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], Christophe Sabourin[1]
[1] *Univ Paris Est Creteil, LISSI, F-77567 Lieusaint, France*
[2] *IoV team, Paris Research Center, Huawei Technologies France*
{*yann.koeberle1,stefano.sabatini,dzmitry.tsishkou*} *@huawei.com, sabourin@u-pec.fr*

Keywords: Driving simulation,Learning from demonstrations, Adversarial Imitation Learning.

Abstract: Traffic simulation has gained a lot of interest for autonomous driving companies for qualitative safety evaluation of self driving vehicles. In order to improve self driving systems from synthetic simulated experiences, traffic agents need to adapt to various situations while behaving as a human driver would do. However, simulating realistic traffic agents is still challenging because human driving style cannot easily be encoded in a driving policy. Adversarial Imitation learning (AIL) already proved that realistic driving policies could be learnt from demonstration but mainly on highways (NGSIM Dataset). Nevertheless, traffic interactions are very restricted on straight lanes and practical use cases of traffic simulation requires driving agents that can handle more various road topologies like roundabouts, complex intersections or merging. In this work, we analyse how to learn realistic driving policies on real and highly interactive driving scenes of Interaction Dataset based on AIL algorithms. We introduce a new driving policy architecture built upon the Lanelet2 map format which combines a path planner and an action space in curvilinear coordinates to reduce exploration complexity during learning. We leverage benefits of reward engineering and variational information bottleneck to propose an algorithm that outperforms all AIL baselines. We show that our learning agent is not only able to imitate humane like drivers but can also adapts safely to situations unseen during training.

1 INTRODUCTION

For real world deployment, self driving systems require quantitative safety guarantees in presence of real human drivers. Traffic simulation appears as a crucial tool to continuously provide statistics of driving performances on arbitrary number of locations and scenarios (Scheel et al., 2022) without endangering human drivers. Simulation enables to expose the Self Driving Vehicle (SDV) to various interactive situations with controlled scenario modifications so that critical failures can be identified. However the reality gap between simulated and real driving behaviours can result in positive improvements in simulation whereas dramatic issues could still occur in real settings.

Animating each traffic agent during simulation requires a decision process called driving policy which can be designed in several different ways. Heuristic based simulated agents are controlled with explicit rules and can easily perform maneuvers as changing lane or car following but generated trajectories are statistically different from trajectories generated by humans (Treiber et al., 2000). Learning based methods offer more flexibility to adjust driving behaviours in various situations (Suo et al., 2021; Scheel et al., 2022). Reinforcement Learning enables to learn through interactions with a simulator where the learning agent is penalized for catastrophic failures. The main limitation of this approach comes from the fact that the true reward representative of human driving style is unknown and expensive to design because it depends on various human preferences (Knox et al., 2021). In contrast, supervised methods enables to directly leverage real demonstrations and thus can capture more naturalistic driving behaviors. The most simple supervised learning method is Behavior Cloning which maximizes the likelihood of expert actions on a training set. This approach has big limitations in long term simulations because it suffers from errors compounding and poor generalization capabilities (Codevilla et al., 2018). Alternatively, Adversarial Imitation Learning (AIL) enables to exploit real data through simulation interactions and deliberately exposes the policy to situations out of expert experiences. AIL provides a guidance through a data-driven reward that the learning agent is expected to maximise during training which significantly reduces

catastrophic failures and helps the agent to get closer to the expert trajectory. Several works (Ho and Ermon, 2016; Kuefler et al., 2017) already proved that realistic driving policies can be learnt with AIL on highways scenarios but interactions on straight lanes are not representative of the driving task complexity that encompasses challenges of various road topologies. Some complex intersections, roundabouts or merging with numerous traffic agents reveals particularly difficult to handle because it induces complex interplay between agents and slight trajectory offsets could lead to crashes. Another difficulty encountered by AIL algorithms is the ability to explore efficiently in the long term in order to find a correct human-like behaviors. The driving scene does not allow arbitrary displacements and exploring naively the plane with incremental shifts could quickly lead the agent to absurd situations like going off road.

In this work, we investigate to which extent AIL algorithms can be used for learning human-like driving policies able to adapt to new situations for long horizon simulation. Our main contributions are:

- An analysis of imitation performances of driving policies trained with Adversarial Imitation learning algorithms on highly interactive driving scenes extracted from Interaction Dataset.
- An action space for parsimonious exploration based on curvilinear coordinates with respect to a reference path generated by a planner that exploit the lanelet2 map format.
- An AIL algorithm that combines the benefits from variationnal information bottleneck and reward engineering.

