FLUIDS Imitation Learning Experiments

Hankun Zhao*[1], Andrew Cui*[1], Schuyler A. Cullen[3], Brian Paden[3], Michael Laskey[1], Ken Goldberg[1,2]

To evaluate FLUIDS as a platform for developing autonomous vehicle agents at multiple levels, we explore learning a velocity planner from the FLUIDS predictive velocity planner. We collect training data from rollouts of the supervisor on cars interacting in a four-way intersection. We study how quickly the data can be collected, how fast the learned policy can be evaluated, and how sensitive the learned policy is to noise.

In IL, a supervisor provides demonstrations to an agent, and a policy mapping observed states to controls is learned. Imitation learning has been applied for on robots for tasks of grasping in clutter [2] and two-hand grasping [4]. For our purposes, the states are the LIDAR view and the controls will be the target velocity. We chose to learn this part of the pipeline because it is the most computationally expensive for the implemented supervisor.

We can formalize this learning as follows: denote a policy as a measurable function $\pi : \mathcal{X} \to \mathcal{V}$ from the driver's view state space $\mathcal{X}$ to target velocities inputs $\mathcal{V}$. We consider policies $\pi_\theta : \mathcal{X} \to \mathcal{U}$ parameterized by some $\theta \in \Theta$, such as the weights of a neural network, or the decision boundaries in a decision tree. For our experiments we use SciKit-Learn's decision tree implementation.

Under the assumptions, any such policy π_θ induces a probability density over the set of trajectories of length T:

$$p(\tau|\pi_\theta) = p(\mathbf{x}_0) \prod_{t=0}^{T-1} p(\mathbf{x}_{t+1}|\pi_\theta(\mathbf{x}_t), \mathbf{x}_t)\,,$$

where a trajectory τ of length T is a sequence of state and velocity tuples: $\tau = \{(\mathbf{x}_t, v_t)\}_{t=0}^{T-1}$

We collect demonstrations using FLUIDS' velocity planner π_{θ^*}, where θ^* may not be contained in Θ. To reduce the effect of errors compounding from execution, we also inject noise in the supervisor's velocity controller when generating training data [3]. Demonstrations are featurized using the quasi-LIDAR featurizer. We choose the quasi-LIDAR representation for its lower dimensional featurization. We collect 800 trajectories of data from the implemented supervisor. From each trajectory, we extract data points as state-action pairs for every vehicle in the trajectory.

We measure the difference between the learner and supervisor using a surrogate loss $l : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ [6], [5]. The surrogate loss we consider is the indicator function, since the target velocities are discretized, $l(v_1, v_2) = \mathbb{1}(v_1 \neq v_2)$. The objective of LfD is to minimize the expected surrogate loss under the distribution induced by the robot's policy:

$$\min_\theta \; E_{p(\tau|\pi_\theta)} \sum_{t=0}^{T-1} l(\pi_\theta(\mathbf{x}_t), \pi_{\theta^*}(\mathbf{x}_t))\,. \tag{1}$$

However, in practice, this objective is difficult to optimize because of the coupling between the loss and the robot's distribution on states. Thus, we instead minimize an upper-bound on this objective [1] via sampling N trajectories from the supervisor's policy.

$$\min_\theta \; \sum_{n=0}^{N-1} \sum_{t=0}^{T-1} l(\pi_\theta(\mathbf{x}_{t,n}), \pi_{\theta^*}(\mathbf{x}_{t,n}))\,; \quad \tau \sim p(\tau|\pi_{\theta^*})\,. \tag{2}$$

We evaluate the sensitivity of the learned velocity controller to parameters in the simulator. We perform a grid search over six parameters of the simulator:

- The number of cars in the scene, in the range $[2, 5]$.
- The number of pedestrians in the scene, in the range $[0, 4]$.
- The variance, λ_l of quasi-LIDAR noise, in the range $[0, 1]$.
- The probability of omitting a sensor reading from the quasi-LIDAR state, in the range $[0, 1]$.
- The presence of traffic lights, $[0, 1]$.
- The mass of the cars, in the range $[0, 200]$.

The analysis reports that the performance of the learned policy is very sensitive to perturbations in the number of cars in the scene, with more cars providing a challenge and leading to a lower success rate. The sensitivity analysis also allows us to visualize the effect of quasi-LIDAR omissions, with our imitation learned agent being robust to increased omission probability until all the quasi-LIDAR observations are omitted, after which the success rate drops drastically. Additionally, the sensitivity analysis indicates the importance of traffic lights, which coordinate traffic through the intersection and prevent gridlock. Finally, the sensitivity analysis shows the importance of car mass, with slightly different car masses providing a model mismatch and leading to lower success rates.

