Learning human like driving policies from real interactive driving scenes

Yann Koeberle, Stefano Sabatini, Dzmitry Tsishkou, Christophe Sabourin

Learning human like driving policies from real interactive driving scenes

Yann Koeberle[1,2],Stefano Sabatini[2],Dzmitry Tsishkou[2] and Christophe Sabourin[1]

***Abstract*— Traffic simulation has gained a lot of interest for massive safety evaluation of self-driving systems in a risk free setting but the reality gap remains a big challenge. Adversarial Imitation learning (AIL) already proved that it is possible to learn driving policies from real demonstrations and more specifically on Highways (NGSIM dataset). However traffic interactions remains very restricted on those scenarios and it is necessary to cope with various and multiple real traffic participants to get real insight of human driving style. INTERACTION dataset was specifically designed for those concerns with complex and rich interactions on a variety of scenes like intersections, roundabouts, ramp merging in different countries. In this paper we introduce our training pipeline that is built upon the Lanelet2 road map format for learning human like driving policies based on most recent implementations of Adversarial Imitation Learning (AIL) algorithms. We compare different AIL algorithms and Behavioural Cloning (BC) baseline on various driving scenes and investigate how realistic driving policies can become as well as their ability to generalise on new scenes. We show that driving policies not only follow expert trajectories but also get safer with less off-road driving and collisions than BC baseline. This work opens new possibilities for multi agent traffic learning based on AIL techniques with real and highly interactive traffic data.**

I. INTRODUCTION

Traffic simulation is a powerful tool to evaluate massively a candidate driving policy in a risk free setting on a large set of scenarios. Quantitative safety evaluation through simulation may become a new standard for Autonomous driving industry before real world deployment of self driving vehicles [38]. However simulating a realistic traffic that could interact consistently with a candidate driving policy is still an open challenge [48]. Heuristic based simulators [39] enables to capture reactive behaviours but hand crafted rules limit traffic interaction diversity especially when road lanes intersect or merge. Learning based approached are more flexible but suffer from several different weaknesses. Reinforcement learning enables to learn general driving policies based on domain knowledge [35],[2] but are not guaranteed to behave as real human driver which is critical for practical safety evaluation in realistic context. In contrast, imitation learning [34],[32] directly leverages demonstrations but lacks common sense and are not robust to distributional drift. As traffic simulation involves multiple agents, it is possible to learn joint traffic plans to animate agents in a supervised way as in [38] but joint policies cannot easily scale to various scenarios with variable number of agents and specific goal and driving style specifications. Learning single agent driving policy shared among agents has already proved its efficiency [22],[9] and could even benefit from expert demonstrations supposing that true environment reward is available thanks to offline reinforcement learning [15]. Practically, rewards are not available even if sparse signals can be incorporated [31] as an incentive to visit expert state distributions which reveals insufficient to find an appropriate balance between imitation and domain rules. Indeed, a traffic simulator is expected to be realistic with respect to real demonstrations regarding metrics as average distance error or other occupancy measures [4] however it should also avoid catastrophic failures like off-road driving and collisions on scenarios unseen during training. One promising way to balance imitation and safety is based on Adversarial Imitation Learning (AIL) [6] with early attempts [22] that already proved that human like driving policies could be learnt on highway environments from NGSIM 80 and 101. Later improvements of original GAIL algorithm [16], enabled to speed up and stabilize the training process [4], as well as extending it to a multi agent setting [6]. Additionally, [7] showed that domain rules can be enforced as a complement to the AIL objective such a way to avoid undesired behaviors on the specific case of highway environments. In this paper, in the light of most recent AIL progress [27] we investigate to which extent AIL can be used for learning single agent driving policy from various and highly interactive real demonstrations from Interaction Dataset [45]. The driving policy is intended for animating traffic agents in simulation and is expected to behave safely as a human driver on new driving scenarios. Additionally, we leverage the Lanelet2 [30] road map format to provide explicit representations of the driving scene to the policy as well as a route based action space that enables to ease exploration during learning. We evaluate performance of the driving policy with regard to imitation metrics that quantify how policy reproduce expert and safety metrics that quantify how safe the driving policy behaves. In our experiments we show that AIL can recover expert behaviour in highly interactive scenes with additional safety improvements on unseen driving scenarios. In summary, our main contributions are:

- A comparison between performance of most recent AIL algorithms on real and highly interactive driving data from Interaction Dataset with regard to imitation and safety metrics,
- An analysis of how driving policy trained with AIL can generalize to new driving scenario unseen during

[1] Univ Paris Est Creteil, LISSI, F-77567 Lieusaint, France
[2] IoV team, Paris Research Center, Huawei Technologies France

training,
- A specific driving policy architecture based on a high level command that specify the task and an action space in curvilinear coordinates locally conditioned on a the route to follow.

The next section discusses related work. Subsequent sections, explain our method and present experiments results.

II. Related Works

We first review main approaches used to simulate locally a traffic and main challenges associated. In a second part, we review related works that specifically tackle driving policy learning from demonstration and justify our choice based on AIL. Finally we review main design choices of the driving policy architecture.