We start to review main approaches for learning realistic driving policies in Section 2. Subsequently, Section 3 explains how we learn a driving policy from demonstration and describes the neural networks architectures. Finally, we detail experiments settings and report our results in Section 4.

2 RELATED WORKS

We review main traffic simulation approaches in Section 2.1 before studying more in-depth how driving polices can be learnt from demonstrations in Section 2.2.

2.1 Traffic simulation

Traffic simulation has the potential to accelerate the development of Self Driving Vehicle (SDV) (Cao et al., 2020).The main challenge consists in guaranteeing that interacting with simulated traffic agents could provide valuable experiences. Heuristic based traffic simulator (Lopez et al., 2018) enables to generate traffic patterns with very few crashes but usually largely differ from real human traffic. Rule based driving policies like IDM (Treiber et al., 2000) designed for longitudinal control on highways or Mobil (Kesting et al., 2007) designed for lane changes cannot handle arbitrary road topologies where various trajectory profiles are plausible like complex intersections. Learning based methods offer more flexibility to adapt the driving policy to various scenes because they can leverage human demonstrations.

Reinforcement learning offers the possibility to design custom driving behaviours based on hand crafted rewards so that the driving agent can adapt to various situation based on explicit feedbacks like incentive to slow down at intersections, penalty in case of over-speeding, etc. Several works (Chen et al., 2021; Sharma and Sharma, 2021) show that RL driving policies are able to avoid safety critical failures in urban scenarios but the main limitation comes from the fact that the hand crafted reward does not constitute the true driving performance but acts as a proxy (Knox et al., 2021). Consequently a policy that has high expected return is not guaranteed to behave as a human would do which may lead to unrealistic traffic patterns. In order to drive as a human, data based methods leverage the huge amount of real driving demonstrations available to learn trajectories that match human preferences. Simnet (Bergamini et al., 2021) or TrafficSim (Suo et al., 2021) builds upon prediction models to learn a centralized policy to animate each agent in the driving scene, but a centralized policy can not easily scale to large scenarios because it has to handle exponential number of interactions in the number of agents for long simulation horizons. In contrast, the RAIL algorithm (Bhattacharyya et al., 2019) formulates traffic simulation as a multi agent imitation problem and shows that realistic traffic patterns can be learnt but it also assumes that all driving agents are controlled by the same policy, driven by the same reward neglecting the diversity of real road users. Even if complementary approaches based on multi agent reinforcement enables to learn multiple policies with a centralized critic it is still very challenging to maintain training stability and convergence guarantees (Lyu et al., 2021). Consequently we consider traffic simulation as a decentralized problem which consists in learning single agent driving policies behaving close to human drivers for long simulation horizons.

2.2 Learning to drive from demonstration

In order to learn a realistic driving policy from human demonstrations it is possible to formulate the driving task as a supervised problem called Behavioural Cloning (BC) with the objective of matching actions generated by the learning policy and the ones selected by the expert (Codevilla et al., 2018). This technique suffers from compounding errors during test time because BC is trained offline in open loop with i.i.d samples whereas the closed loop evaluation of the policy induces a distributional shift due to sub optimal past decisions. To compensate deviations in closed loop, it is possible to complete the training dataset by querying an interactive expert but online supervision is highly constraining because it either requires a human in the loop or a specific expert system (Ross et al., 2011). An alternative is to resort to uncertainty-based regularization with ensemble of expert policies to reduce deviations from expert trajectories distribution (Brantley et al., 2019). While this technique encourages the learner to stay close to the expert trajectories, it cannot provide explicit guidance out of distribution and remains computationally expensive to estimate accurately.

Adversarial Imitation Learning (AIL) offers more flexibility to guide the policy and several works already showed that generative adversarial imitation learning is able to learn realistic driving policies on highways scenario despite collisions still happens (Ho and Ermon, 2016; Kuefler et al., 2017). More importantly, standard AIL algorithms suffer from training instabilities due to the asymmetric competition between discriminator and the generator. The discriminator quickly tends to get too accurate which leads to uninformative gradients for the policy which struggles to match the expert driving strategy. To balance the performance between the generator and the discriminator, VAIL(Peng et al., 2018) enforces a constraint on the mutual information between the input observation and the discriminator's internal representation which prevents the accuracy from getting too high. Even if this method enables to maintain informative guidance for the policy, the discriminator cannot understand the causal structure of the driving task which may lead to crashes once the policy is too far from training distribution (De Haan et al., 2019).

