2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

SHAIL: Safety-Aware Hierarchical Adversarial Imitation Learning for Autonomous Driving in Urban Environments[†]

Arec Jamgochian[*1], Etienne Buehrle[*2], Johannes Fischer[2], and Mykel J. Kochenderfer[1]

***Abstract*— Designing a safe and human-like decision-making system for an autonomous vehicle is a challenging task. Generative imitation learning is one possible approach for automating policy-building by leveraging both real-world and simulated decisions. Previous work that applies generative imitation learning to autonomous driving policies focuses on learning a low-level controller for simple settings. However, to scale to complex settings, many autonomous driving systems combine fixed, safe, optimization-based low-level controllers with high-level decision-making logic that selects the appropriate task and associated controller. In this paper, we attempt to bridge this gap in complexity by employing Safety-Aware Hierarchical Adversarial Imitation Learning (SHAIL), a method for learning a high-level policy that selects from a set of low-level controller instances in a way that imitates low-level driving data on-policy. We introduce an urban roundabout simulator that controls non-ego vehicles using real data from the Interaction dataset. We then demonstrate empirically that even with simple controller options, our approach can produce better behavior than previous approaches in driver imitation that have difficulty scaling to complex environments. Our implementation is available at `https://github.com/sisl/InteractionImitation`.**

I. Introduction

The development of autonomous vehicles will greatly impact urban traffic. Of particular importance is the safety and predictability of autonomous vehicles when interacting with complex environments. Achieving safe and human-like behavior will require a) multiple levels of safety redundancy, b) large amounts of real, "expert" driving data, and c) advanced simulators to test behavior before deploying.

Recent reinforcement learning approaches add levels of redundancy to policies learned in simulation by allowing for a hierarchy of control that passes between high-level action selectors and safe low-level optimization-based driving controllers [1], [2]. Though the addition of hierarchical safety layers is intuitive and adds levels of redundancy, the success of any reinforcement learning-based approach hinges on the design of the reward function. A misspecified reward function can be catastrophic.

Fig. 1: With SHAIL, the ego vehicle learns to choose from a set of safe high-level options to navigate a complex driving environment derived from the Interaction dataset [7]. The learner requires only low-level expert states and actions as opposed to high-level actions or a reward function.

To resolve the issue of reward misspecification, imitation learning approaches instead rely on demonstrations from an expert in the environment. Data availability invites the use of imitation learning methods that do not interact with the environment (i.e. off-policy methods, such as behavior cloning) [3]. However, these methods suffer from cascading errors when vehicles encounter out-of-distribution states [4]. Some on-policy approaches will query an expert to help guide the learning process safely [4], but querying an expert can be costly or impractical. Adversarial imitation learning approaches have been applied with simulators on-policy to circumvent the need for a queryable driving expert [5], [6], but these approaches have mostly been tested in simple driving settings and are still not collision-free.

We approach the safety and environment simplicity limitations of these prior applications of adversarial imitation learning to autonomous driving by taking a hierarchical approach. We note that many autonomous driving systems combine fast, safe, optimization-based controllers for low-level control with high-level logic to select appropriate tasks, controllers, and controller parameters. High-level logic might choose between different options (e.g. `LaneChangeLeft`, `Accelerate`, `TurnRight`, `EasyBrake`, `HardBrake`), then pass control to an instance of a low-level controller with the appropriate task and parameters for the chosen option. However, labels for these high-level choices are typically inaccessible in expert trajectories, making direct learning difficult.

[*]Denotes equal contribution

[1]Stanford University, Stanford, CA 94305 USA {arec, mykel}@stanford.edu

[2]Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany {etienne.buehrle, johannes.fischer}@kit.edu

[†]This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1656518. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This work is also supported by the COMET K2—Competence Centers for Excellent Technologies Programme of the Federal Ministry for Transport, Innovation and Technology (bmvit), the Federal Ministry for Digital, Business and Enterprise (bmdw), the Austrian Research Promotion Agency (FFG), the Province of Styria, and the Styrian Business Promotion Agency (SFG).

A large body of work exists around hierarchical imitation learning formulations for different robotics problems [8]–[13]. In this paper, we employ a method for learning a high-level controller-selection policy that imitates low-level driving data on-policy given a set of *known* low-level controller instances, as is appropriate for autonomous driving. A depiction of our problem setting is shown in Figure 1. We introduce Safety-Aware Hierarchical Adversarial Imitation Learning (SHAIL), which maintains the same low-level occupancy measure-matching objective of previous adversarial imitation learning approaches applied to driving [5], [14], but assumes that low-level data is generated within the options framework [15] and reformulates the objective accordingly. Additionally, SHAIL implements a safety awareness layer to adjust the high-level controller selection policy based on active reasoning about the safety or feasibility of different options.

