2017 IEEE Intelligent Vehicles Symposium (IV)
June 11-14, 2017, Redondo Beach, CA, USA

Imitating Driver Behavior with Generative Adversarial Networks

Alex Kuefler[1], Jeremy Morton[2], Tim Wheeler[2], and Mykel Kochenderfer[2]

***Abstract*— The ability to accurately predict and simulate human driving behavior is critical for the development of intelligent transportation systems. Traditional modeling methods have employed simple parametric models and behavioral cloning. This paper adopts a method for overcoming the problem of cascading errors inherent in prior approaches, resulting in realistic behavior that is robust to trajectory perturbations. We extend Generative Adversarial Imitation Learning to the training of recurrent policies, and we demonstrate that our model rivals rule-based controllers and maximum likelihood models in realistic highway simulations. Our model both reproduces emergent behavior of human drivers, such as lane change rate, while maintaining realistic control over long time horizons.**

I. INTRODUCTION

Accurate human driver models are critical for realistic simulation of driving scenarios, and have the potential to significantly advance research in automotive safety. Traditionally, human driver modeling has been the subject of both rule-based and data-driven approaches. Early rule-based attempts include parametric models of car following behavior, with strong built-in assumptions about road conditions [1] or driver behavior [2]. The Intelligent Driver Model (IDM) [3] extended this work by capturing asymmetries between acceleration and deceleration, preferred free road and bumper-to-bumper headways, and realistic braking behavior. Such car-following models were later extended to multilane conditions with controllers like MOBIL [4], which maintains a utility function and "politeness parameter" to capture intelligent driver behavior in both acceleration and turning. These controllers are all largely characterized by smooth, collision-free driving, but rely on assumptions about driver behavior and a small set of parameters that may not generalize well to diverse driving scenarios.

In contrast, imitation learning (IL) approaches rely on data typically provided through human demonstration in order to learn a policy that behaves similarly to an expert. These policies can be represented with expressive models, such as neural networks, with less interpretable parameters than those used by rule-based methods. Prior human behavior models for highway driving have relied on behavioral cloning (BC), which treats IL as a supervised learning problem, fitting a model to a fixed dataset of expert state-action pairs [5]–[8]. ALVINN [9], an early BC approach, trained a neural network to map raw images and rangefinder inputs to discrete turning actions. Recent advances in computing and deep learning have allowed this approach to scale to realistic scenarios, such as parking lot, highway, and markerless road conditions [10]. These BC approaches are conceptually sound [11], but tend to fail in practice as small inaccuracies compound during simulation. Inaccuracies lead the policy to states that are underrepresented in the training data (e.g., an ego-vehicle edging towards the side of the road), which leads to yet poorer predictions, and ultimately to invalid or unseen situations (e.g., off-road driving). This problem of cascading errors [12] is well-known in the sequential decision making literature and has motivated work on alternative IL methods, such as inverse reinforcement learning (IRL) [13].

Inverse reinforcement learning assumes that the expert follows an optimal policy with respect to an unknown reward function. If the reward function is recovered, one can simply use RL to find a policy that behaves identically to the expert. This imitation extends to unseen states; in highway driving a vehicle that is perturbed toward the lane boundaries should know to return toward the lane center. IRL thus generalizes much more effectively and does not suffer from many of the problems of BC. Because of these benefits, some recent efforts in human driver modeling emphasize IRL [14], [15]. However, IRL approaches are typically computationally expensive in their recovery of an expert cost function. Instead, recent work has attempted to imitate expert behavior through direct policy optimization, without first learning a cost function [16], [17]. Generative Adversarial Imitation Learning (GAIL) [17] in particular has performed well on a number of benchmark tasks, leveraging the insight that expert behavior can be imitated by training a policy to produce actions that a binary classifier mistakes for those of an expert.

In this work, we apply GAIL to the task of modeling human highway driving behavior. Our major contributions are twofold. First, we extend GAIL to the optimization of recurrent neural networks, showing that such policies perform with greater fidelity to expert behavior than feedforward counterparts. Second, we apply our models to a realistic highway simulator, where expert demonstrations are given by real-world driver trajectories included in the NGSIM dataset [18], [19]. We demonstrate that policy networks optimized by GAIL capture many desirable properties of earlier IL models, such as reproducing emergent driver behavior and assigning high likelihood to expert actions, while simultaneously reducing collision and off-road rates necessary for long horizon highway simulations. Unlike past work, our model learns to map raw LIDAR readings and simple, hand-picked road features to continuous actions, adjusting only turn-rate and acceleration each time step.