To help the discriminator to avoid catastrophic failures, domain knowledge could be used to feed the discriminator with high level semantic signals like off-road driving or collision indicators (Wang et al., 2021). However the value assigned to the signal is subjective and case sensitive which can deter the discriminator to exploit relevant features in the state action pair originally provided as input. To limit side effects on the discriminator, one can just add a penalty to the discriminator reward when the policy goes in undesired situations (Bhattacharyya et al., 2019). We build upon those recent advances in AIL to learn a realistic driving policy that can drive on new scenarios with better safety.

3 LEARNING A REALISTIC DRIVING POLICY

We formulate the driving task in Section 3.1 before detailing how to learn realistic driving policies in Section 3.2.

3.1 Problem setting

We aim to learn realistic driving policies for animating traffic agents in simulation. Traffic simulation consists in generating driving episodes from predefined driving scenarios. A driving scenario $S = (\mathcal{M}, \mathcal{F}, \rho_0, H, \mathcal{G})$ is composed of a simulation horizon H, a bounded map of a road-network $\mathcal{M}$ and a traffic flow $\mathcal{F}$ that spawns traffic agents at specific time on the map according to an initial state distribution ρ_0 and a set of destinations $\mathcal{G}$. We consider decentralized traffic simulation where each agent is animated by its own driving policy assigned by the traffic flow. As real driving episodes are likely to include diverse policies, learning them simultaneously can turn highly unstable. Consequently, we propose to learn a single agent driving policy per episode called actor policy while other agents in the scene called workers are controlled by fixed driving policies. In order to learn a single agent driving policy, we formulate the task as Partially Obsersable Markov Decision Process (POMDP): $(\mathcal{S}, O, \mathcal{A}, \mathcal{T}, \mathcal{R})$. We condition the policy π on a goal g provided at initialization by the traffic flow to specify the driving task. At each decision step, the policy gets an ego centric observation o of the scene and take an action a to reach its goal. The observation is provided by an observer model $O : \mathcal{S} \mapsto O$ that operates on the driving scene states $\mathcal{S}$. The driving scene state s which encompasses the actor state as well as traffic workers states evolves according to the transition dynamic $\mathcal{T}$ which take into account the actor action and implicitly traffic workers decisions. The reward process $\mathcal{R}$ is composed of two components based on domain knowledge and human demonstrations as detailed in Section 3.2.1.

3.2 Learning from real driving data

We aim to learn a single agent driving policy π_θ from real demonstrations parametrized by a neural network. We build upon adversarial imitation learning and train jointly a policy and a discriminator so that the policy generates expert like trajectories while the discriminator is trained to distinguish the policy and expert trajectory samples. Similarly to GAIL (Ho and Ermon, 2016), we aim to solve the following problem:

$$min_{\pi_\theta} max_{D_\varphi} \mathbb{E}_{(s,a)\sim\rho_{\pi_e}}[log(D_\varphi(s,a))] + \mathbb{E}_{(s,a)\sim\rho_{\pi_\theta}}[log(1-D_\varphi(s,a))] - \lambda H(\pi_\theta) \quad (1)$$

where $H(\pi_\theta)$ denotes the policy causal entropy and $\rho_{\pi_e}, \rho_{\pi_\theta}$ the marginal observation action distributions induced by expert and policy respectively. In order to solve this bi-level optimization problem (Liu et al., 2021), we alternate between optimizing the discriminator and optimizing the policy. The training procedure, depicted in Figure 1, is decomposed in three main steps repeated until the maximum number of training iterations is reached. The first step consists in collecting in parallel, multiple driving episodes with the current policy π_θ and gathering them in the policy buffer $\mathcal{B}_\pi$. Subsequently, we train the discriminator on recent policy samples and expert demonstrations as described in Section 3.2.2. The third steps consists in updating the policy based on Proximal Policy Optimization (PPO) (Schulman et al., 2017) on a training batch extracted from $\mathcal{B}_\pi$ as detailed in Section 3.2.1. Note that the simulation horizon is progressively increased during training along with our policy improvements. In the two following sections, we explain how additionally we augment the usual AIL training procedure summarized above to leverage domain knowledge (Section 3.2.1) and to regularize the discriminator (Section 3.2.2).