A. Traffic simulation

Traffic simulation has experienced substantial progress since first development of simulators such as SUMO [21] originally designed to study urban mobility or TORCS [42] for racing cars. For autonomous driving applications like driving policy safety evaluation, rule based simulator [21] cannot provide realistic and diverse multi agent interactions to quantitatively estimate safety of a candidate driving policy. Learning a driving policy gives more flexibility and is made possible through Reinforcement learning [28],[35],[2] with high fidelity simulator such as CARLA (Car Learning to Act) [11]. It is even possible to learn simultaneously multiple driving policies for animating each traffic agent in the scene thanks to Multi Agent Reinforcement Learning (MARL) which gave rise to the SMARTS project [48]. Nevertheless MARL reveals much more challenging [8] than RL especially in mixed setting like urban traffic scenes where agents neither fully compete neither fully cooperate. Leveraging human demonstration is crucial to gain sample efficiency and obtain realistic behaviours but SMARTS does not provide a convenient support to exploit common driving data set even if some prior knowledge can be incorporated during learning [19]. In contrast, the BARK project [5] provides a simulator designed specifically to exploit real data from Interaction Dataset [45] but lacks the efficient multi-agent learning interface of SMARTS based on the Ray framework [26] that makes possible massive parallel and distributed simulation crucial to traffic simulation. Indeed, any agent in the scene needs to be controlled by a policy even if it is not learning and multiple episode need to be collected on a single driving scenario before any improvement can be observed which implies massive computational load. Alternatively it is also possible to learn a joint behaviour model for agents in the scene directly from data as in [38] but the ability to scale to various scenarios with changing number of agents with different driving style is weaker than MARL oriented approaches that could adopt the same centralized perspective with for example a centralized critic [14]. Based on those considerations and the limits of aforementioned approaches we motivate the design of our own pipeline that combines bests of SMART and BARK and shortly detail it in III-B.

B. Learning to drive from demonstration

In order to learn human like driving policies we consider several approaches that enables to learn from demonstrations. One of the earliest formulation is Behavioural Cloning (BC) [33] that consists in regressing actions from state features based on expert demonstrations, later extended to conditional imitation learning [10] that direct the policy toward a destination. However BC ignores the temporal effect of each action and thus suffers from covariate shift [37] when evaluated with states which are not covered by the training data contrary to model free RL that [35] can handle long term interactions but often differently from expert. An alternative is to learn the reward function that yields the distribution over all expert trajectories with the maximum entropy through Inverse Reinforcement Learning [50]. IRL applications for self driving vehicles remain limited to specific structured scenarios [18] and often relies on importance sampling which introduce estimate with high variance [13]. Combining best of RL and IRL methods, Generative Adversarial Imitation Learning (GAIL) [16] solves the causal entropy-regularized apprenticeship learning problem thanks to a bi level optimization formulation. Overall, the algorithm consist in learning a policy that imitate an expert policy based on a surrogate reward provided by a discriminator that tries to distinguish expert and apprentice transitions. One of the first application of GAIL was imitating a human driver [22] later extended to a Multi Agent Learning (MAL) setting where the policy is shared among agents during training [6]. To reduce training instabilities proper to MAL, H-GAIL [4] introduced an horizon curriculum to control the complexity of the driving task. Despite the emergence of realistic traffic patterns in [7] it remains very challenging to solve the multi agent imitation learning problem in a whole because multiple sub problems interweave like credit assignment, non stationnarity issues, behaviour multi modality, coordination etc [8], [48]. An alternative [22] is to start training a single agent driver in a simplified environment where the dynamics of other agents is based on a combination of episode replay behaviour and rule based policy like IDM [35]. To avoid that interactions turns unrealistic when apprentice tends to deviate too far from expert reference trajectory one can enforce early episode termination [29] or insert absorbing states [20] in case of undesired events like collision. Additionally, it also possible to enforce domain knowledge during learning with additional cost [7] or directly with semantic signal injected in the discriminator [41]. Despite aforementioned key strengths, the main drawback of GAIL is the original objective of the discriminator which consists in Jensen Shannon divergence that cannot handle distributions that are located in lower dimensional manifolds without overlaps contrary to Wasserstein distances that can still provide a meaningful and smooth representation of the distance in-between. WGAIL [44] use WGAN [3] to estimate a smooth distance between apprentice and expert distribution and could theoretically handle larger policy exploration. Nevertheless, as pointed out in a recent survey [27] the performance of AIL is often evaluated on

synthetic demonstrations but the performance on real data are often poorer. This motivates a deeper analysis of AIL algorithms performance on real driving demonstrations

C. Driving policy architecture

We review main aspects of driving policy design. Concerning action space, a recent survey [49] proposes a classification of driving policies based on the level at which they generate actions with respect to the full self driving system. There are primarily driving policies that operate at the control level with acceleration and turn rate command based on a bicycle model [22],[7] which makes exploration harder because a naive policy can quickly irreversibly diverge from expert. Another method proposed in [17] and [47] decompose the driving task into a high level decision policy that output abstract motion primitives i.e change to left lane, keep current lane etc and a downstream planning and control module that generate corresponding trajectory. Despite a reduced search space, it is often difficult to recover various type of fined grained real trajectories with a fixed planner conditioned on abstract command. As for observation space, we can mainly distinguish raster embedding based on multi level semantic maps obtained with convolutionnal architectures [32], [35] and embedding extracted from specific environment components based on semantic features i.e lane corridor geometry, local neighborhood, most recent trajectory history etc [22] that are lighter to generate for simulation. Another key aspect of general driving policy is its ability to perform various task conditioned on goal specification [32]. Conditioning low level decisions on top level abstract plans seems most suited for autonomous driving because routing module can easily produce a traffic free path to follow as in [40] and [34]. Based on those considerations, we introduce our observation space inspired from [22] and our new action space that we detail in section III-B.