[1]Alex Kuefler is in the Symbolic Systems Program at Stanford University, Stanford, CA 94305, USA akuefler@stanford.edu

[2]Are in the department of Aeronautics and Astronautics at Stanford University, Stanford, CA 94305, USA {jmorton2, wheelert, mykel}@stanford.edu

978-1-5090-4804-5/17/$31.00 ©2017 IEEE

II. PROBLEM FORMULATION

We regard highway driving as a sequential decision making task in which the driver obeys a stochastic policy $\pi(a \mid s)$ mapping observed road conditions s to a distribution over driving actions a. Given a class of policies π_θ parameterized by θ, we seek the policy that best mimics human driving behavior. We adopt an IL approach to infer this policy from a dataset consisting of a sequence of state-action tuples (s_t, a_t). IL can be performed using BC or reinforcement learning.

A. Behavioral Cloning

Behavioral cloning solves a regression problem in which the policy parameterization is obtained by maximizing the likelihood of the actions taken in the training data. BC works well for states adequately covered by the training data. It is forced to generalize when predicting actions for states with little or no data coverage, which can lead to poor behavior. Unfortunately, even if simulations are initialized in common states, the stochastic nature of the policies allow small errors in action predictions to compound over time, eventually leading to states that human drivers infrequently visit and are not adequately covered by the training data. Poorer predictions can cause a feedback cycle known as cascading errors [20].

In a highway driving context, cascading errors can lead to off-road driving and collisions. Datasets rarely contain information about how human drivers behave in these situations, which can lead BC policies to act erratically when they encounter such states.

Behavioral cloning has been successfully used to produce driving policies for simple behaviors such as car-following on freeways, in which the state and action space can be adequately covered by the training set. When applied to learning general driving models with nuanced behavior and the potential to drive out of lane, BC only produces accurate predictions up to a few seconds [5], [6].

B. Reinforcement Learning

Reinforcement learning (RL) instead assumes that drivers in the real world follow an expert policy π_E whose actions maximize the expected, global return

$$R(\pi, r) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right] \tag{1}$$

weighted by a discount factor $\gamma \in [0, 1)$. The local reward function $r(s_t, a_t)$ may be unknown, but fully characterizes expert behavior such that any policy optimizing $R(\pi, r)$ will perform indistinguishably from π_E.

Learning with respect to $R(\pi, r)$ has several advantages over maximum likelihood BC in the context of sequential decision making [21]. First, $r(s_t, a_t)$ is defined for all state-action pairs, allowing an agent to receive a learning signal even from unusual states. In contrast, BC only receives a learning signal for those states represented in a labeled, finite dataset. Second, unlike labels, rewards allow a learner to establish preferences between mildly undesirable behavior (e.g., hard braking) and extremely undesirable behavioral (e.g., collisions). And finally, RL maximizes the global, expected return on a trajectory, rather than local instructions for each observation. Once preferences are learned, a policy may take mildly undesirable actions now in order to avoid awful situations later. As such, reinforcement learning algorithms provide robustness against cascading errors.

III. POLICY REPRESENTATION

Our learned policy must be able to capture human driving behavior, which involves:

- *Non-linearity* in the desired mapping from states to actions (e.g., large corrections in steering to avoid collisions caused by small changes in the current state).
- *High-dimensionality* of the state representation, which must describe properties of the ego-vehicle, in addition to surrounding cars and road conditions.
- *Stochasticity*, as humans may take different actions each time they encounter a given traffic scene.

To address the first and second points, we represent all learned policies π_θ using neural networks. To address the third point, we interpret the network's real-valued outputs given input s_t as the mean μ_t and logarithm of the diagonal covariance $\log \nu_t$ of a Gaussian distribution. Actions are chosen by sampling $a_t \sim \pi_\theta(a_t \mid s_t)$. We evaluate both feedforward and recurrent network architectures. An example feedforward model is shown in Fig. 2.

Feedforward neural networks directly map inputs to outputs. The most common architecture, multilayer perceptrons (MLPs), consist of alternating layers of tunable weights and element-wise nonlinearities. Neural networks have gained widespread popularity due to their ability to learn robust hierarchical features from complicated inputs [22], [23], and have been used in automotive behavioral modeling for action prediction in car-following contexts [6], [24]–[27], lateral position prediction [28], and maneuver classification [29].

The feedforward MLP is limited in its ability to adequately address partially observable environments. In real world driving, sensor error and occlusions may prevent the driver from seeing all relevant parts of the driving state. By maintaining sufficient statistics of past observations in memory, recurrent policies [30], [31] disambiguate perceptually similar states by acting with respect to histories of, rather than individual, observations. In this work, we represent recurrent policies using Gated Recurrent Unit (GRU) networks due to their comparable performance with fewer parameters than other architectures [32].