3.2.1 Updating the policy

The policy is trained with a reward signal r that is decomposed in two terms a synthetic reward r_S based on domain knowledge and data driven term r_{D_φ} computed with the discriminator D_φ as detailed in Section 3.2.2.

$$r = r_S + r_{D_\varphi} \quad (2)$$

The synthetic reward is itself composed of two terms: one that penalizes collisions when they occur r_{col} and one that favors forward displacement r_{ds}:

$$r_s = r_{col} + \alpha r_{ds} \quad (3)$$

While the synthetic reward enables to avoid crashes and motionless behaviour, the data driven is expected

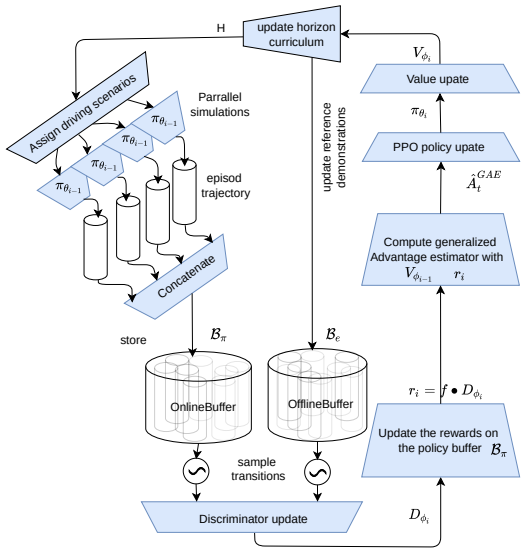

Figure 1: Driving policy training procedure based on adversarial imitation learning.

to drive the policy close to humane trajectories. Since PPO algorithm is used to update the policy, we leverage the Generalized Advantage Estimator to modulate the bias-variance trade off of the policy gradient. For each policy sample (s_t, a_t, r_t) and its associate future trajectory $\tau_t = [(s_t, a_t), ..., (s_{H-t}, a_{H-t})$ where $r_t = r_S(o_t, a_t) + r_{D_\varphi}(o_t, a_t)$, we estimate the GAE $\hat{A}_t^{GAE}(s_t, a_t)$ as detailed in (Schulman et al., 2015). Subsequently, PPO objective $max_\theta J^{PPO}(\theta)$ can be optimized with a clipping mechanism that tries to avoid abrupt changes of the policy parameters limited by the threshold ε_π.

$$J^{PPO}(\theta) = \mathbb{E}_{(a_t,o_t)\sim\pi_{\theta_{old}}}[min(\mathcal{L}^\pi(\theta), \mathcal{L}^\pi_{clip}(\theta))]$$
$$\mathcal{L}^\pi(\theta) = \mu_t(\theta)\hat{A}_t^{GAE}(o_t, a_t)$$
$$\mathcal{L}^\pi_{clip}(\theta) = clip(\mu_t(\theta), 1-\varepsilon_\pi, 1+\varepsilon_\pi)\hat{A}_t^{GAE}(o_t, a_t)$$

3.2.2 Updating the discriminator

In the original GAIL algorithm, the discriminator tends to get too accurate too quickly during training which limits the progresses of the policy. Indeed, as the classification loss is easier to optimize compared to PPO loss that has high variance, the discriminator output progressively gets close to zero for policy samples. Consequently, the data driven reward $r_{D_\varphi} = log(D_\varphi(o,a)+\varepsilon) - log(1-D_\varphi(o,a)+\varepsilon)$, bounded with $\varepsilon = 10^{-8}$ to avoid exploding gradients, saturates at a negative value instead of guiding the policy. In order to balance the performance of the policy and the discriminator, we complement the original

