we also evaluate the learned policies directly on-board our full-scale autonomous vehicle to test generalization to the real-world.

C. Evaluation in the Real World

Next, we evaluate **VISTA** and baseline models deployed in the real-world. First, we note that models trained solely in CARLA did not transfer, and that training with data viewpoint augmentation [2] strictly improved performance of the baselines. Thus, we compare against baselines with augmentation.

Each model is trained 3 times and tested individually on every road on the test track. At the end of a road, the vehicle

TABLE I

REAL-WORLD PERFORMANCE COMPARISON. EACH ROW DEPICTS A DIFFERENT PERFORMANCE METRIC EVALUATED ON OUR TEST TRACK. BOLD CELLS IN A SINGLE ROW REPRESENT THE BEST PERFORMERS FOR THAT METRIC, WITHIN STATISTICAL SIGNIFICANCE.

		DR-AUG (Tobin et al. [13])	**S2R-AUG** (Bewley et al. [9])	**IMIT-AUG** (Bojarski et al. [2])	**VISTA** (Ours)	**HUMAN** (Gold Std.)
Lane Following	# of Interventions	13.6 ± 2.62	4.33 ± 0.47	3.00 ± 0.81	$\mathbf{0.0 \pm 0.0}$	0.0 ± 0.0
	Dev. from mean [m]	$\mathbf{0.26 \pm 0.03}$	$\mathbf{0.31 \pm 0.06}$	$\mathbf{0.30 \pm 0.04}$	$\mathbf{0.29 \pm 0.05}$	0.22 ± 0.01
Near Crash Recovery (rate)	Trans. R (+1.5m)	0.57 ± 0.03	0.6 ± 0.05	0.71 ± 0.03	$\mathbf{1.0 \pm 0.0}$	1.0 ± 0.0
	Trans. L (+1.5m)	0.51 ± 0.08	0.51 ± 0.08	0.67 ± 0.09	$\mathbf{0.97 \pm 0.03}$	1.0 ± 0.0
	Yaw CW ($+30°$)	0.35 ± 0.06	0.31 ± 0.11	0.44 ± 0.06	$\mathbf{0.91 \pm 0.06}$	1.0 ± 0.0
	Yaw CCW ($-30°$)	0.37 ± 0.03	0.33 ± 0.05	0.37 ± 0.03	$\mathbf{0.93 \pm 0.05}$	1.0 ± 0.0

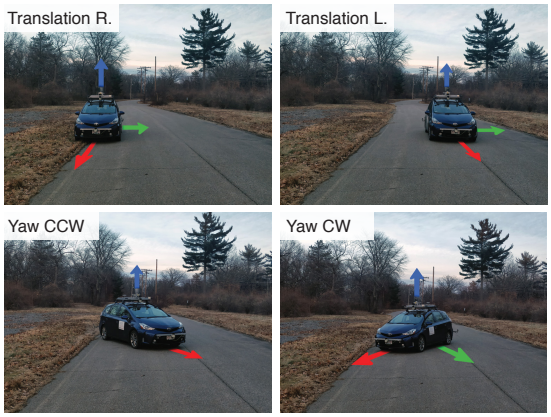

Fig. 6. **Robustness analysis.** We test robustness to recover from near crash positions, including strong translations (top) and rotations (bottom). Each model and starting orientation is repeated at 15 locations on the test track. A recovery is successful if the car recovers within 5 seconds.

is restarted at the beginning of the next road segment. The test driver intervenes when the vehicle exits its lane. The mean trajectory of the three trials are shown in Fig. 5A, with intervention locations drawn as red points. Road boundaries are plotted in black for scale of deviations. **IMIT-AUG** yielded highest performance out of the three baselines, as it was trained directly with real-world data from the human driver. Of the two models trained with only CARLA control labels, **S2R-AUG** outperformed **DR-AUG** requiring an intervention every 700m compared to 220m. Even though **S2R-AUG** only saw control labels from simulation, it received both simulated and real perception. Thus, the model learned to effectively transfer some of the details from simulation into the real-world images allowing it to become more stable than purely randomizing away certain properties of the simulated environment (ie. **DR-AUG**). **VISTA** exhibited the best performance of all the considered models and never required any interventions throughout the trials (totaling $>$ 10km of autonomous driving).

The variance across trials is visualized in Fig. 5B-C (line color in (B) indicates variance at that location). For each baseline, the variance tended to spike at locations that resulted in interventions, while the variance of **VISTA** was highest in ambiguous situations such as approaching an intersection, or wider roads with multiple possible correct control outputs.

We also initiated the vehicle from off-orientation positions with significant lateral and rotational offsets to evaluate robustness to recover from these near-crash scenarios. A successful recovery is indicated if the vehicle is able to successfully maneuver and drive back to the center of its lane within 5 seconds. We observed that agents trained in **VISTA** were able to recover from these off-orientation positions on real and previously unencountered roads, and also significantly outperformed models trained with imitation learning on real world data (**IMIT**) or in CARLA with domain transfer (**DR-AUG** and **S2R-AUG**). On average, **VISTA** successfully recovered over $2\times$ more frequently than the next best, **IMIT-AUG**. The performance of **IMIT-AUG** improved with translational offsets, but was still significantly outperformed by **VISTA** models trained in simulation by approximately 30%. All models showed greater robustness to recovering from translations than rotations since rotations required significantly more aggressive control to recover with a much smaller room of error. In summary, deployment results for all models are shown in Table I.