We use similar architectures for the feedforward and recurrent policies. The recurrent policies consist of five feedforward layers that decrease in size from 256 to 32 neurons, with an additional GRU layer consisting of 32 neurons. Exponential linear units (ELU) were used throughout the network, which have been shown to combat the vanishing gradient problem while supporting a zero-centered distribution of activation vectors [33]. The MLP policies have the same architecture, except the GRU layer is replaced

with an additional feedforward layer. For each network architecture, one policy is trained through BC and one policy is trained through GAIL. In all, we trained four neural network policies: GAIL GRU, GAIL MLP, BC GRU, and BC MLP.

IV. POLICY OPTIMIZATION

Contrary to BC, which is trained with traditional regression techniques, reinforcement learning policies do not have training labels for individual actions. Controller performance is instead evaluated by expected return. This approach is problematic in modeling human drivers, as the reward function $r(s_t, a_t)$ is unknown. We first discuss a method for training a policy with a known reward function and then provide a method for learning the reward function.

A. Trust Region Policy Optimization

Policy gradient algorithms are a particularly effective class of reinforcement learning techniques for optimizing differentiable policies, including neural networks. As with standard backpropagation, network parameters are optimized using gradient-based updates, but the gradient can only be approximated using simulated rollouts of the policy interacting with the environment.

This empirical gradient estimate typically exhibits a high amount of variance. In practice, this variance can cause parameter updates that do not improve or even reduce performance. In this work, we use Trust Region Policy Optimization (TRPO) to learn our human driving policies [34]. TRPO updates policy parameters through a constrained optimization procedure that enforces that a policy cannot change too much in a single update, and hence limits the damage that can be caused by noisy gradient estimates.

Although the true reward function that governs the behavior of any particular human driver is unknown, domain knowledge can be used to craft a surrogate reward function such that a policy maximizing this quantity will realize a similar stochastic state-action mapping as π_E. Drivers avoid collisions and going off road, while also favoring smooth driving and minimizing lane-offset. If such features can be combined into a reward function that closely approximates the true reward function for human driving $r(s_t, a_t)$, then modeling driver behavior reduces to RL. However, handcrafting an accurate reward function is often difficult, which motivates the use of Generative Adversarial Imitation Learning.

B. Generative Adversarial Imitation Learning

Although $r(s_t, a_t)$ is unknown, a surrogate reward $\tilde{r}(s_t, a_t)$ may be learned directly from data, without making use of domain knowledge. GAIL [17] trains a policy to perform expert-like behavior by rewarding it for "deceiving" a classifier trained to discriminate between policy and expert state-action pairs. Consider a set of simulated state-action pairs $\mathcal{X}_\theta = \{(s_1, a_1), (s_2, a_2), ..., (s_T, a_T)\}$ sampled from π_θ and a set of expert pairs $\mathcal{X}_E$ sampled from π_E. For a neural network D_ψ parameterized by ψ, the GAIL objective is given by:

$$\max_{\psi} \min_{\theta} V(\theta, \psi) = \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{X}_E}[\log D_\psi(s, a)] + \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{X}_\theta}[\log(1 - D_\psi(s, a))]. \tag{2}$$

When fitting ψ, Equation (2) can simply be interpreted as a sigmoid cross entropy objective, maximized by minibatch gradient ascent. Positive examples are sampled from $\mathcal{X}_E$ and negative examples are sampled from rollouts generated by interactions of π_θ with the simulation environment. However, $V(\theta, \psi)$ is non-differentiable with respect to θ, requiring optimization via RL.

In order to fit π_θ, a surrogate reward function can be formulated from Eq. (2) as:

$$\tilde{r}(s_t, a_t; \psi) = -\log(1 - D_\psi(s_t, a_t)), \tag{3}$$

which approaches infinity as tuples (s_t, a_t) drawn from $\mathcal{X}_\theta$ become indistinguishable from elements of $\mathcal{X}_E$ based on the predictions of D_ψ. After performing rollouts with a given set of policy parameters θ, surrogate rewards $\tilde{r}(s_t, a_t; \psi)$ are calculated and TRPO is used to perform a policy update. Although $\tilde{r}(s_t, a_t; \psi)$ may be quite different from the true reward function optimized by experts, it can be used to drive π_θ into regions of the state-action space similar to those explored by π_E.