To demonstrate the effectiveness of SHAIL, we develop a simulator based on real driving data from complex urban driving scenarios in the Interaction dataset [7]. Our simulator allows us to adjust the acceleration of an ego vehicle along its real path while other agents in a scene behave according to data. We test a basic implementation of SHAIL in roundabouts, a dynamic driving scenario that is typically difficult for an autonomous vehicle to safely navigate. We compare SHAIL against an IDM adaptation, behavior cloning (BC), non-hierarchical generative adversarial imitation learning (GAIL), and an ablation of SHAIL without the safety layer (HAIL), observing that SHAIL indeed yields safer and more realistic driving behavior.

In summary, the main contributions of this paper are to:

- Introduce SHAIL, a methodology for learning a safe high-level action-selection policy that imitates low-level observations and actions,
- Introduce a simulator for complex driving scenarios based on real data, and,
- Empirically demonstrate the efficacy of SHAIL compared to IDM, behavior cloning, and non-hierarchical imitation learning.

The remainder of this paper is organized as follows: Section II outlines background and related work, Section III describes the methodology for hierarchical adversarial imitation learning with safety constraints, Section IV presents experiments and results, and Section V concludes our research.

II. Background

This section provides necessary background on reinforcement learning, imitation learning, and hierarchical planning.

A. Reinforcement and Imitation Learning

Optimal decision-making is often framed in the context of Markov Decision Processes (MDPs). An MDP can be defined as a tuple $\langle \mathcal{S}, \mathcal{A}, T, R, b_0, \gamma \rangle$ and includes a finite state space $\mathcal{S}$, the action space $\mathcal{A}$, a stochastic transition function $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, a reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, an initial state distribution $b_0 : \mathcal{S} \to [0, 1]$, and a positive discount factor $\gamma < 1$. An MDP policy maps states to a distribution over actions to take $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$. An optimal policy maximizes expected cumulative discounted reward, $\pi^* \in \arg\max_{\pi \in \Pi} \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 \sim b_0(\cdot)]$.

In the reinforcement learning setting, the exact transition and reward functions T and R are unknown, but we can interact with an environment to receive generated samples of next state and reward $s', r \sim G(s, a)$. There is a body of work in which reinforcement learning is used to generate policies for different autonomous driving scenarios [16]–[18]. These works requires the use of a driving simulator to produce realistic transitions, as well as the manual specification of a reward function. Designing a reward function to capture all desired behavior is extremely difficult, and it is common for learning agents to exploit misspecified reward functions [19].

In the imitation learning setting, instead of receiving a reward signal, we rely on data in the form of trajectory rollouts from an expert who interacts with the environment. The imitation learning problem can be viewed as a problem of moment-matching between the expert and learner distributions, and methods can broadly be characterized as seeking to match Q-value moments off-policy, Q-value moments on-policy, or reward moments on-policy [20]. In off-policy Q-value moment matching, the learned imitating policy cannot interact with the environment until execution time [21]. The most straightforward approach to learn a policy in this setting is through behavior cloning (BC), in which a supervised learner regresses states to actions. This approach has a long history in autonomous driving systems [3], [22].

Behavior cloning suffers from an accumulation of errors during testing as an agent ends up in states it has not seen during training, a phenomenon often referred to as covariate shift [4]. In the on-policy Q-value-matching setting, this accumulation of errors is reduced by introducing a query-able expert who can correct deviating trajectories during training [4]. However, training with a query-able expert can be costly, timely, and impractical. In contrast, on-policy reward moment-matching algorithms assume no access to a query-able expert, but still assume interactions with the environment during training (e.g. using a simulator).

The state-action occupancy measure under a policy π is the (unnormalized) γ-discounted stationary distribution of states and actions visited under that policy, $\rho^\pi(s, a) = \pi(a \mid s)\rho^\pi(s)$, where $\rho^\pi(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s \mid \pi)$. We can similarly define the state-action occupancy measure of the expert policy, $\rho^{\text{exp}}(s, a)$. One perspective formulates imitation learning as moment-matching between expert and learned occupancy measures, done by minimizing some f-divergence between the associated distributions $\min_\pi D_f((1-\gamma)\rho^{\text{exp}}(\cdot, \cdot) \| (1-\gamma)\rho^\pi(\cdot, \cdot))$ [14]. In the on-policy reward matching setting, this objective can be written as a two-player game between a policy generator π_θ and an