III. Learning to simulate from traffic data

We tackle the problem that consist in learning a realistic and safe driving policy from real demonstrations. We first formulate the learning objective (section III-A) before introducing the driving environment and the policy architecture design (section III-B). Lastly, we detail the full training process in (section III-C).

A. Formulation

The driving task can be modeled as a partially observable infinite horizonγ discounted MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \mu_0, \gamma, \mathcal{O}, \Omega)$. We aim to learn a single agent driving policy π_θ that output at each decision step an action $a \sim \pi(.|o, c) \in \mathcal{A}$ given current observation $o \in \mathcal{O}$ such a way to follow a high level command c provided by a routing module. The observation is generated according to a deterministic model $o = \Omega(s)$. Note that the policy has to interpret the context encoded in o according to the command c to derive a safe action a. The driving scene state $s \in \mathcal{S}$ evolves according to initial state distribution μ_0 and the transition dynamic $\mathcal{T}$. In order to enforce the apprentice policy π_θ to behave as an expert policy π_e we aim to reduce the measure of discrepancy denoted d_ϕ between their respective occupancy measures ρ_π and ρ_E induced by the Markov chain of $\mathcal{M}$ as detailed in [43] while maximising the causal entropy $\mathcal{H}(\pi_\theta)$ of π_θ. Note that the discrepancy has to be learned as well and estimated from expert and apprentice trajectories hence it is parametrized by ϕ. We obtain the following general objective:

$$argmin_\pi d_\phi(\rho_\pi, \rho_E) - \mathcal{H}(\pi_\theta) \tag{1}$$

On one hand, the GAIL algorithm [16] use the Jensen-Shannon divergence as a discrepancy measure $D_{JS}(\rho_\phi, \rho_E)$ which practically results in the following objective based on finite set of transitions samples that are sampled from an online apprentice buffer $\mathcal{B}_\pi$ and an offline expert buffer $\mathcal{B}_e$:

$$max_\phi \mathbb{E}_{(s,a)\sim\mathcal{B}_\pi} log(D(s,a)) + \mathbb{E}_{(s,a)\sim\mathcal{B}_e} log(1 - D(s,a) \tag{2}$$

The above objective finds parameters ϕ of a binary classifier called discriminator and denoted D that best distinguish expert and apprentice transitions. On the other hand WGAIL [46], [43] estimate 1-Wasserstein distance between occupancy measure $W_1^d(\rho_\phi, \rho_E)$ where d is a valid distance metric over $\mathcal{S} \times \mathcal{A}$. The practical objectives express as :

$$max_{D_\phi \in 1-Lip(\mathcal{S}\times\mathcal{A}^\mathbb{R})} \mathbb{E}_{(s,a)\sim\mathcal{B}_e} D(s,a) - \mathbb{E}_{(s,a)\sim\mathcal{B}_\pi} D(s,a) \tag{3}$$

The discriminator can be interpreted as a hard margin Lipschitz classifier and the constrains over the Lipschitz constant of the discriminator is practically enforced thanks to a gradient penalty term added to the loss [3] which results in so called WGAIL-GP algorithm or by controlling the spectral norm of each layer of D [25] which results in WGAIL-SN algorithm. The main advantage of WGAIL is the smoothness of reward function $r(s,a)$ directly obtained by the discriminator D_ϕ that matches the Kantorovich potential in the dual form of optimal transport (OT) with the cost $c(x,y) = |x - y|$ [43]. For GAIL algorithm, we use the following reward function $log(D(s,a)) - log(1 - D(s,a))$ as suggested in [20] in order to reduce reward bias inherent to strictly positive/negative reward. As for the policy π_θ, we optimize PPO objective [36] with weightings coefficients c_v, c_e for the value loss and the entropy bonus.

$$max_\theta \mathbb{E}_{(s_t,a_t,\hat{A}_t^{GAE})\sim\Gamma}[L_{clip}(\theta) + c_V.L_V(\theta) + c_e.\mathcal{H}[\pi_\theta](s_t) \tag{4}$$

First two terms of the losses are based on up to date reward recomputed by discriminator from the training batch denoted Γ in figure 2 which enables to re estimate the Generalized Advantage Estimator (GAE) denoted $\hat{A}_t^{GAE}$ as defined in [36]. The policy term expresses as:

$$L_{clip}(\theta) = min(r_t(\theta).\hat{A}_t^{GAE}, clip(r_t(\theta), 1-\epsilon, 1+\epsilon).\hat{A}_t^{GAE}) \tag{5}$$

where $r_t(\theta) = \frac{\pi_\theta(s_t,a_t)}{\pi_{old}(s_t,a_t)}$ and the value function V_θ loss term expresses as $(V_\theta(s_t) - V_{target})^2)$ with a target values V_{target} computed as suggested in [1].