V. DATASET

We use the public Next-Generation Simulation (NGSIM) datasets for US Highway 101 [19] and Interstate 80 [18]. NGSIM provides 45 minutes of driving at 10 Hz for each roadway. The US Highway 101 dataset covers an area in Los Angeles approximately 640 m in length with five mainline lanes and a sixth auxiliary lane for highway entrance and exit. The Interstate 80 dataset covers an area in the San Francisco Bay Area approximately 500 m in length with six mainline lanes, including a high-occupancy vehicle lane and an onramp.

Traffic density in both datasets transitions from uncongested to full congestion and exhibits a high degree of vehicle interaction as vehicles merge on and off the highway and must navigate in congested flow. The diversity of driving conditions and the forced interaction of traffic participants makes these sources particularly useful for behavioral studies. The trajectories were smoothed using an extended Kalman filter [35] on a bicycle model and projected to lanes using centerlines extracted from the NGSIM CAD files. Cars, trucks, buses, and motorcycles are in the dataset, but only car trajectories were used for model training.

VI. EXPERIMENTS

In this work, we use GAIL and BC to learn policies for two-dimensional highway driving. The performance of these policies is subsequently evaluated relative to baseline models.

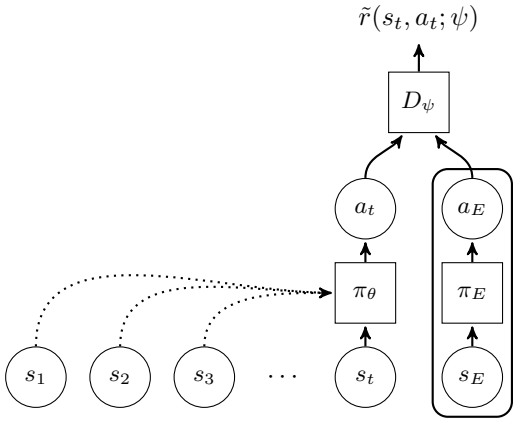

Fig. 1: GAIL diagram. Circles represent values, rectangles represent operations, and dotted arrows represent recurrence relations. Whereas policy π_θ states s_t and actions a_t are evaluated by discriminator D_ψ during both training of π_θ and D_ψ, expert pairs (boxed) are sampled only while training D_ψ.

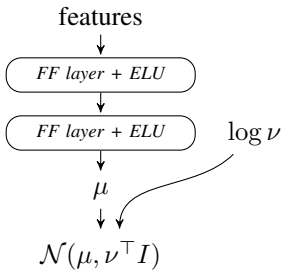

Fig. 2: Architecture for the feedforward multilayer perceptron driving policy. The network output μ and covariance parameters ν are used to construct a Gaussian distribution over driver actions.

A. Environment

All experiments were conducted with the rllab reinforcement learning framework [36]. The simulation environment is a driving simulation on the NGSIM 80 and 101 road networks. Simulations are initialized to match frames from the NGSIM data, and the ego vehicle is randomly chosen from among the traffic participants in the frame. Simulations are run for 100 steps at 10 Hz and are ended prematurely if the ego vehicle is involved in a collision, drives off road, or drives in reverse.

The ego vehicle is driven according to a bicycle model with acceleration and turn-rate sampled from the policy network. All other traffic participants are replayed directly from the NGSIM data, but are augmented with emergency braking in the event of an imminent rear-end collision with the ego vehicle. Specifically, if the acceleration predicted by the Intelligent Driver Model (IDM) [3] is less than an activation threshold of $-2\,\mathrm{m/s^2}$, the vehicle then accelerates according to the IDM while tracking the closest lane centerline. The IDM is parameterized with a desired speed equal to the vehicle's speed at transition, a minimum spacing of 1 m, a desired time headway of 0.5 s, a nominal acceleration of $3\,\mathrm{m/s^2}$, and a comfortable braking deceleration of $2.5\,\mathrm{m/s^2}$.

B. Features

All experiments use the same set of features. These features can be decomposed into three sets. The first set, the *core features*, are eight scalar values that provide basic information about the vehicle's odometry, dimensions, and the lane-relative ego state. These are listed in Table I.

The core features alone are insufficient to describe the local context. Information about neighboring vehicles and the local road structure must be incorporated as well. Several approaches exist for encoding such information in hand-selected features relevant to the driving task [37]. Rather than restrict the model to a subset of vehicle relationships, we introduce a more general and flexible feature representation.

TABLE I: Core features used by the neural networks.