observation-action discriminator D_ϕ:

$$\min_{\pi_\theta} \max_{D_\phi} \mathbb{E}_{(s,a)\sim\rho^{\exp}(\cdot,\cdot)}[D_\phi(s,a)] - \mathbb{E}_{(s,a)\sim\rho^{\pi_\theta}(\cdot,\cdot)}[f^*(D_\phi(s,a))], \quad (1)$$

where f^* denotes the convex conjugate of f. This objective can be optimized by alternating between discriminator gradient ascent steps to optimize the discriminator parameters ϕ and policy gradient ascent steps to optimize the parameters θ of a stochastic policy. This latter step can be viewed as reinforcement learning with an 'imagined' reward signal of $r(s,a) = f^*(D_\phi(s,a))$. These steps use Monte Carlo methods (and a replay buffer) to estimate the expectations [23], [24].

These generative methods have been used to imitate highway driving behavior [5]. Later work improves upon this by augmenting the learned reward model with soft constraints to avoid bad states and actions [6]. Two shortcomings of these works are that they a) mostly consider highway driving, a relatively simple driving scenario, and b) only learn low-level controllers, for which safety is more difficult to guarantee in comparison to traditional optimization-based controllers.

B. Hierarchical Planning

Human driving in complex environments is naturally hierarchical. One can model hierarchical planning through the context of options [15], in which a low-level controller o is chosen from a finite set of options $\mathcal{O}$ and executed until termination, upon which a new valid option is chosen. An options model can be defined with the tuple $\langle \mathcal{S}, \mathcal{A}, \{\mathcal{I}_o, \pi_o^L, \beta_o\}_{o\in\mathcal{O}}, T, R, b_o, \gamma\rangle$. In addition to the components of an MDP, an options model defines K options (indexed by o) which each define a set of states from which they can be initialized $\mathcal{I} \subseteq \mathcal{S}$, a low-level control policy $\pi^L : \mathcal{S} \times \mathcal{A} \to [0,1]$, and the probability that the option will be terminated from any given next state $\beta : \mathcal{S} \to [0,1]$. A high-level policy over options denotes the probability of choosing from the valid set of options in any given state $\pi^H(o \mid s)$, where $s \in \mathcal{S}$ and $o \in \mathcal{O}$ such that $s \in \mathcal{I}_o$.

Recent work considers hierarchical reinforcement learning for planning in driving scenarios [1], [2], [25]. While still suffering from the pitfalls of a manually specified reward function, these approaches have the benefit that a high-level action-selector can hand over control to safe, low-level, optimization-based planners. Mirchevska *et al.* use this approach to learn a high-level controller that can choose safe gaps in highway traffic for an optimization-based low-level controller to navigate to [1]. In their approach, the high-level controller only targets reachable gaps, while if a targeted gap no longer is reachable during low-level execution, control is passed back to the high-level controller.

Additional work considers hierarchical approaches to imitation learning. For example, Henderson *et al.* perform imitation learning hierarchically (as opposed to hierarchical imitation learning) by learning multiple generators and discriminators to match low-level data, and learning a mixture-of-experts policy over those generators to follow [26]. Le *et al.* describes a formulation to perform hierarchically-guided behavior cloning and dataset aggregation, however, this assumes labeling of the high-level option, which we do not [9].

Jing *et al.* perform hierarchical on-policy reward moment-matching by framing an objective to match the moments over states, actions, and options and alternating between expert option label inference and joint policy training. In this work, we keep low-level options fixed, as is more appropriate for driving. As a result, our work is in line with moment-matching objectives over state and action [14] as opposed to those additionally over options [13]. This can be viewed as a subclass of Jing *et al.* [13] where there is no need to infer latent options in the data, or as an extension of Ghasemipour *et al.* [14] in which the state-action pairs are drawn hierarchically.

Much of recent work that performs vehicle behavior prediction hierarchically (e.g. [27]) can be easily extended to off-policy hierarchical imitation learning, as many policies learned to predict vehicle behavior when conditioned on a goal could be extended to control an ego vehicle. A flavor of on-policy hierarchical imitation learning has been applied to driving policies, in which long-horizon planning learned to mimic a query-able expert is interleaved with fast, short-horizon, low-level optimal control [28]. In contrast, we propose learning a high-level controller on-policy that imitates low-level data without a query-able expert, and characterize a broader class of high-level control options.

III. Methodologies

This section formulates SHAIL, first by reformulating the occupancy measure-matching objective for a policy that generates low-level data hierarchically, and then by designing a safety-aware high-level controller.