B. Driving Environment and policy

We train the driving policy in simulation based on a set of driving scenarios $S = (m, f, h)$ generated by an expert database editor. A driving scenario is defined on a bounded map m endowed with a road network in Lanelet2 format [30] extracted from Interaction Dataset. The main structure involved in decision making is the lanelet that constitute semantic peace of road and the lanelet graph that relates all the lanelets of the scene (Fig. 1). The traffic flow of the episode denoted f defines which traffic participants to spawn on the map up to a maximum temporal horizon h at which simulation is artificially terminated. As we focus on a single agent learning, we denote as workers all traffic participants that populate the scene but that don't learn. The dynamics of the scene is mainly based on workers behaviours that consist in replaying original real episode except in emergency situations near collision where it could be replaced by an IDM model [35]. Replaying other agent trajectories is however very likely to lead to collisions with inexperienced ego agents and episode should be terminated earlier thanks to the insertion of an associate absorbing state [20] in the original MDP $\mathcal{M}$. This trick enables to avoid the episodes turns very unrealistic due to side effects on the traffic induced by IDM models that locally adapt to the context and consequently change their future trajectory. Concerning the learning policy, it receives at each decision step an observation o that consists in an egocentric representation of the local neighborhood and a command provided as a traffic free reference path to follow (Fig. 1). Each observation contains several fields like ego trajectory history for last 2 seconds, n-nearest neighbors configurations, local reference route, local lane corridor geometry based on lanelet graph and indicators like being off-road. The command is a high level description of the driving task to perform at least up to next transition of the MDP and is intended to lead to final destination. The action space consists in longitudinal and lateral shifts $a = (ds, dn)$ with respect to the local route to follow (Fig. 1) provided by a routing module which considerably reduce exploration complexity compared to egocentric displacements. Actions are sampled from a diagonal Gaussian distribution whose mean and variance are computed by the policy π_θ.

C. Training process

The algorithm use an expert database composed of a set of driving scenarios and their associate expert demonstrations stored in an offline buffer $\mathcal{B}_e$. In order to optimize the training objective we alternate between training the discriminator and training the policy with a specific training interplay balance[27]. Depending on the algorithm used for training i.e, GAIL,WGAIL-GP,WGAIL-SN, the discriminator D_ϕ maximize the distribution discrepancy as detailed in III-A between policy samples stored in an online buffer $\mathcal{B}_\pi$ and expert samples stored in $\mathcal{B}_e$ where samples consist in observation action pairs. Parallel data collection enables to collect multiple apprentice episodes on different scenarios to feed the online buffer. The policy is trained based on PPO algorithm [36] as detailed in III-A based on the current

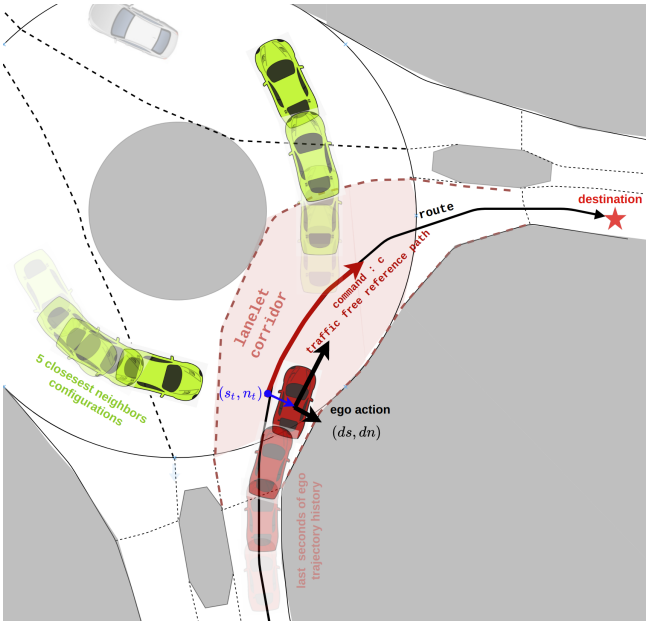


Fig. 1. Driving Scene: Ego agent takes its decision based on the configurations of the 5 nearest neighbors, local reference route, ego motion state,ego last 2 seconds trajectory,lanelet corridor geometry and the command encoded as the traffic free path to follow. Ego action encode longitudinal ds and lateral shift dn with respect to the local reference route.

training batch – built during data collection with rewards computed after discriminator update. In order to stabilize the training we use an horizon curriculum inspired from [4] that enables to increase progressively the simulation horizon based on an imitation cost threshold. Additionally, absorbing states inserted in the MDP as explained in III-B based on failure criteria like collision or distance threshold with respect to center lane prevents unrealistic transitions from filling the replay buffer.

IV. Experiments

In this section we detail experiments we realised to evaluate if AIL can recover realistic and safe driving policies from real data. We first details how we designed the training pipeline and how we compute our metrics. In second time, we give results about several experiments.