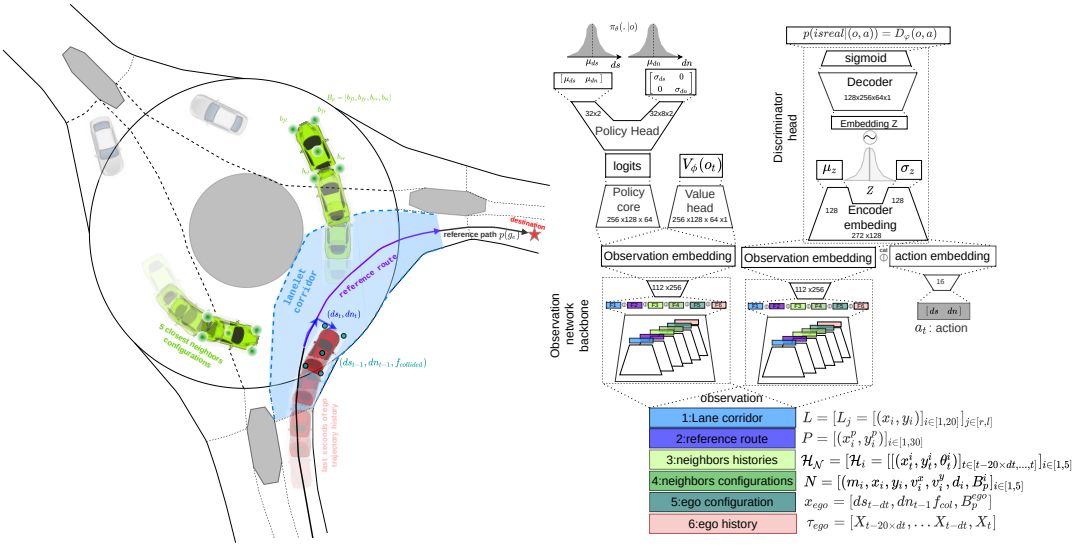

Figure 2: On the left side, illustration of the components of the observation provided to the policy as well as the action generated by the policy and on the right side, architecture of the discriminator-actor-critic neural network.

problem 3.2 by constraining the information flow in the discriminator by means of an information bottleneck. We enforce a constraint on the mutual information $I((O,A),Z)$ between the input of the discriminator [1] and its internal representation Z to modulate the discriminator's accuracy. Consequently the discriminator D_φ is composed of two parts: an encoder E_φ that maps an observation action pair (o,a) to a stochastic encoding $z \sim E(z|(o,a))$ and a decoder D_φ that classifies samples drawn from the encoder distribution as human-like or not. The encoder outputs the mean $\mu_z(\phi)$ and the standard deviation $\sigma_z(\phi)$ of a multi variate Gaussian distribution $\mathcal{N}(\mu_z(\phi), \sigma_z(\phi))$ that enables to define the constrain on the information flow. The algorithm called VAIL optimizes the standard GAIL objective while maintaining the mutual information constrain by introducing a Lagrangian multiplier β updated with dual gradient ascent (Peng et al., 2018).

$$min_\varphi max_\beta J^{Disc}(\varphi) + \beta M(\varphi) \quad (4)$$

The discriminator objective $J^{Disc}(\varphi)$ is optimized with mini batch gradient descent for K epochs on policy and expert buffers.

$$J^{Disc}(\varphi) = \mathbb{E}_{(o,a)\sim\mathcal{B}_\mathcal{E}}[\mathbb{E}_{z\sim E_\varphi(z|(o,a))}[log(D_\varphi(z)]] + \mathbb{E}_{(o,a)\sim\mathcal{B}_\pi}[\mathbb{E}_{z\sim E_\varphi(z|(o,a))}[log(1-D_\varphi(z)]] \quad (5)$$

As the mutual information $I((O,A),Z)$ cannot be easily computed due to the difficulty to estimate the marginal distribution $p(z)$, we approximate $p(z)$ with a multivariate normal $r(z) = \mathcal{N}(0,I)$. Consequently we obtain an upper bound denoted $M(\varphi)$ on $I((O,A),Z)$ which we use as a regularizer.

$$M(\varphi) = \mathbb{E}_{(o,a)\sim\tilde{\pi}}[KL[E_\varphi[z|(o,a)]||r(z)]] - I_c \quad (6)$$

where $I_c = 0.5$ denotes a threshold value. We update the Lagrangian multiplier β with dual gradient ascent in order to maintain the constrain on the mutual information as detailed in (Peng et al., 2018).

Once the discriminator is trained, the data driven reward is computed with the discriminator output $D_\varphi(o,a)$ that represents the probability that (o,a) was generated from an expert. In order to reduce the reward bias induced by purely negative or positive rewards, we use the reward formula introduced in (Kostrikov et al., 2018):

$$r_{D_\varphi}(o,a) = log(D_\varphi(o,a)) - log(1-D_\varphi(o,a)) \quad (7)$$

4 Driving policy

We detail in Section 4.1 how the driving policy observes and takes action before describing our neural network architecture in Section 4.2.