Feature	Units	Description
Speed	$\mathrm{m\,s^{-1}}$	longitudinal speed
Vehicle Length	m	bounding box length
Vehicle Width	m	bounding box width
Lane Offset	m	lateral centerline offset
Lane-Relative Heading	rad	heading angle in the Frenet frame
Lane Curvature	$\mathrm{m^{-1}}$	curvature of closest centerline point
Marker Dist. (L)	m	lat. dist. to left lane marking
Marker Dist. (R)	m	lat. dist. to right lane marking

In addition to the core features, a set of LIDAR-like beams emanating from the vehicle are used to gather information about its surroundings. These beams measure both the distance and range rate of the first vehicle struck by them, up to a maximum range. Our work used a maximum range of 100 m, with 20 range and range rate beams, each spaced uniformly in complete 360° coverage around the ego vehicle's center, as shown in Fig. 3.

Finally, a set of three indicator features are used to identify when the ego vehicle encounters undesirable states. These features take on a value of one whenever the ego vehicle is involved in a collision, drives off road, or travels in reverse, and are zero otherwise. All features were concatenated into a single 51-element vector and fed into each model.

The previous action taken by the ego vehicle is not included in the set of features provided to the policies. We found that policies can develop an over-reliance on previous actions at the expense of relying on the other features contained in their input. To counteract this problem, we studied

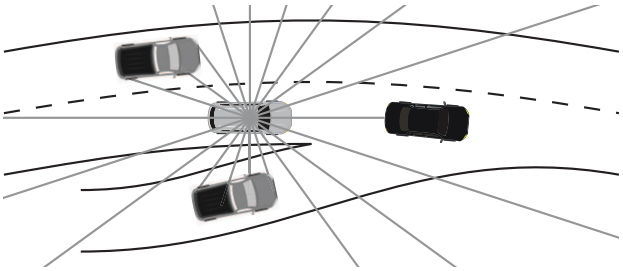

Fig. 3: LIDAR-like beams used for measuring range and range rate.

the effect of replacing the previous actions with random noise early in the training process. However, it was found that even with these mitigations the inclusion of previous actions had a detrimental effect on policy performance.

C. Baseline Models

The first baseline that we used to compare against our deep policies is a static Gaussian (SG) model, which is an unchanging Gaussian distribution $\pi(a \mid s) = \mathcal{N}(a \mid \mu, \Sigma)$ fit using maximum likelihood.

The second baseline model is a BC approach using mixture regression (MR) [6]. The model has been used for model-predictive control and has been shown to work well in simulation and in real-world drive tests. Our MR model is a Gaussian mixture over the joint space of the actions and features, trained using expectation maximization [38]. The stochastic policy is formed from the weighted combination of the Gaussian components conditioned on the features. Greedy feature selection is used during training to select a subset of predictors up to a maximum feature count threshold while minimizing the Bayesian information criterion [39].

The final baseline model uses a rule-based controller to govern the lateral and longitudinal motion of the ego vehicle. The longitudinal motion is controlled by the Intelligent Driver Model with the same parameters as the emergency braking controller used in the simulation environment. For the lateral motion, MOBIL [4] is used to select the desired lane, with a proportional controller used to track the lane centerline. A small amount of noise is added to both the lateral and longitudinal accelerations to make the controller nondeterministic.

D. Validation

To evaluate the relative performance of each model, we performed a systematic validation procedure. For each model, 1,000 ten-second scenes were simulated 20 times each in an environment identical to the one used to train the GAIL policies. As these rollouts were performed, several metrics were extracted to quantify the ability of each model to simulate human driver behavior.

1) Root-Weighted Square Error: The root-weighted square error (RWSE) captures the deviation of a model's probability mass from real-world trajectories [40]. For predicted variable v over m trajectories, we estimate the RWSE by sampling $n = 20$ simulated traces per recorded trajectory:

$$\text{RWSE}_H = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(v_H^{(i)} - \hat{v}_H^{(i,j)}\right)^2}, \qquad (4)$$

where $v_H^{(i)}$ is the true value in the ith trajectory at time horizon H and $\hat{v}_H^{(i,j)}$ is the simulated variable under sample j for the ith trajectory at time horizon H. We extract the RWSE in predictions of global position, centerline offset, and speed over time horizons up to 5 s.

2) Kullback-Leibler Divergence: Driver models should produce distributions over emergent quantities that match those observed in real-world data. For each model, empirical distributions were computed over speed, acceleration, turn-rate, jerk, and inverse time-to-collision (iTTC) over simulated trajectories. The closeness between the simulated and real-world distributions was quantified using the Kullback-Leibler (KL) divergence. Piecewise uniform distributions with 100 evenly spaced bins were used.

3) Emergent Behavior: We also extracted a set of emergent metrics that indicate model imitation performance in relation to the NGSIM dataset. These additional metrics are the lane change rate, the offroad duration, the collision rate, and the hard brake rate.