A. Implementation and metrics

1) Implementation of training pipeline : We use Rllib [24] based on the Ray framework [26] to implement our training pipeline (Fig. 2). We adopt the data flow perspective introduced in [23] to adapt Rllib PPO trainer into an AIL trainer endowed with a discriminator network that can recompute the reward from apprentice transitions (o_t, a_t) as depicted on Fig. 2. Rllib enables to make massive parallel simulations potentially distributed on several machines without the burden of data transfer management which is crucial for learning on multiple driving scenarios. The learning agent is controlled through Rllib while other traffic participants are controlled by simulator internal behaviour modules. The driving simulator is based on the Lanelet2 road map format

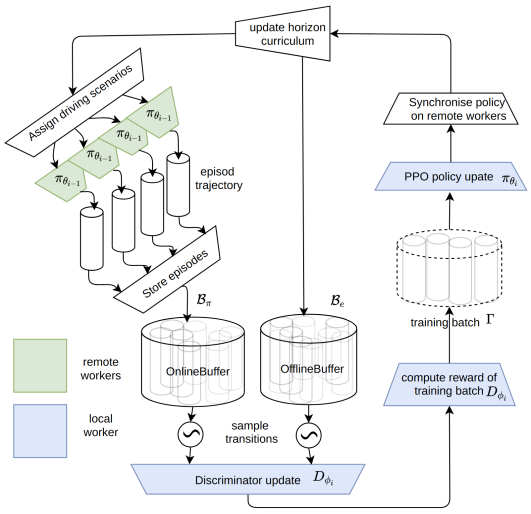


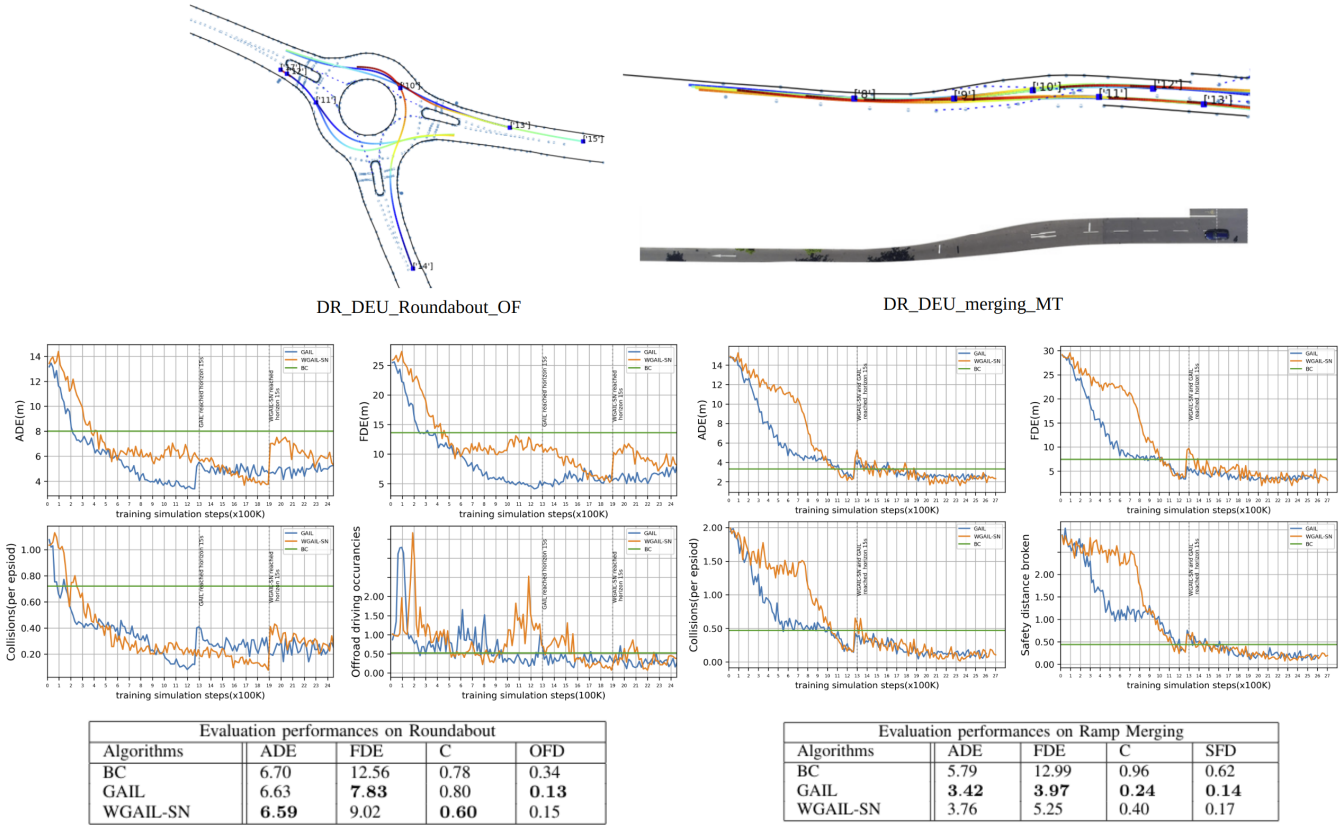
Fig. 2. Pipeline architecture : The training loop starts with policy parallel data collection on multiple workers from various driving scenarios and collected episode trajectories are stored in a online replay buffer. Depending on the interplay balance, discriminator training is launched on buffers and stopped appropriately. Next the rewards of the collected trajectories are updated with the discriminator. Finally, a PPO training step is applied on the set of collected trajectories called training batch and the whole process is repeated until convergence