4.1 Observation and Action space

The driving policy $\pi(a|o,p(g))$ is both conditioned on an ego-centric observation of the driving scene and on a reference path $p(g)$ that leads to its destination. The reference path $p(g)$ is computed by a top level path

[1] The discriminator input is an Observation Action pair, hence the mutual information is written $I((O,A),Z)$.

metrics	Merging			Roundabout			Intersection		
scenarios	ADE-5(m)	ADE-15(m)	CR(%)	ADE-5(m)	ADE-15(m)	CR(%)	ADE-5(m)	ADE-15(m)	CR(%)
BC	8.18	14.26	75	6.29	14.16	79	4.71	11.50	46.5
GAIL	4.20	6.67	30	3.20	5.98	41	3.95	7.22	28
SGAIL	3.78	7.5	10	3.23	5.13	37	3.96	7.26	26
SVAIL	**3.37**	**5.34**	**10**	**2.75**	**5.04**	**31**	**3.59**	**6.49**	**25**

Table 1: Comparison of imitation and safety metrics of driving policies evaluated on different scenes: roundabout(R), intersection(I), merging(M)

planner independently from the scene context leveraging the lanelet2 map format of the driving scene. In order to move as much as possible along $p(g)$ we define the action space based on curvilinear coordinates (s,n) with respect to $p(g)$ as depicted in Figure 2. We enforced the policy to output longitudinal and lateral (ds,dn) shifts at each decision step which enables to explore parsimoniously plausible trajectories and to stop simulation when actor gets too far from the route. In contrast, controlling directly the forward acceleration and the turn rate would not guide the displacement of the agent toward the goal during exploration but let it easily deviate from $p(g)$.

In order to infer appropriate moves, we provide an observation composed of several contextual components of the driving scene in a vectorized format. We provide information about the map like the lane corridor and the reference route. The Lane corridor $L = [L_r = [(x_i,y_i)]_{i\in[1,20]}, L_l = [(x_i,y_i)]_{i\in[1,20]}]$ is composed of the right and left borders coordinates relative to the actor of the drivable area [2] in a 10 meter radius around the reference path. The reference route is a piece of the reference path $p(g)$ 10 meters in front of the actor. We also provide information about the traffic context with five nearest neighbors configurations $N = [(m_i,x_i,y_i,v_i^x,v_i^y,d_i,B_{p]}^i]_{i\in[1,5]}$ and their trajectory histories for the last 2 seconds relative to current actor position $\mathcal{H}_{\mathcal{N}} = [\mathcal{H}_i = [(x_t^i,y_t^i,\theta_t^i]_{t\in[t-20\times dt,\ldots,t]}$. Each neighbor configuration contains a mask m to indicate if the i-th neighbor exists, its position relative to the actor x_i,y_i, its speed vector v_i^x,v_i^y, relative distances d_i as well as its spatial extension with the four border points coordinates on front, rear, left and right side $B_p^i = [b_{fl}^i,b_{fr}^i,b_{rr}^i,b_{rl}^i]$. Lastly, we add two components to specify the actor current state x_{ego} and the coordinates of its trajectory for the last 2 seconds $\tau_{ego} = [X_{t-20\times dt},\ldots X_{t-dt},X_t]$. The actor configuration contains, its last action $[ds_{t-dt},dn_{t-dt}]$, a collision flag f_{col}, and its spatial extension in terms of border points coordinates $B_p^{ego} = [b_{fl}^{ego},b_{fr}^{ego},b_{rr}^{ego},b_{rl}^{ego}]$.

[2]The drivable area is extracted based on the sequence of lanelets (Poggenhans et al., 2018) that constitutes the reference path $p(g)$.

4.2 Neural network Architecture

The driving policy π_θ is implemented with a neural network that parametrizes the next action $[ds_t,dn_t]$ with two independent gaussian distributions whose mean and variance are both learned. The network first embeds each observation components with specific sub networks built upon Fully Connected (FC) layers and generates a set of observation features $\mathcal{F} = [F_i]_{i\in[1,..,6]}$ depicted with colored rectangles in Figure 2. In order to handle the variable number of neighbors in the scene, individual configuration and history are embedded for each neighbor with two separate networks shared for all neighbors. The feature vectors F_3,F_4 that represent neighbors histories and neighbors configurations are computed by summing individual embeddings when the agent exists as indicated by a mask. At the end of the Observation network backbone, observation features are combined by concatenation and a FC layer computes the final observation embedding vector. The policy and value heads composed of consecutive FC layers share the same observation embedding vector but the value gradients are not back-propagated through the backbone which would perturb the policy for the next data collection. Regarding the discriminator, it shares the same architecture for computing the observation embedding vector but has a separate network. The encoder and decoder networks are built with consecutive (FC) layers. Note that z is sampled using the reparametrization trick : $z = \mu_z + \varepsilon.\sigma_z$ with $\varepsilon \sim \mathcal{N}(0,I)$ during discriminator updates whereas the mean μ_z is directly used to feed the decoder D_φ when the reward is computed for PPO updates.

