Inverse Reinforcement Learning on GPUDrive

Naicheng He* [1] Arjun Prakash* [1] Gokul Swamy [2] Amy Greenwald [1] Eugene Vinitsky [3]

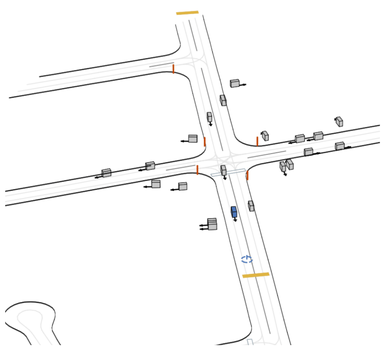
[1]Brown University [2]Carnegie Mellon University [3]New York University

Questions

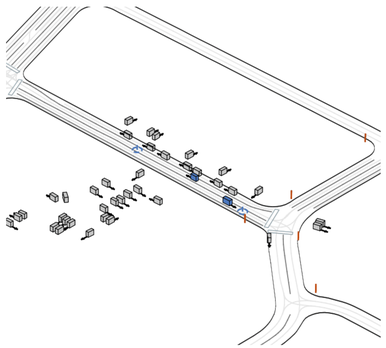
- How can we learn robust multi-agent policies in large-scale, real-world environments using inverse reinforcement learning?
- How can we derive reward signals that support re-training and transfer to unseen environments?
- Can inverse reinforcement learning be used to align human expert behavior with PPO agents?

Introduction

GPUDrive [4] is a GPU-accelerated, data-driven, multi-agent driving simulator built on the Waymo Open Motion Dataset [1]. Compared to standard MuJoCo IRL benchmarks [3, 9], multi-agent driving presents a more challenging and realistic setting due to: (i) *multi-agent interactions*—even small policy deviations can cause large scene changes, complicating credit assignment and stability [10]; and (ii) *scene diversity*, which makes generalization both necessary and difficult.



(a) Single controlled agent.



(b) Three controlled agents.

Experiment Setup and Evaluation

We conduct preliminary experiments across 75 worlds with varying numbers of controlled agents where agents imitate trajectories generated by a ppo expert agent. In the single-agent setting, all but one controlled agent operate in log-replay mode. Agent performance is evaluated using: (i) goal rate and crash rate, and (ii) a reward metric that assigns $+1$ for reaching the goal and -1 for collisions/crashes (hidden from the agent during training).

We perform a sweep over several design choices, and report results for both the single-agent and 64-agent cases in the middle column. Two choices were particularly impactful: (i) replacing GAIL-style rewards with AIRL-style rewards, and (ii) tuning the discriminator capacity via learning-rate adjustments and regularization. Regularization on the inner-loop RL networks is found to stabilize training.

Imitation Gap in Single-Agent Worlds

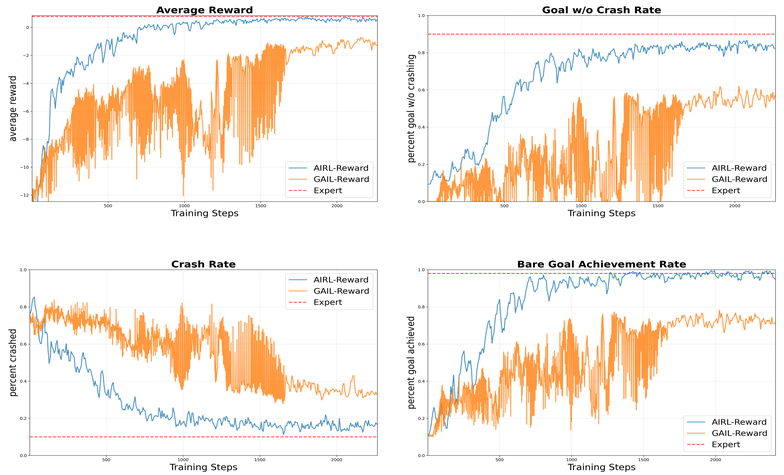


Figure 2. One interesting observation is that the imitated policy almost always goes to the goal with or without crashing. Also, note that here the expert PPO policy is not meant for log-replay mode and thus results in a 10% crashing rate.