[24] that is exploited by a routing module to provide a traffic free reference path given a destination based on the lanelet graph. It also enables to provide road structure representations as the lane corridor (Fig. 1) for the driving policy observation. As explained above, the expert database is generated by our custom expert database editor and contains a selection of driving scenarios with their associate demonstrations converted in the same observation and action space as the apprentice. We select driving scenarios based on several criteria like average speed and average number of traffic agents to avoid ambiguous or uninteresting situations where the expert is alone in the scene or are idle because parked on border of the road. Once extracted, demonstration are stored on the disk leveraging Rllib offline data interface and are selectively loaded through an offline buffer during training.

2) Imitation and safety metrics : Traffic simulation is concerned with realism and safety. In order to estimate how realistic the apprentice behaves with respect to expert demonstrations we use a set of imitation metrics inspired from [6] which consists in Average Distance Error in meters (ADE) with respect to associate expert trajectory and Final Distance Error (FDE) with respect to associate expert trajectory.Additionally, we evaluate safety of the apprentice thanks to three other metrics: average number of collisions per episode (C), average number of decision steps spent off road per episode (OFD) and average number of decision steps spent with safety distance broken per episode (SDB). We compute SDB as follows : at each decision step we check if the distance between ego and front or back neighbors $distance(a_{ego}, a_{F/R})$ is above a margin equal to $d_{margin} = 6$ meters.If the distance is lower than d_{margin} we assign a penalty $max(1 - distance(a_{ego}, a_{F/R})/d_{margin}, 0$ for the decision step otherwise we assign 0.As long as safety margin are kept, SDB equals 0 otherwise it grows up. For each of those metrics, we compute the average over either the whole training set either the whole evaluation set.Note that the decision period is based on sampling period of Interaction Dataset which is 100 ms.

B. Results

We first analyse performance of AIL algorithms on two different datasets extracted from Interaction Dataset. The First dataset is based on a map with a roundabout $DR\_DEU\_Roundabout\_OF$ and the second on map with a ramp Merging $DR\_DEU\_Merging\_MT$. We included images of a driving episode for both map on top of figure 3 where the color gradient indicates position of agent changing along time. For each dataset, we trained AIL algorithms and BC baseline on 100 base episodes of 15 seconds. We consider relatively small dataset because we ultimatly aim to generate new simulations on arbitrary target location which may be visited only few times by self driving vehicles. We subdivide base episodes into non overlapping episodes of 7.5 seconds for horizon curriculum that helps to stabilize training improvements. We first start to train AIL algorithm on episodes of 7.5 seconds and the metrics curve on figure 3 are computed relatively to this horizon for the first stage of the training and relatively to horizon 15s for the second stage of training. For all experiments the BC baseline is trained directly with full length episodes of 15s and thus we reported directly metrics of the BC baseline for horizon 15s in all curves of figure 3. In order to increase the simulation horizon we use an imitation cost that should stay bellow a fixed threshold equal to 3.8 for at least 10 training steps. The imitation cost consists in the sum of average distance error and early ending which counts how many decision steps are missing in the episode because agent failed prematurely. For Roundabout dataset, training curves 3 show that horizon 15 seconds is reached after 1.3 M transitions for GAIL and after 1.9 M transitions for WGAIL-SN while WGAIL-GP required 5 M transitions, hence we did not report its curves for clarity. On Ramp merging dataset,GAIL and WGAIL-SN reach horizon 15s after 1.3M transitions sampled. For both experiments, GAIL and WGAIL-SN outperformed the BC baseline on training set and significantly reduce the average number of collisions per episode by at least a factor two. We observe in training curves of figure 3 that agent on the roundabout tends to avoid collision while staying in the lane corridor 1 and that agent on Ramp Merging tends to avoid collision while keeping the safety distance margin which is the expected strategy to stay in safety. Lastly we reported evaluation results for both dataset in arrays at the bottom of figure 3.We evaluate our algorithms on both datasets with 25 new episodes of 25 seconds for each of them. We observe that GAIL and WGAIL-SN outperformed the BC baseline for Ramp Merging whereas only WGAIL-SN outperformed the baseline on Roundabout scenarios. We explain the relative efficiency of BC baseline on new roundabout scenarios from



Evaluation performances on Roundabout				
Algorithms	ADE	FDE	C	OFD
BC	6.70	12.56	0.78	0.34
GAIL	6.63	**7.83**	0.80	**0.13**
WGAIL-SN	**6.59**	9.02	**0.60**	0.15

Evaluation performances on Ramp Merging				
Algorithms	ADE	FDE	C	SFD
BC	5.79	12.99	0.96	0.62
GAIL	**3.42**	**3.97**	**0.24**	**0.14**
WGAIL-SN	3.76	5.25	0.40	0.17

Fig. 3. Experiments Results

evaluation experimental results depicted on figure 4 that shows that in average BC keeps a constant longitudinal displacement ds without any update for lateral displacement dn and with almost no reactions with respect to neighbors. This strategy enables to avoid some collisions by chance and our validation set reveals favorable for this strategy. In contrast, AIL algorithms requires a lot of training steps to properly control the lateral shift and need to experience collisions to let the discriminator understand that those states are undesirable .