A. Hierarchical Adversarial Imitation Learning

We first formulate the occupancy measure-matching objective in Equation (1) for a policy that is generating states and actions hierarchically. We do this by expanding the occupancy measure over options that would lead to state s and action a during their execution, and states in which the options begin executing. We expand over the initiation states $s_\tau = h$ that begin executing the options o at time τ under which low-level states and actions s and a can be observed at time t:

$$\rho^\pi(s,a) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s, a_t = a) \quad (2)$$

$$= \sum_{h,o} \sum_{t=0}^{\infty} \sum_{\tau=0}^{t} \gamma^\tau P(s_\tau = h, o_\tau = o) \cdot \gamma^{t-\tau} P(s_t = s, a_t = a \mid s_\tau = h, o_\tau = o) \quad (3)$$

$$= \sum_{h,o} \sum_{\tau=0}^{\infty} \gamma^\tau P(s_\tau = h, o_\tau = o) \cdot \sum_{t=\tau}^{\infty} \gamma^{t-\tau} P(s_t = s, a_t = a \mid s_\tau = h, o_\tau = o) \quad (4)$$

$$= \sum_{h,o} \sum_{\tau=0}^{\infty} \gamma^\tau P(s_\tau = h, o_\tau = o)$$
$$\cdot \sum_{t=0}^{\infty} \gamma^t P(s_t = s, a_t = a \mid s_0 = h, o_0 = o) \quad (5)$$
$$= \sum_{h,o} \rho^{\pi^H}(h,o)\rho^{\pi^L}(s,a \mid h,o), \quad (6)$$

where $\rho^{\pi^H}(h,o) = \pi^H(o \mid h)\sum_{\tau=0}^{\infty}\gamma^\tau P(s_\tau = h)$ is the discounted occupancy measure for ending up in a high-level state h and initiating option o, and $\rho^{\pi^L}(s,a \mid h,o) = \pi_o^L(a \mid s)\sum_{t=0}^{\infty}\gamma^t P(s_t = s \mid s_0 = h, o_0 = o)$ is the discounted occupancy measure of states and actions under an option o which was initialized in state h at time 0.

We apply this hierarchical representation of occupancy measure $\rho^\pi(s,a)$ to reformulate the measure-matching objective in Equation (1) for policy data that is generated hierarchically:

$$\min_{\pi} \max_{D_\phi} \mathbb{E}_{(s,a)\sim\rho^{\text{exp}}(\cdot,\cdot)}[D_\phi(s,a)] - \sum_{s,a}\rho^\pi(s,a)r(s,a) \quad (7)$$

$$\min_{\pi_\theta^H} \max_{D_\phi} \mathbb{E}_{(s,a)\sim\rho^{\text{exp}}(\cdot,\cdot)}[D_\phi(s,a)] - \sum_{h,o}\rho^{\pi_\theta^H}(h,o)\sum_{s,a}\rho^{\pi^L}(s,a \mid h,o)r(s,a) \quad (8)$$

$$\min_{\pi_\theta^H} \max_{D_\phi} \mathbb{E}_{(s,a)\sim\rho^{\text{exp}}(\cdot,\cdot)}[D_\phi(s,a)] - \mathbb{E}_{(h,o)\sim\rho^{\pi_\theta^H}(\cdot,\cdot)}[\tilde{r}(h,o)], \quad (9)$$

where $\tilde{r}(h,o) = \mathbb{E}_{(s,a)\sim\rho^{\pi^L}(\cdot,\cdot|h,o)}[f^*(D_\phi(s,a))]$.

Optimizing this objective can be done in a fashion similar to optimizing the objective in Eq. (1). Discriminator updates remain identical, while generator updates require performing policy gradients on $\pi_\theta^H(o \mid h)$ where the new 'imagined' high-level reward $\tilde{r}(h,o)$ accumulates the discounted low-level 'imagined' discrimination rewards from the execution of the chosen option. That is, $\tilde{r}(h,o)$ can be estimated as $\sum_{t=0}^{T_o}\gamma^t r(s_t,a_t)$, where $s_0 = h, a_t \sim \pi_o^L(\cdot|s_t)$, and T_o is the duration of the option.