I. Acknowledgments

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, the Real-Time Intelligent Secure Execution (RISE) Lab, and the CITRIS "People and Robots" (CPAR) Initiative and with UC Berkeley's Center for Automation and Learning for Medical Robotics (Cal-MR). The authors were supported in part by donations from Siemens, Google, Honda, Intel, Comcast, Cisco, Autodesk, Amazon Robotics, Toyota Research Institute, ABB, Samsung, Knapp, and Loccioni.

*Denotes Equal Contribution
[1]Department of Electrical Engineering and Computer Science
[2]Department of Industrial Engineering and Operations Research
[3]Samsung Strategy and Innovation Center
[1−2]The AUTOLAB at UC Berkeley; Berkeley, CA 94720, USA
jerryz123@berkeley.edu , andycui97@berkeley.edu, laskeymd@berkeley.edu, goldberg@berkeley.edu

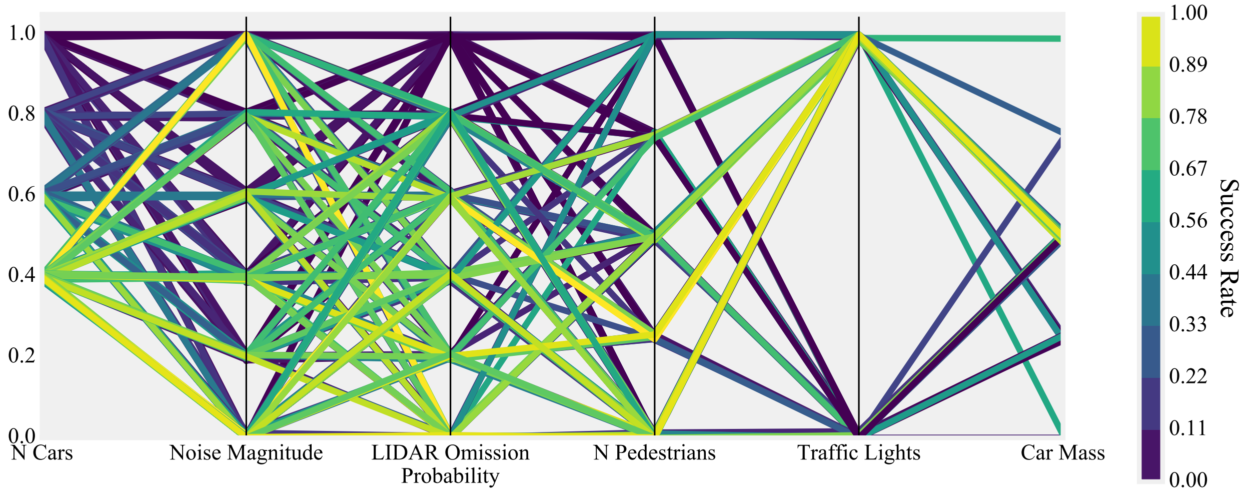

Fig. 1: Sensitivity analysis of the imitation learner to state complexity and observation noise, performed over 256 configurations by evaluating the learned policy on 5120 trajectories. The simulator facilitates a rapid parallel search over these configurations. For visualization we normalize all parameters between 0 and 1. We report success rate as the percentage of evaluations in which the learned policy successfully guides all vehicles to their goals. The analysis reports that our imitation learned agent is tolerant to noisy observations, but fails when run in an environment with a large number of cars or pedestrians. We also observe model mismatch when we alter the car mass and remove traffic lights.

We thank our colleagues who provided helpful feedback and suggestions, in particular, Daniel Seita, and Sanjay Krishnan.

References

[1] M. Laskey, C. Chuck, J. Lee, J. Mahler, S. Krishnan, K. Jamieson, A. Dragan, and K. Goldberg, "Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 358–365.

[2] M. Laskey, J. Lee, C. Chuck, D. Gealy, W. Hsieh, F. T. Pokorny, A. D. Dragan, and K. Goldberg, "Robot grasping in clutter: Using a hierarchy of supervisors for learning from demonstrations," in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*. IEEE, 2016, pp. 827–834.

[3] M. Laskey, J. Lee, R. Fox, A. Dragan, and K. Goldberg, "Dart: Noise injection for robust imitation learning," in *Proceedings of the 1st Annual Conference on Robot Learning (CORL)*, 2017, pp. 143–156.

[4] J. Liang, J. Mahler, M. Laskey, P. Li, and K. Goldberg, "Using dvrk teleoperation to facilitate deep learning of automation tasks for an industrial robot."

[5] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 661–668.

[6] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 627–635.