Imitation Gap in Multi-Agent Worlds

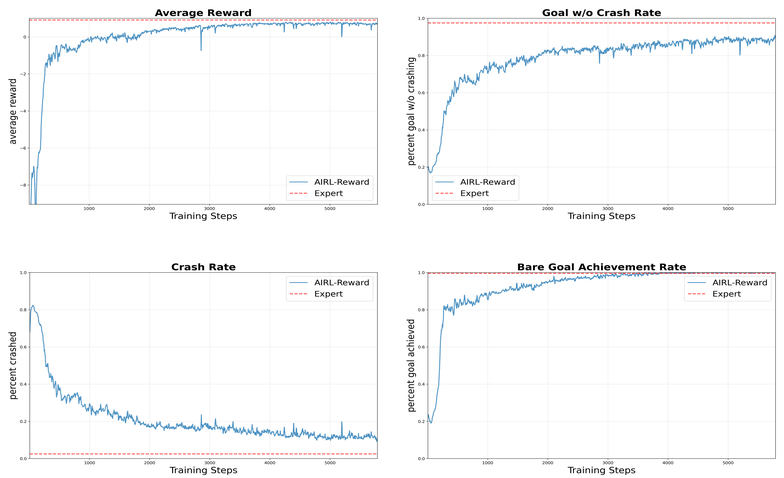


Figure 3. Based on visualization, worlds where the agents are supposed to drive in the opposite direction in nearby lanes are particularly hard for multi-agent IRL. These scenes account for the imitation gap presented in the plot.

Algorithm Background

Imitation learning (IL) aims to reproduce expert behavior π_E from demonstrations. The simplest approach, *behavioral cloning* (BC) [7], treats IL as supervised learning: maximizing the likelihood of expert actions given states. However since training is restricted to expert data, BC suffers from compounding errors [8].

Inverse reinforcement learning (IRL) instead infers a cost function that explains expert behavior [5, 11]. A useful view is the two-player zero-sum game between a policy and a reward model:[a]

$$\min_{r\in R}\max_{\pi\in\Pi}\ \mathbb{E}_{\pi}[r(s,a)] - \mathbb{E}_{\pi_{\mathrm{E}}}[r(s,a)] - \lambda H(\pi). \tag{1}$$

which matches trajectory-level statistics with policy causal entropy regularization.

Generative adversarial imitation learning (GAIL) [3] is one practical solver for equation (1). We optimize the reward model (discriminator) by gradient descent on the cross-entropy classification loss distinguishing expert and generated samples, and we optimize the policy through any RL policy updates. Concretely, GAIL seeks a saddle point (π, D) of

$$\mathbb{E}_{\pi}\left[\log D(s,a)\right] + \mathbb{E}_{\pi_E}\left[\log\Big(1 - D(s,a)\Big)\right] - \lambda H(\pi), \tag{2}$$

by alternating updates to increase equation (2) with respect to D and decrease it with respect to π.

Adversarial Inverse Reinforcement Learning (AIRL) [2] introduces a structured reward formulation $r(s,a) = \log D(s,a) - \log\Big(1 - D(s,a)\Big)$ to make the reward function recoverable instead of the original GAIL reward $r(s,a) = -\log\Big(1 - D(s,a)\Big)$. Empirically, AIRL reward perform better when imitating RL-generated policies, while GAIL reward suit human expert demonstrations [6].

References

[1] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov. Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset, 2021.

[2] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

[3] J. Ho and S. Ermon. Generative adversarial imitation learning, 2016.

[4] S. Kazemkhani, A. Pandya, D. Cornelisse, B. Shacklett, and E. Vinitsky. Gpudrive: Data-driven, multi-agent driving simulation at 1 million fps, 2025.

[5] A. Y. Ng, S. Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[6] M. Orsini, A. Raichuk, L. Hussenot, D. Vincent, R. Dadashi, S. Girgin, M. Geist, O. Bachem, O. Pietquin, and M. Andrychowicz. What matters for adversarial imitation learning? *Advances in Neural Information Processing Systems*, 34:14656–14668, 2021.

[7] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[8] S. Ross, G. J. Gordon, and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011.

[9] G. Swamy, S. Choudhury, J. A. Bagnell, and Z. S. Wu. Of moments and matching: A game-theoretic framework for closing the imitation gap, 2021.

[10] J. Tang, G. Swamy, F. Fang, and Z. S. Wu. Multi-agent imitation learning: Value is easy, regret is hard, 2024.

[11] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[a] IRL is *ill-posed* without additional structure (e.g. a constant reward renders expert optimal).