B. Safety-Aware Hierarchical Adversarial Imitation Learning

Many practical implementations of policy gradients rely on a fixed-size action space [29], [30]. Because of this restriction, we are limited to an option set where any option can be initialized from every state, i.e. $\mathcal{I}_o = \mathcal{S}$ for all $o \in \mathcal{O}$. This assumption can be very limiting in terms of safety. Often times, we have information about restricted options from different states (e.g. an `Accelerate` option should not be taken from a red light). Additionally, we might be able to make predictions about the safety of different controllers. For example, this can be done strictly with formulations of reachability of a controller, or more loosely through notions of scene understanding (e.g. 'it is probably unsafe to make a turn since there are vehicles crossing the intersection'). SHAIL improves upon the formulation of hierarchical adversarial imitation learning presented in the previous section by designing a high-level option-selection policy that incorporates sensitivity to option safety.

Safety awareness is incorporated by assuming that the agent can reason about the safety or availability of different options from different states. We introduce a binary random variable z which predicts safety or availability of a low-level controller, denoting the probability that an option o is safe when executed from a high-level state s as $p_{\text{safe}}(z^1 \mid s,o)$. This allows us to design the options such that control is passed back to the high-level option selector according to this safety prediction, i.e. $\beta_o(s) = 1 - p_{\text{safe}}(z^1 \mid s,o)$. This option termination formulation expands on the one used by Mirchevska *et al.* in the hierarchical reinforcement learning setting [1] to admit probabilistic controller safety predictions rather than binary controller availability.

With this formulation, we additionally design a high-level controller that conditions on controller safety:

$$\pi_\theta^H(o \mid s, z^1) \propto p(o, z^1 \mid s) = p_{\text{safe}}(z^1 \mid s,o)\psi_\theta^H(o \mid s), \quad (10)$$

where ψ_θ^H is a learnable controller selector. This high-level controller reweights options based on predictions of their safety or availability. It can be easily shown that learning with this substituted policy is equivalent to minimizing the divergence to an agent occupancy measure conditioned on safety, $\rho^\pi(s,a \mid z^1)$. This scheme requires at least one option with nonzero safety probability (e.g. a permanent 'safe' controller), otherwise the high-level policy will not represent a valid distribution over controllers. Additionally, to learn a useful option selector, the options should have some semantic meaning that holds across different initialization states.

Learning ψ_θ^H with policy gradients on this policy formulation requires storing the safety probabilities seen during option initiation in the replay buffer. That is, for each option o initiated from a state h, the replay buffer consists of samples of the form $(h, o, p_{\text{safe}}(z^1 \mid h,\cdot), \tilde{r}(h,o))$.

IV. Experiments

Our experiments demonstrate the effective use of SHAIL in a driving simulator. We introduce our own simulator based on real data in urban driving environments, and demonstrate the improvements regarding safety that can be achieved in comparison with baseline models, even by a simple SHAIL implementation.

A. Setup

1) Simulator: To test this approach in a more complicated driving environment, we introduce the Interaction simulator.[1] The Interaction simulator is an OpenAI Gym [31] simulator that uses underlying data from the Interaction dataset of complex urban driving scenes [7]. The dataset consists of recorded track files from driving scenarios in different urban driving situations like roundabouts or intersections.

The simulator itself fixes vehicle paths and spawn times based on the recorded data in the Interaction dataset, and admits control of vehicle accelerations along the path. This

[1] `https://github.com/sisl/InteractionSimulator`

is a reasonable navigation strategy, as it may be common for a separate module to determine the path of a vehicle navigating a complex scene.

In our experiments[2], we focus on controlling a single vehicle that is modeled with double integrator dynamics and moves along its recorded path while non-ego vehicles follow their recorded trajectories. Simulations are terminated when the vehicle leaves the scene.

2) Features: We assume that the ego vehicle encodes its absolute velocity, yaw rate, and lidar-like measurements of the relative position and velocity of the closest vehicle in each 72° sector of its surroundings. We use this very simple subset of autonomous vehicle features in order to avoid overfitting to our small dataset.

3) Models: We evaluate the following baseline models in our experiments:

Expert: The expert model uses the default accelerations from the Interaction dataset.

IDM: The Intelligent Driver Model (IDM) models vehicle accelerations in fixed-lane driving when following a vehicle [32]. This model is problematic for uncontrolled driving, where it is not clear which vehicle the ego should 'follow'. We choose the follow vehicle as the closest vehicle that lies within two meters of the ego's planned path and has less than a 30° difference in heading. We target a desired speed of 8.94 m/s (20 mph), a minimum spacing of 3 m, a desired time headway of 0.5 s, a nominal acceleration of 3 m/s^2, and a comfortable braking deceleration of 2.5 m/s^2.