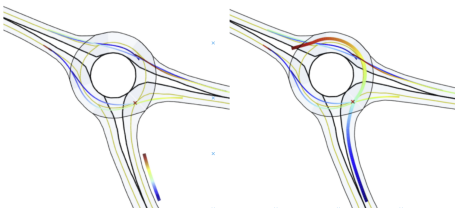


Fig. 4. BC behaviour tends in average to go forward along route : on the right side if we perturb its initial position, on an evaluation scenario it does not fix lateral offset and on the left side if we start at original initial position it does not avoid collision indicated by a red cross but goes straight

V. CONCLUSIONS

As demonstrated in experiments, AIL algorithms and more specifically GAIL and WGAIL-SN are able to recover expert behaviour on different type of interactive road networks for a reasonable simulation horizon of 15 seconds. Overall,whereas WGAIL-GP reveals much slower, GAIL and WGAIL-SN outperformed the BC baseline on the training set. We also show that WGAIL-SN is able to behave realistically and safely on new scenarios while GAIL achieve competitive results but sill collide too frequently. Future works may consider the multi agent learning setting with shared policies and a centralized critic such a way to obtain a fully interactive driving simulation. As a multi agent traffic often implies diverse behaviours we should also enforce some diversity constrains over the set of learnt policies such a way to capture various modes in demonstrations. As shown by GAIL evaluation results, driving policy robustness to out-of-training-distribution (OOD) [12] still constitute a main challenges and need to be fixed before traffic simulation can be reliably used for safety evaluation.

REFERENCES

[1] Marcin Andrychowicz et al. “What Matters In On-Policy Reinforcement Learning? A Large-Scale Empirical Study”. In: *CoRR* (2020).

[2] Szilárd Aradi. “Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles”. In: *IEEE Transactions on Intelligent Transportation Systems* (2020), pp. 1–20.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein GAN". In: *arXiv:1701.07875* (Dec. 2017).

[4] Feryal Behbahani et al. "Learning From Demonstration in the Wild". In: *2019 International Conference on Robotics and Automation (ICRA)*. May 2019, pp. 775–781.

[5] Julian Bernhard et al. "BARK: Open Behavior Benchmarking in Multi-Agent Environments". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Oct. 2020).

[6] Raunak P. Bhattacharyya et al. "Multi-Agent Imitation Learning for Driving Simulation". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2018, pp. 1534–1539.

[7] Raunak P. Bhattacharyya et al. "Simulating Emergent Properties of Human Driving Behavior Using Multi-Agent Reward Augmented Imitation Learning". In: *2019 International Conference on Robotics and Automation (ICRA)*. May 2019, pp. 789–795.

[8] Lorenzo Canese et al. "Multi-Agent Reinforcement Learning: A Review of Challenges and Applications". en. In: *Applied Sciences* 11.11 (Jan. 2021).

[9] Dong Chen et al. "Deep Multi-agent Reinforcement Learning for Highway On-Ramp Merging in Mixed Traffic". In: *arXiv preprint arXiv:2105.05701* (2021).

[10] Felipe Codevilla et al. "End-to-End Driving Via Conditional Imitation Learning". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. May 2018, pp. 4693–4700.

[11] Alexey Dosovitskiy et al. "CARLA: An open urban driving simulator". In: *Conference on robot learning*. PMLR. 2017, pp. 1–16.

[12] Angelos Filos et al. "Can Autonomous Vehicles Identify, Recover From, and Adapt to Distribution Shifts?" In: *arXiv:2006.14911 [cs, stat]* (Sept. 2020).

[13] Chelsea Finn, Sergey Levine, and Pieter Abbeel. "Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization". In: *International Conference on Machine Learning*. PMLR, June 2016, pp. 49–58.

[14] Jakob Foerster et al. "Counterfactual multi-agent policy gradients". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[15] Yang Gao et al. "Reinforcement Learning from Imperfect Demonstrations". In: (2019). arXiv: 1802.05313 [cs.AI].

[16] Jonathan Ho and Stefano Ermon. "Generative Adversarial Imitation Learning". In: *Advances in Neural Information Processing Systems*. Vol. 29. 2016.

[17] Junning Huang et al. "Learning a Decision Module by Imitating Driver's Control Behaviors". In: *arXiv:1912.00191 [cs, eess]* (May 2021).

[18] Zhiyu Huang, Jingda Wu, and Chen Lv. "Driving Behavior Modeling Using Naturalistic Human Driving Data With Inverse Reinforcement Learning". In: *IEEE Transactions on Intelligent Transportation Systems* (2021), pp. 1–13.

[19] Zhiyu Huang, Jingda Wu, and Chen Lv. "Efficient Deep Reinforcement Learning with Imitative Expert Priors for Autonomous Driving". In: *arXiv preprint arXiv:2103.10690* (2021).

[20] Ilya Kostrikov et al. "Discriminator-Actor-Critic: Addressing Sample Inefficiency and Reward Bias in Adversarial Imitation Learning". en. In: Sept. 2018.