The lane change rate is the average number of times a vehicle makes a lane change within a 10-second trajectory. Offroad duration is the average number of time steps per trajectory that a vehicle spends more than 1 m outside the closest outer road marker. The collision rate is the fraction of trajectories where the ego vehicle intersects with another traffic participant. The hard brake rate captures the frequency at which a model chooses to brake harder than $-3\ \text{m/s}^2$.

The environment in which validation occurs is not entirely realistic, as the non-ego vehicles have pre-recorded trajectories and do not always properly respond to deviations of the ego vehicle from its original trajectory, leading to an artificially high number of collisions. Hence, we also extract the hard brake rate to help quantify how often dangerous driving situations occur.

VII. RESULTS

We present validation results for root-weighted square error in Fig. 4. The RWSE results show that the BC models have competitive short-horizon performance, but accumulate error over longer time horizons. GAIL produces more stable trajectories and its short term predictions perform well. We clearly see the controller adhere to the lane-centerline, so its lane offset error is close to a constant 0.5, which demonstrates that human drivers do not always closely track the nearest lane-centerline.

KL divergence scores are given in Fig. 5. The KL divergence results show very good tracking for SG in everything but jerk. SG cannot overfit and always takes the average action, so its performance on other metrics is unremarkable. GAIL GRU performs well on the iTTC, speed, and acceleration metrics, but it does poorly with respect to turn-rate and jerk. This poor performance is likely due to the

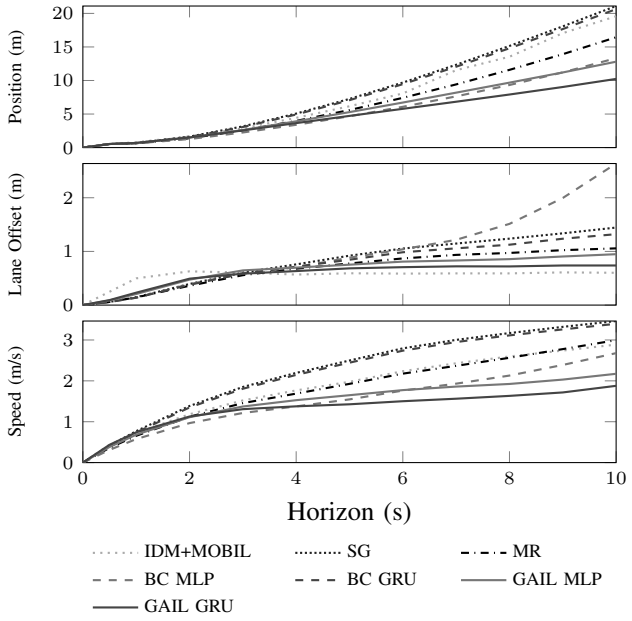

Fig. 4: The root weighted square error for each candidate model vs. prediction horizon. Deep policies outperform the other methods.

fact that although on average the GAIL GRU policy takes similar actions to humans, it tends to oscillates between these actions. For instance, rather than outputting a turn-rate of zero on straight road stretches, it alternates between outputting small positive and negative turn-rates. In comparison with the GAIL policies, the BC policies perform worse on iTTC. The GRU version has the largest KL divergence in acceleration, mostly due to its accelerations being generally small in magnitude, but does reasonably well with turn-rate and jerk.

Validation results for emergent variables are given in Fig. 6. The emergent values show that the GAIL policies outperform the BC policies, acting more like the hand-coded controller. In comparison to BC, the GAIL GRU policy has the closest match to the data everywhere except for hard brakes, as it rarely takes extreme actions. Mixture regression largely performs better than SG and is on par with the BC policies, but is still susceptible to cascading errors. Offroad duration is perhaps the most striking statistic; only GAIL (and of course IDM + MOBIL) stay on the road for extended stretches. SG never brakes hard as it only drives straight, causing many collisions as a consequence. It is interesting that the collision rate for IDM + MOBIL is roughly the same as the collision rate for GAIL GRU, despite the fact that IDM + MOBIL should not collide. The inability of other vehicles within the simulation environment to fully react to the ego-vehicle may explain this phenomenon.

The results demonstrate that GAIL-based models capture many desirable properties of both rule-based and machine learning methods, while avoiding common pitfalls. With the exception of the hand-coded controller, GAIL policies achieve the lowest collision and off-road driving rates, considerably outperforming baseline and similarly structured BC models. However, GAIL also achieves a lane change rate closer to real human driving in comparison to other learning methods.

Furthermore, extending GAIL to recurrent policies leads to improved performance. This result is an interesting contrast with the BC policies, where the addition of recurrence tends not to yield better results. Thus, we find that recurrence by itself is insufficient for addressing the detrimental effects that cascading errors can have on BC policies.