BC: We implement a behavior cloning agent that directly regresses features to a mean and standard deviation parameterizing a normal distribution for ego vehicle acceleration. Our model is a feedforward neural network with layers of fixed hidden size. We train our model by minimizing the negative log-likelihood of expert actions under the action distribution predicted from their preceding states.

GAIL: We compare against a model learned through Generative Adversarial Imitation Learning (GAIL) [5], [23]. Both discriminator and policy models are feedforward neural networks, the latter again outputting parameters for a normal distribution over next action. We use the same optimization objective as Ho and Ermon, as well as proximal policy optimization (PPO) [29] to learn a policy. We do not compare recurrent policies, as our episodes are too short to compare against recurrent hierarchical policies.

SHAIL: To demonstrate the effectiveness of SHAIL, we formulate a *very simple* hierarchical policy. Our high level controller chooses from a set of options which target a particular velocity at a particular future time, $\mathcal{O} = \{(v, t) \mid v \in \mathcal{V},\ t \in \mathcal{T}\}$, where $\mathcal{V}$ and $\mathcal{T}$ are discrete velocity and time sets. The low-level controller for each option commands a fixed acceleration to bring the vehicle to the desired velocity at the desired time. The safety predictor returns a binary indicator for whether the option is scheduled to collide with other vehicles if they maintain their velocity. Additionally, we overwrite the largest deceleration option to always be valid, thereby rendering it a default 'safe' option `HardBrake`. Again, we use the objective from Ho and Ermon, and PPO for policy gradients. We also learn a version of SHAIL without the safety layer or early option termination for ablation (**HAIL**).

4) Training and Metrics: Our experiments focus on model performance in roundabouts, a customarily tricky scenario for an autonomous system to navigate. Specifically, we look at the `DR_USA_Roundabout_FT` scene, which includes five recordings consisting of over 750 ego vehicle tracks. We believe it to be the most difficult roundabout scene in the dataset. Upon collision of a controlled vehicle, the environment is reset with a new random vehicle.

We perform two experiments. In our first experiment (*in-distribution*), we train and test models in the same environment, which selects vehicles only from the first track file. The purpose of this experiment is to compare absolute potential model performance. This in-distribution testing corresponds with what was done by Kuefler *et al.* [5].

In our second experiment (*out-of-distribution*), we train and validate in an environment that randomly selects vehicles from scene recordings 1–4, and we report metrics on scene 5. This out-of-distribution testing evaluates how the models perform on unseen vehicle data, though we acknowledge that we are still operating in the same driving setting. In both experiments, hyperparameters (e.g. model architecture, options sets, etc.) are optimized by choosing the ones which yield the highest success rate in the training environment. Please refer to our code for all parameters.

To avoid collisions caused by non-ego vehicles following their recorded trajectories, our test environment overrides non-ego accelerations with the IDM policy in case of an impending not-at-fault collision. For each model, we report success rate (the rate at which an episode does not terminate in collision), the average travelled distance (m), the root mean square error in position to the expert measured at 10 seconds into the trajectory (m), the average absolute difference between average speed of each vehicle under expert and modelled control (m/s), and the Jensen-Shannon divergence between the distributions over all accelerations. The divergence is estimated by fitting a histogram distribution to the two sets of samples.

B. Results and Discussion

Our simulation environment is visualized in Figure 2, while results from multiple runs of both in- and out-of-distribution roundabout experiments are shown in Table I. From the success rate and average distance traveled metrics, we can see that the simple IDM rule-following policy, behavior cloning, and GAIL all have trouble successfully navigating through the roundabout. From the in-distribution experiment, we see that incorporating safe options, even those as simple as the ones we suggest, can yield better performance when driving in complex environments. We see improvements in the success rate and travel distance metrics. We see that metrics that judge similarity to expert

[2] `https://github.com/sisl/InteractionImitation`

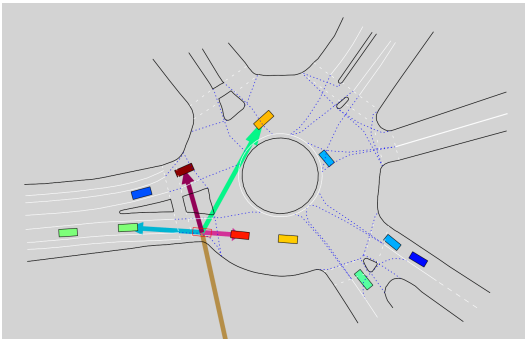

Fig. 2: A single time-step of a policy learned by SHAIL interacting with the environment. The ego vehicle has available to it its own motion state and lidar-like measurements capturing relative state information about up to five surrounding vehicles.