5 EXPERIMENTS

5.1 Simulation Dataset and metrics

In order to learn a driving policy from real demonstrations through interactions, we build a driving simulator on top of driving scenes extracted from the Interaction dataset (Zhan et al., 2019). The dynamics of traffic workers is approximated by replaying their original trajectories during simulations because

we ignore how to animate them realistically on new situations. We focus on three different maps with high number of interactions: the roundabout called *DR_DEU_Roundabout_OF*, the intersection called *DR_USA_Intersection_EP*0 and the merging called *DR_DEU_Merging_MT*. Since we aim to imitate the driving style of road users on specific spots we train our driving policy separately on three training sets for each map. For each of the training set, we extracted 200 driving scenarios for each of the following temporal horizons: 2.5, 5, 7.5, 10, 12.5 and 15 seconds which gives a total of 1200 training scenarios per area[3]. The three associate validation sets are each composed of 128 new real scenarios of 15 seconds in order to focus on long term simulation performances. In order to evaluate the imitation of the actor driving policy we compute the Average Distance Error with respect to the human demonstration (ADE) after 5 and 15 second of simulation. We also report the percentage of episodes that contains a collision (CR) to estimate if the driving behaviour is safe.

5.2 Results

We analyse the performances of different AIL algorithms for learning realistic driving policy based on real driving demonstration of the validation set. We trained several baselines that we list below:

- **BC**: implements standard behavioural cloning that only exploit expert demonstrations (Codevilla et al., 2018).
- **GAIL**: implements a standard implementation of GAIL algorithm without the variational information bottleneck (Ho and Ermon, 2016).
- **SGAIL**: implements GAIL algorithm whose reward is augmented by the addition of a synthetic component $r_S = r_{col} + 0.1 r_{ds}$ as detailed in Section 3 inspired from (Bhattacharyya et al., 2019). The penalty equals $r_{col} = -2$ when the collision occurs and the bonus for longitudinal displacement $r_{ds} = max(1.0, \frac{ds}{ds_{max}})$ depends on the maximal speed allowed $ds_{max} = 50 km/hour$.
- **SVGAIL**: implements the VAIL algorithm detailed in Section 3.2 and adds the synthetic reward the same way as *SGAIL*.

We trained all those baselines with the neural network architecture detailed in Section 4.2 on the demonstrations of each training set separately. We compare imitation and safety metrics on the validation set and we report the metrics in Table 1. We first compare GAIL performances with Behaviour Cloning that does not learn through simulation interactions. We observe that BC suffers from compounding errors with high ADEs and tends to collide on a majority of scenarios. GAIL outperforms BC performances for long term imitation and gets significantly safer than BC with a much lower rate of episode with collisions. As GAIL still often collides it could be beneficial to penalize it more intensively with a synthetic reward for this specific event. We observe that *SGAIL* successively improves safety performance of GAIL while keeping similar imitation metrics. The results of *SVAIL* shows that imitation performances can further been improved compared to *SGAIL* by modulating the accuracy of the discriminator. We conclude that original GAIL can significantly benefit from modifications proposed in *SGAIL* and *VAIL*.
Comparing the ADE-15 variations of *SVAIL* between the three maps, we note that its is harder to imitate human driving style on scenarios built on the intersection. This can be explained by the fact that the interaction map contains more lanes and junctions than the roundabout and the merging with numerous agent spawning locations encoded in the initial state distribution ρ_O[4]. Consequently the expert trajectory distribution on the intersection has a bigger support which is more difficult to learn both for the discriminator and the policy.