VI. CONCLUSION

Simulation has emerged as a potential solution for training and evaluating autonomous systems on challenging situations that are often difficult to collect in the real-world. However, successfully transferring learned policies from model-based simulation into the real-world has been a long-standing field in robot learning. In this paper, we present **VISTA**, an end-to-end data-driven simulator for training autonomous vehicles for deployment into the real-world. **VISTA** supports training agents anywhere within the feasible band of trajectories that can be synthesized from data collected by a human driver on a single trajectory. In the future, we will focus on not only synthesizing perturbations to the ego-agent, but also to other dynamic obstacles in the environment (i.e. cars, pedestrians, etc) [27], [36] or the environment [37].

Our experiments empirically validate the ability to train models in **VISTA** using RL, and directly deploy these learned policies on a full-scale autonomous vehicle that can then successfully drive autonomously on real roads it has never seen before. We demonstrate that our learned policies exhibit greater robustness in recovery from near-crash scenarios. While we treat lane-stable control as the problem of choice, the methods and simulator presented here are extendable to robust learning of more complex policies such as point-to-point navigation [3], object avoidance [38], and lane changes [39]. We believe our approach represents a major step towards the direct, real world deployment of end-to-end learning techniques for robust training of autonomous vehicle controllers.

REFERENCES

[1] D. A. Pomerleau, "ALVINN: An Autonomous Land Vehicle in a Neural Network," in *Advances in Neural Information Processing Systems 1*, 1989, pp. 305–313.

[2] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[3] A. Amini, G. Rosman, S. Karaman, and D. Rus, "Variational end-to-end navigation and localization," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

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

[5] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[6] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019. [Online]. Available: https://drake.mit.edu

[7] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*. Springer, 2018, pp. 621–635.

[8] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.

[9] A. Bewley, J. Rigley, Y. Liu, J. Hawke, R. Shen, V.-D. Lam, and A. Kendall, "Learning to Drive from Simulation without Real World Labels," *arXiv preprint arXiv:1812.03823*, 2018. [Online]. Available: http://arxiv.org/abs/1812.03823

[10] M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *arXiv preprint arXiv:1808.00177*, 2018.

[11] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, p. eaau4984, 2019.

[12] F. Sadeghi and S. Levine, "Cad2rl: Real single-image flight without a single real image," *arXiv preprint arXiv:1611.04201*, 2016.

[13] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 23–30.

[14] J. Bruce, N. Sünderhauf, P. Mirowski, R. Hadsell, and M. Milford, "Learning deployable navigation policies at kilometer scale from a single traversal," *arXiv preprint arXiv:1807.05211*, 2018.

[15] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *arXiv preprint arXiv:1812.03079*, 2018.

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

[17] M. Henaff, A. Canziani, and Y. LeCun, "Model-predictive policy learning with uncertainty regularization for driving in dense traffic," *arXiv preprint arXiv:1901.02705*, 2019.

[18] Z.-W. Hong, C. Yu-Ming, S.-Y. Su, T.-Y. Shann, Y.-H. Chang, H.-K. Yang, B. H.-L. Ho, C.-C. Tu, Y.-C. Chang, T.-C. Hsiao, *et al.*, "Virtual-to-real: Learning to control in visual semantic segmentation," *arXiv preprint arXiv:1802.00285*, 2018.

[19] A. Amini, W. Schwarting, G. Rosman, B. Araki, S. Karaman, and D. Rus, "Variational autoencoder for end-to-end control of autonomous driving with novelty detection and training de-biasing," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 568–575.

[20] M. Toromanoff, E. Wirbel, F. Wilhelm, C. Vejarano, X. Perrotton, and F. Moutarde, "End to end vehicle lateral control using a single fisheye camera," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3613–3619.

[21] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *arXiv preprint arXiv:1906.03199*, 2019.

[22] A. Amini, L. Paull, T. Balch, S. Karaman, and D. Rus, "Learning steering bounds for parallel autonomous systems," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[23] A. Amini, A. Soleimany, S. Karaman, and D. Rus, "Spatial uncertainty sampling for end-to-end control," *arXiv preprint arXiv:1805.04829*, 2018.

[24] X. Pan, Y. You, Z. Wang, and C. Lu, "Virtual to real reinforcement learning for autonomous driving," 2017.

[25] F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: real-world perception for embodied agents," in *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[26] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "Flightgoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," 2019.

[27] W. Li, C. Pan, R. Zhang, J. Ren, Y. Ma, J. Fang, F. Yan, Q. Geng, X. Huang, H. Gong, *et al.*, "Aads: Augmented autonomous driving simulation using data-driven algorithms," *arXiv preprint arXiv:1901.07849*, 2019.

[28] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, vol. 2, no. 6, 2017, p. 7.

[29] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 85–100.

[30] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 1094–1099.

[31] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[32] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[33] A. Giusti, J. Guzzi, D. Ciresan, F.-L. He, J. P. Rodriguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. Di Caro, D. Scaramuzza, and L. Gambardella, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, 2016.

[34] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[35] F. Naser, D. Dorhout, S. Proulx, S. D. Pendleton, H. Andersen, W. Schwarting, L. Paull, J. Alonso-Mora, M. H. Ang, S. Karaman, *et al.*, "A parallel autonomy research platform," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 933–940.

[36] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[37] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 700–708.

[38] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. L. Cun, "Off-road obstacle avoidance through end-to-end learning," in *Advances in neural information processing systems*, 2006, pp. 739–746.

[39] S.-G. Jeong, J. Kim, S. Kim, and J. Min, "End-to-end learning of image based lane-change decision," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1602–1607.