VIII. CONCLUSIONS

This paper demonstrates the effectiveness of deep imitation learning as a means of training driver models that perform realistically over long time horizons while simultaneously capturing microscopic, human-like behavior. Our contributions have been to (1) extend Generative Adversarial Imitation Learning to the optimization of recurrent policies, and to (2) apply this technique to the creation of a new, intelligent model of highway driving that outperforms the state of the art on several metrics. Although behavioral cloning performs on par with Generative Adversarial Imitation Learning on short ($\sim$2 s) horizons, its greedy behavior prevents it from achieving realistic driving over an extended period. The use of *policy optimization* by Generative Adversarial Imitation Learning enables us to overcome this problem of cascading errors to produce long-term, stable trajectories. Furthermore, the use of *policy representation* by deep, recurrent neural networks enables us to learn directly from general sensor inputs (i.e., LIDAR distance and range rate) that can capture arbitrary traffic states and simulate partial observability.

We have argued that deep imitation learning can be used to produce realistic highway driver models, but we must stress that this framework is quite general. There is no reason, in principle, why LIDAR feature representations or recurrent policy optimization through Generative Adversarial Imitation Learning can not be applied directly to the modeling of other driving conditions (e.g., urban or rural driving), assuming the availability of sufficient training data. In fact, the memory mechanism introduced by gated recurrent units may provide a natural way to allow, for instance, an intersection policy to learn about turn-taking at stop signs. However, we anticipate difficulties capturing the entire range of human driving behavior in a single neural network, as intelligent knowledge transfer between interrelated tasks remains an open problem in deep learning. Future work will explore how latent variable models, such as variational autoencoders, can be combined with our approach in order to capture how drivers change their style depending on scene context. The code associated with this paper can be found at `https://github.com/sisl/gail-driver`.

ACKNOWLEDGMENT

This material is based upon work supported by the Ford Motor Company, Robert Bosch LLC, and the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-114747. We also thank Jayesh Gupta and Max Egorov for useful discussions.

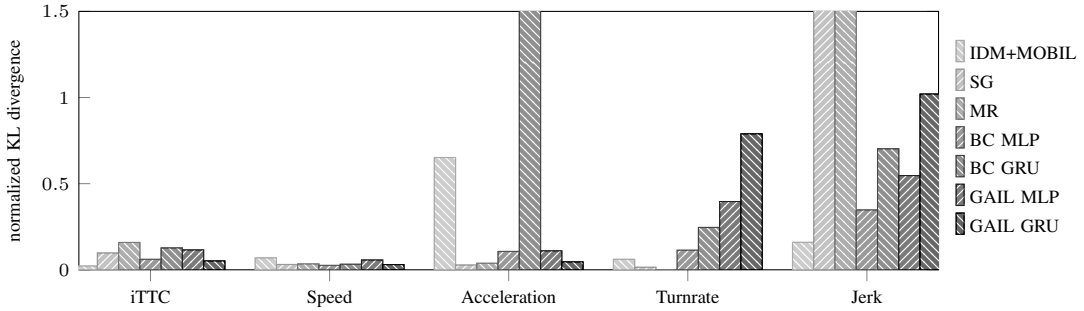

Fig. 5: The KL divergence for various emergent metrics pulled from 10 s trajectories.

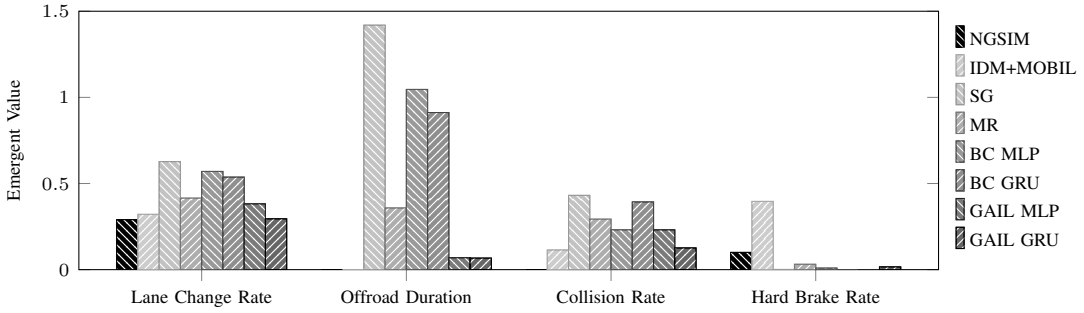

Fig. 6: Emergent values for each model.

REFERENCES

[1] P. A. Seddon, "A program for simulating the dispersion of platoons of road traffic", *Simulation*, vol. 18, no. 3, pp. 81–90, 1972.