[21] Daniel Krajzewicz et al. "SUMO (Simulation of Urban MObility) - an open-source traffic simulation". In: *4th Middle East Symposium on Simulation and Modelling*. Ed. by A. Al-Akaidi. 2002, pp. 183–187.

[22] Alex Kuefler et al. "Imitating driver behavior with generative adversarial networks". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. June 2017, pp. 204–211.

[23] Eric Liang et al. "Distributed Reinforcement Learning is a Dataflow Problem". In: *ArXiv* (2020).

[24] Eric Liang et al. "RLlib: Abstractions for Distributed Reinforcement Learning". In: *International Conference on Machine Learning*. July 2018, pp. 3053–3062.

[25] Takeru Miyato et al. "Spectral Normalization for Generative Adversarial Networks". en. In: Feb. 2018.

[26] Philipp Moritz et al. "Ray: A Distributed Framework for Emerging {AI} Applications". In: 2018, pp. 561–577.

[27] Manu Orsini et al. "What Matters for Adversarial Imitation Learning?" In: *arXiv:2106.00672 [cs]* (June 2021).

[28] Blazej Osinski et al. "CARLA Real Traffic Scenarios - novel training ground and benchmark for autonomous driving". In: *ArXiv* abs/2012.11329 (2020).

[29] Fabio Pardo et al. "Time Limits in Reinforcement Learning". en. In: *International Conference on Machine Learning*. July 2018, pp. 4045–4054. (Visited on 09/06/2021).

[30] Fabian Poggenhans et al. "Lanelet2: A high-definition map framework for the future of automated driving". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (2018).

[31] Siddharth Reddy, Anca D. Dragan, and Sergey Levine. "SQIL: Imitation Learning via Reinforcement Learning with Sparse Rewards". In: (2019). arXiv: 1905.11108.

[32] Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. "Deep Imitative Models for Flexible Inference, Planning, and Control". en. In: Sept. 2019.

[33] Stephane Ross, Geoffrey Gordon, and Drew 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*. JMLR Workshop and Conference Proceedings, June 2011, pp. 627–635.

[34] Axel Sauer, Nikolay Savinov, and Andreas Geiger. "Conditional Affordance Learning for Driving in Urban Environments". en. In: *Conference on Robot Learning*. PMLR, Oct. 2018, pp. 237–252.

[35] Dhruv Mauria Saxena et al. “Driving in Dense Traffic with Model-Free Reinforcement Learning”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. ISSN: 2577-087X. May 2020, pp. 5385–5392.

[36] John Schulman et al. “Proximal Policy Optimization Algorithms”. In: *arXiv:1707.06347 [cs]* (Aug. 2017).

[37] Jonathan Spencer et al. “Feedback in Imitation Learning: The Three Regimes of Covariate Shift”. In: *arXiv:2102.02872 [cs, stat]* (Feb. 2021).

[38] Simon Suo et al. “TrafficSim: Learning to Simulate Realistic Multi-Agent Behaviors”. In: *arXiv:2101.06557 [cs]* (Jan. 2021).

[39] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. “Congested traffic states in empirical observations and microscopic simulations”. In: *Physical Review E* 62.2 (Aug. 2000).

[40] Jingke Wang et al. “Learning hierarchical behavior and motion planning for autonomous driving”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2020, pp. 2235–2242.

[41] Pin Wang et al. “Decision Making for Autonomous Driving via Augmented Adversarial Inverse Reinforcement Learning”. In: *arXiv:1911.08044 [cs]* (Mar. 2021).

[42] Bernhard Wymann et al. “Torcs, the open racing car simulator”. In: *Software available at http://torcs. sourceforge. net* 4.6 (2000), p. 2.

[43] Huang Xiao et al. “Wasserstein Adversarial Imitation Learning”. In: *arXiv:1906.08113 [cs, stat]* (June 2019).

[44] Huang Xiao et al. “Wasserstein Adversarial Imitation Learning”. In: *CoRR* abs/1906.08113 (2019).

[45] Wei Zhan et al. “INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps”. In: *arXiv:1910.03088 [cs, eess]* (Sept. 2019).

[46] Ming Zhang et al. “Wasserstein Distance guided Adversarial Imitation Learning with Reward Shape Exploration”. In: *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*. Nov. 2020, pp. 1165–1170.

[47] Guanjie Zheng et al. “Objective-aware Traffic Simulation via Inverse Reinforcement Learning”. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. Montreal, Canada, Aug. 2021, pp. 3771–3777.

[48] Ming Zhou et al. “SMARTS: Scalable Multi-Agent Reinforcement Learning Training School for Autonomous Driving”. In: *arXiv:2010.09776 [cs, eess]* (Oct. 2020).

[49] Zeyu Zhu and Huijing Zhao. “A Survey of Deep RL and IL for Autonomous Driving Policy Learning”. In: *arXiv:2101.01993 [cs]* (Jan. 2021).

[50] Brian D. Ziebart et al. “Maximum entropy inverse reinforcement learning”. In: *Proceedings of the 23rd national conference on Artificial intelligence - Volume 3*. AAAI’08. Chicago, Illinois, July 2008, pp. 1433–1438.