6 CONCLUSIONS

In this work, we analyse imitation and safety performances of AIL algorithms on real and highly interactive scenarios extracted from Interaction Dataset. We propose an efficient action space based on a reference path and curvilinear coordinates which enables to explore plausible human trajectories parsimoniously. We propose an advanced version of the VAIL algorithm that exploits a synthetic reward based on domain knowledge and the lanelet2 map format for observation extraction to better guide the policy during training. We show that our algorithm outperforms all the baselines in imitation and safety on driving scenarios unseen during training. For future works, we plan to better control the trade off between human imitation and safe driving which still limits practical applications.

[3]This enables to extract as much demonstrations as possible even if some agents do not stay a long time on the recorded scene

[4]The reader can refer to Interaction Dataset description (Zhan et al., 2019) to observe the difference.

REFERENCES

Bergamini, L., Ye, Y., Scheel, O., Chen, L., Hu, C., Del Pero, L., Osiński, B., Grimmett, H., and Ondruska, P. (2021). Simnet: Learning reactive self-driving simulations from real-world observations. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5119–5125. IEEE.

Bhattacharyya, R. P., Phillips, D. J., Liu, C., Gupta, J. K., Driggs-Campbell, K., and Kochenderfer, M. J. (2019). Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 789–795. IEEE.

Brantley, K., Sun, W., and Henaff, M. (2019). Disagreement-regularized imitation learning. In *International Conference on Learning Representations*.

Cao, Z., Bıyık, E., Wang, W. Z., Raventos, A., Gaidon, A., Rosman, G., and Sadigh, D. (2020). Reinforcement learning based control of imitative policies for near-accident driving. *arXiv preprint arXiv:2007.00178*.

Chen, J., Li, S. E., and Tomizuka, M. (2021). Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*.

Codevilla, F., Müller, M., López, A., Koltun, V., and Dosovitskiy, A. (2018). End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4693–4700. IEEE.

De Haan, P., Jayaraman, D., and Levine, S. (2019). Causal confusion in imitation learning. *Advances in Neural Information Processing Systems*, 32.

Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.

Kesting, A., Treiber, M., and Helbing, D. (2007). General lane-changing model mobil for car-following models. *Transportation Research Record*, 1999(1):86–94.

Knox, W. B., Allievi, A., Banzhaf, H., Schmitt, F., and Stone, P. (2021). Reward (mis) design for autonomous driving. *arXiv preprint arXiv:2104.13906*.

Kostrikov, I., Agrawal, K. K., Dwibedi, D., Levine, S., and Tompson, J. (2018). Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. *arXiv preprint arXiv:1809.02925*.

Kuefler, A., Morton, J., Wheeler, T., and Kochenderfer, M. (2017). Imitating driver behavior with generative adversarial networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 204–211. IEEE.

Liu, R., Gao, J., Zhang, J., Meng, D., and Lin, Z. (2021). Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 2575–2582. IEEE.

Lyu, X., Xiao, Y., Daley, B., and Amato, C. (2021). Contrasting centralized and decentralized critics in multi-agent reinforcement learning. *arXiv preprint arXiv:2102.04402*.

Peng, X. B., Kanazawa, A., Toyer, S., Abbeel, P., and Levine, S. (2018). Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow. *arXiv preprint arXiv:1810.00821*.

Poggenhans, F., Pauls, J.-H., Janosovits, J., Orf, S., Naumann, M., Kuhnt, F., and Mayr, M. (2018). Lanelet2: A high-definition map framework for the future of automated driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1672–1679. IEEE.

Ross, S., Gordon, G., and Bagnell, D. (2011). 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*, pages 627–635. JMLR Workshop and Conference Proceedings.

Scheel, O., Bergamini, L., Wolczyk, M., Osiński, B., and Ondruska, P. (2022). Urban driver: Learning to drive from real-world demonstrations using policy gradients. In *Conference on Robot Learning*, pages 718–728. PMLR.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sharma, A. and Sharma, S. (2021). Wad: A deep reinforcement learning agent for urban autonomous driving. *arXiv preprint arXiv:2108.12134*.

Suo, S., Regalado, S., Casas, S., and Urtasun, R. (2021). Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10400–10409.

Treiber, M., Hennecke, A., and Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805.

Wang, P., Liu, D., Chen, J., Li, H., and Chan, C.-Y. (2021). Decision making for autonomous driving via augmented adversarial inverse reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1036–1042. IEEE.

Zhan, W., Sun, L., Wang, D., Shi, H., Clausse, A., Naumann, M., Kummerle, J., Konigshof, H., Stiller, C., de La Fortelle, A., et al. (2019). Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*.