[2] P. G. Gipps, "A behavioural car-following model for computer simulation", *Transportation Research Part B: Methodological*, vol. 15, no. 2, pp. 105–111, 1981.

[3] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations", *Physical Review E*, vol. 62, no. 2, p. 1805, 2000.

[4] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model MOBIL for car-following models", *Journal of the Transportation Research Board*, vol. 1999, pp. 86–94, 2007.

[5] T. A. Wheeler, P. Robbel, and M. Kochenderfer, "Analysis of microscopic behavior models for probabilistic modeling of driver behavior", in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016.

[6] S. Lefèvre, C. Sun, R. Bajcsy, and C. Laugier, "Comparison of parametric and non-parametric approaches for vehicle speed prediction", *American Control Conference (ACC)*, pp. 3494–3499, 2014.

[7] T. Gindele, S. Brechtel, and R. Dillmann, "Learning context sensitive behavior models from observations for predicting traffic situations", in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[8] G. Agamennoni, J. Nieto, and E. Nebot, "Estimation of multivehicle dynamics by considering contextual information", *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 855–870, 2012.

[9] D. A. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network", DTIC Document, Tech. Rep. AIP-77, 1989.

[10] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, and J. Zhao, "End to end learning for self-driving cars", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] U. Syed and R. E. Schapire, "A game-theoretic approach to apprenticeship learning", in *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[12] S. Ross and J. Bagnell, "Efficient reductions for imitation learning", in *International Conference on Artificial Intelligence and Statistics*, 2010.

[13] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning", in *International Conference on Machine Learning (ICML)*, 2004.

[14] D. S. González, J. Dibangoye, and C. Laugier, "High-speed highway scene prediction based on driver models learned from demonstrations", in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016.

[15] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverages effects on human actions", in *Robotics: Science and Systems*, 2016.

[16] J. Ho, J. K. Gupta, and S. Ermon, "Model-free imitation learning with policy optimization", in *Inter-*

national Conference on Machine Learning (ICML), 2016.

[17] J. Ho and S. Ermon, "Generative adversarial imitation learning", in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[18] J. Colyar and J. Halkias, "US highway 80 dataset", Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-06-137, 2006.

[19] J. Colyar and J. Halkias, "US highway 101 dataset", Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-07-030, 2007.

[20] J. A. Bagnell, "An invitation to imitation", DTIC Document, Tech. Rep. CMU-RI-TR-15-08, 2015.

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[22] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations", in *International Conference on Machine Learning (ICML)*, 2009.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[24] J. Hongfei, J. Zhicai, and N. Anning, "Develop a car-following model using data collected by five-wheel system", in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2003.

[25] S. Panwai and H. Dia, "Neural agent car-following models", *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 60–70, 2007.

[26] A. Khodayari, A. Ghaffari, R. Kazemi, and R. Braunstingl, "A modified car-following model based on a neural network model of the human driver effects", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 6, pp. 1440–1449, 2012.

[27] J. Morton, T. A. Wheeler, and M. J. Kochenderfer, "Analysis of recurrent neural networks for probabilistic modeling of driver behavior", *IEEE Transactions on Intelligent Transportation Systems*, 2016.

[28] Q. Liu, B. Lathrop, and V. Butakov, "Vehicle lateral position prediction: A small step towards a comprehensive risk assessment system", in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[29] P. Boyraz, M. Acar, and D. Kerr, "Signal modelling and hidden Markov models for driving manoeuvre recognition and driver fault diagnosis in an urban road scenario", in *IEEE Intelligent Vehicles Symposium*, 2007.

[30] D. Wierstra, A. Förster, J. Peters, and J. Schmidhuber, "Recurrent policy gradients", *Logic Journal of IGPL*, vol. 18, no. 5, pp. 620–634, 2010.

[31] N. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver, "Memory-based control with recurrent neural networks", *ArXiv preprint arXiv:1512.04455*, 2015.

[32] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling", *ArXiv preprint arXiv:1412.3555*, 2014.

[33] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)", *ArXiv preprint arXiv:1511.07289*, 2015.

[34] J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz, "Trust region policy optimization", in *International Conference on Machine Learning (ICML)*, 2015.

[35] R. E. Kalman, "A new approach to linear filtering and prediction problems", *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[36] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control", *ArXiv preprint arXiv:1604.06778*, 2016.

[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.

[38] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*. Springer, 2001.

[39] G. Schwarz, "Estimating the dimension of a model", *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[40] J. Cox and M. J. Kochenderfer, "Probabilistic airport acceptance rate prediction", in *AIAA Modeling and Simulation Conference*, 2016.