TABLE I: A comparison of performance between expert, IDM, BC, GAIL, HAIL (ablation), and SHAIL policies in both in-distribution (above) and out-of-distribution (below) roundabout experiments. We report metric means and two standard deviations after training each model five times.

Model	Success %	Travel Dist.	RMSE_{10s}	$\lvert\Delta V_{\text{avg}}\rvert$	Accel. JSD
Expert	100	82.1	—	—	—
IDM	66.2	67.5	19.2	1.52	0.050
BC	45.3±3.0	49.8±1.4	22.0±1.9	1.88±0.11	0.275±0.017
GAIL	68.3±4.8	65.3±3.9	14.9±1.1	**1.08**±0.14	**0.016**±0.021
HAIL (ab)	53.0±24.1	54.5±15.3	18.3±4.8	1.87±0.86	0.333±0.008
SHAIL	**77.7**±3.1	**70.6**±2.3	**14.7**±2.1	1.27±0.23	0.312±0.012
Expert	100	81.9	—	—	—
IDM	56.3	59.9	21.0	1.45	0.061
BC	40.4±3.1	48.8±1.3	22.5±0.5	1.92±0.05	0.290±0.018
GAIL	51.6±6.8	54.5±4.7	**14.7**±1.6	1.41±0.21	**0.031**±0.018
HAIL (ab)	39.9±8.6	46.3±6.5	21.1±5.0	2.10±0.55	0.328±0.016
SHAIL	**64.4**±6.8	**61.6**±3.6	18.1±1.8	**1.37**±0.34	0.300±0.027

position and speed perform comparably well in both GAIL and SHAIL, indicating some human-like behavior in both.

We note that the distribution over SHAIL accelerations is quite far from the expert distribution, especially when compared to GAIL. This disparity can be attributed to our overly simple option design. Our controllers attempt to target different velocities at different points by holding fixed accelerations, ultimately resulting in jerky behavior. We could bridge this gap by implementing more comfortable, human-like controller options.

In our ablation study, implementing the same options without any safety layering (HAIL) results in a severe drop in performance. Intuitively, when implementing options without safety or termination criteria, we are reducing the space of immediate actions available to the agent. Even if we could learn a good imitating controller that predicts which options might be safe from a particular state, we have no method for terminating if the options become unsafe. In our experiments, this is made even worse by our simple controllers, which stick to the action plan that was initiated during option selection and do not adjust their plan based on environment feedback.

We see similar results in our out-of-distribution experiment, noting that the performance gap between SHAIL and other models is even greater. Though none of the learning methods perform as well as they would in-distribution, this performance gap suggests that SHAIL could be a good approach for navigating new situations. We note though that our out-of-distribution experiment tested in the same roundabout setting as training, just with new vehicle data. Though we believe this to be the hardest setting, it would be interesting to train our approach over different settings and test on a fully unseen one to see how well the learned safe option selector could generalize.

V. Conclusion

Previous work applying adversarial imitation learning to autonomous driving focuses on learning low-level control policies. However, since many autonomous driving systems rely on optimization-based control to provide safe low-level policies, it may be more prudent to rely on data-driven tools to learn high-level control policies. In this work, we introduced Safety-Aware Hierarchical Adversarial Imitation Learning (SHAIL), a methodology for learning high-level control policies in a simulator such that low-level expert trajectories are imitated. SHAIL incorporates an additional layer to reason over option safety and availability, allowing to guarantee that only safe and feasible actions are executed. To demonstrate our approach, we developed a simulator based on the Interaction dataset of real complex urban driving scenarios. Finally, we compared SHAIL to previous approaches to empirically demonstrate the safety improvements that it affords when navigating a roundabout.

SHAIL inherits all of the limitations from generative adversarial networks, policy optimization, and simulation-based approaches to policy learning. Additionally, SHAIL limits high-level options to a fixed set of predetermined low-level control policy instances. These options must hold semantic meaning in different initialization states for meaningful learning to occur. One limitation of our experiments is the simplicity of the low-level controllers used, which are meant only to demonstrate the potential of our proposed method. More advanced low-level controllers, safety predictors, and termination criteria can easily be substituted into our method. SHAIL can be extended to perform reward augmentation to design policies that avoid known unfavorable behavior. It can also potentially be applied across settings to learn more generalizable option-selection.

Acknowledgment

The authors acknowledge Kunal Menda for early work on the Interaction simulator.

References

[1] B. Mirchevska, M. Hügle, G. Kalweit, M. Werling, and J. Boedecker, "Amortized Q-learning with model-based action proposals for autonomous driving on highways," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1028–1035.

[2] D. Kamran, Y. Ren, and M. Lauer, "High-level decisions from a safe maneuver catalog with reinforcement learning for safe and cooperative automated merging," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2021, pp. 804–811.

[3] D. A. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems (NeurIPS)*, D. Touretzky, Ed., 1989.

[4] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011, pp. 627–635.

[5] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 204–211.

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

[7] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kümmerle, H. Königshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps," *arXiv:1910.03088 [cs, eess]*, Sep. 2019.

[8] S. Krishnan, A. Garg, R. Liaw, L. Miller, F. T. Pokorny, and K. Goldberg, "HIRL: Hierarchical inverse reinforcement learning for long-horizon tasks with delayed rewards," 2016. arXiv: 1604.06508 [cs.RO].

[9] H. Le, N. Jiang, A. Agarwal, M. Dudik, Y. Yue, and H. Daumé III, "Hierarchical imitation and reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2018, pp. 2917–2926.

[10] M. Sharma, A. Sharma, N. Rhinehart, and K. M. Kitani, "Directed-Info GAIL: Learning hierarchical policies from unsegmented demonstrations using directed information," in *International Conference on Learning Representations*, 2018.

[11] P. Sharma, D. Pathak, and A. Gupta, "Third-person visual imitation learning via decoupled hierarchical controller," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[12] R. Fox, R. Berenstein, I. Stoica, and K. Goldberg, "Multi-task hierarchical imitation learning for home automation," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 1–8.

[13] M. Jing, W. Huang, F. Sun, X. Ma, T. Kong, C. Gan, and L. Li, "Adversarial option-aware hierarchical imitation learning," in *International Conference on Machine Learning (ICML)*, vol. 139, 2021, pp. 5097–5106.

[14] S. K. S. Ghasemipour, R. Zemel, and S. Gu, "A divergence minimization perspective on imitation learning methods," in *Conference on Robot Learning*, 2020, pp. 1259–1277.

[15] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[16] F. Belletti, D. Haziza, G. Gomes, and A. M. Bayen, "Expert level control of ramp metering based on multi-task deep reinforcement learning," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1198–1207.

[17] D. Kamran, C. Fernandez Lopez, M. Lauer, and C. Stiller, "Risk-aware high-level decisions for automated driving at occluded intersections with reinforcement learning," in *IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1205–1212.

[18] J. Chen, B. Yuan, and M. Tomizuka, "Model-free deep reinforcement learning for urban autonomous driving," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2019, pp. 2765–2771.

[19] D. Amodei, C. Olah, J. Steinhardt, P. F. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," *arXiv:1606.06565 [cs]*, 2016.

[20] G. Swamy, S. Choudhury, J. A. Bagnell, and S. Wu, "Of moments and matching: A game-theoretic framework for closing the imitation gap," in *International Conference on Machine Learning (ICML)*, 2021, pp. 10 022–10 032.

[21] I. Kostrikov, O. Nachum, and J. Tompson, "Imitation learning via off-policy distribution matching," in *International Conference on Learning Representations (ICLR)*, 2020.

[22] J. Hawke, R. Shen, C. Gurau, S. Sharma, D. Reda, N. Nikolov, P. Mazur, S. Micklethwaite, N. Griffiths, A. Shah, *et al.*, "Urban driving with conditional imitation learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 251–257.

[23] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4565–4573, 2016.

[24] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adverserial inverse reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2018.

[25] J. Chen, Z. Wang, and M. Tomizuka, "Deep hierarchical reinforcement learning for autonomous driving with distinct behaviors," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1239–1244.

[26] P. Henderson, W.-D. Chang, P.-L. Bacon, D. Meger, J. Pineau, and D. Precup, "OptionGAN: Learning joint reward-policy options using generative adversarial inverse reinforcement learning," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[27] L. Wang, Y. Hu, L. Sun, W. Zhan, M. Tomizuka, and C. Liu, "Transferable and adaptable driving behavior prediction," *arXiv:2202.05140 [cs]*, 2022.

[28] L. Sun, C. Peng, W. Zhan, and M. Tomizuka, "A fast integrated planning and control framework for autonomous driving via imitation learning," in *Dynamic Systems and Control Conference*, 2018.

[29] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International Conference on Machine Learning (ICML)*, 2015, pp. 1889–1897.

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347 [cs]*, 2017.

[31] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI gym," *arXiv:1606.01540 [cs]*, 2016.

[32] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, 2000.