Human-compatible driving partners through data-regularized self-play reinforcement learning

Daphne Cornelisse
New York University
cornelisse.daphne@nyu.edu

Eugene Vinitsky
New York University
eugenevinitsky@nyu.edu

Abstract

A central challenge for autonomous vehicles is coordinating with humans. Therefore, incorporating realistic human agents is essential for scalable training and evaluation of autonomous driving systems in simulation. Simulation agents are typically developed by imitating large-scale, high-quality datasets of human driving. However, pure imitation learning agents empirically have high collision rates when executed in a multi-agent closed-loop setting. To build agents that are realistic and effective in closed-loop settings, we propose Human-Regularized PPO (HR-PPO), a multi-agent algorithm where agents are trained through self-play with a small penalty for deviating from a human reference policy. In contrast to prior work, our approach is RL-first and only uses 30 minutes of imperfect human demonstrations. We evaluate agents in a large set of multi-agent traffic scenes. Results show our HR-PPO agents are highly effective in achieving goals, with a success rate of 93%, an off-road rate of 3.5 %, and a collision rate of 3 %. At the same time, the agents drive in a human-like manner, as measured by their similarity to existing human driving logs. We also find that HR-PPO agents show considerable improvements on proxy measures for coordination with human driving, particularly in highly interactive scenarios. We open-source our code and trained agents at https://github.com/Emerge-Lab/nocturne_lab and share demonstrations of agent behaviors at https://sites.google.com/view/driving-partners.

1 Introduction

Developing autonomous vehicles (AVs) that are compatible with human driving remains a challenging task, especially given the low margin for error in the real world. Driving simulators offer a cost-effective and safe means to develop and refine autonomous driving systems. The purpose of these simulators is to prepare AVs for real-world deployment, where they must smoothly interact and coordinate with a diverse set of human drivers. Therefore, a crucial aspect of both learning and validation in these simulators involves realistic simulations: the traffic scenarios and *other simulation agents with which the controlled AV interacts*. To identify where driving policies fall short, it is important to ensure that the simulated traffic conditions and driver agents closely resemble those in the real world Gulino et al. (2023); Muhammad et al. (2020).

Existing driving simulators typically provide a set of baseline agents to interact with, such as low-dimensional car following models, rule-based agents, or recorded human driving logs (Treiber et al., 2000; Gulino et al., 2023; Dosovitskiy et al., 2017). While these agents provide a form of interactivity, they are limited in their abilities to create interesting and challenging coordination scenarios, which requires driving agents that are reactive and sufficiently human-like. Having effective simulation agents that drive and respond in human-like ways would facilitate the controlled generation of human-AV interactions, which has the potential to unlock realistic training and evaluation in simulation at scale. Additionally, it would reduce the need for continuous real-world large-scale data collection.

Building human-like driving policies is an ongoing challenge. Existing simulated agents are either (1) quite far from human-like behavior (2) struggle with achieving closed-loop stability or (3) frequently get stuck in deadlocks. A ubiquitous way to generate driving policies has been through imitation learning, where a driving policy is learned by mimicking expert behavior using recorded actions from human drivers (Pomerleau, 1988; Xu et al., 2023). Unfortunately, such policies still have high crash rates when put in a multi-agent closed-loop setting where they have to respond to the actions of other agents (Montali et al., 2024). Another approach that has been explored to achieve closed-loop stability is multi-agent RL (Vinitsky et al., 2022). While in principle perfect closed-loop driving may be achieved via self-play, there is no guarantee that the equilibrium the agents find will be at all human-like. For example, self-play agents have no a priori reason to prefer driving on the left side of the road vs. the right. Similarly, because every agent is aware that other agents are a copy of themselves, they may feel comfortable driving much closer to each other than human comfort and reaction times would allow.

As a step towards effective and realistic driving partners for simulation, we propose **Human-Regularized PPO** (HR-PPO). HR-PPO is an on-policy algorithm that includes an additional regularization term that nudges agents to stay close to human-like driving. Concretely, our contributions are:

- We show that adding a regularization term to PPO agents trained in self-play leads to agents that are **more compatible with proxies for human behavior** in a variety of scenarios in `Nocturne`, a benchmark for multi-agent driving.
- Our results also show that **effectiveness** (being able to navigate to a goal without colliding) and **realism** (driving in a human-like way) **can be achieved simultaneously**: Our HR-PPO agents achieve similar performance to PPO while experiencing substantial gains in human-likeness.
- We also show the benefits of training in multi-agent settings: **HR-PPO self-play agents outperform agents trained directly on the test distribution of agents**. This suggests that multi-agent training may provide additional benefits over single-agent training (log-replay).

2 Methods and background

2.1 Human-Regularized PPO

Let o_t, a_t denote the observation and action at time step t and $r(o, a)$ the instantaneous reward for the agent that executes action a in state o. The history up to time T is defined as $x_t = (o_1, a_1, \ldots, a_{T-1}, o_T)$ (e.g. data collected from a rollout). The basic form of a KL-regularized expected reward objective is defined as:

$$\mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^t r(o_t, a_t) - \lambda \cdot D_{\mathrm{KL}}\Big(\tau(\cdot \mid o_t) \,\|\, \pi(\cdot \mid o_t)\Big)\right]$$

where π is the most recent **stochastic policy**, τ is a stochastic **behavioral reference policy** obtained from a dataset $\mathcal{D}$ and λ denotes the **regularization weight**. The KL divergence is defined as the expectation of the logarithmic differences between the pre-trained (fixed) human-policy and RL policy action probability distributions. For a single observation o and discrete actions, the KL Divergence between the action distributions is defined as:

$$D_{\mathrm{KL}}(\,\tau(\cdot \mid o) \,||\, \pi(\cdot \mid o)\,) = \sum_{a \in \mathcal{A}} \tau(a) \cdot \log\left(\frac{\tau(a)}{\pi(a)}\right)$$

where our action space $|\mathcal{A}| = 651$. We use the KL-divergence between τ and π as a regularization term added to the standard Proximal Policy Optimization (PPO) objective (Schulman et al., 2017)

to obtain **Human-Regularized PPO**:

$$\mathcal{L}_t^{\text{HR-PPO}}(\theta) = (1-\lambda) \cdot \mathcal{L}_t^{\text{PPO}}(\theta) + \lambda \cdot D_{\text{KL}}(\tau \| \pi)$$

where λ is a hyperparameter that determines the importance of both objectives. For details on the trained behavioral reference policy distributions, see Appendix C. For training and implementation details, see Appendix D. We implement our code based atop `Stable Baselines3` (Raffin et al., 2021).

Expert demonstrations We obtain a dataset of observation-action pairs $D^k = \{(\mathbf{o}_t^i, \mathbf{a}_t^i), \ldots, (\mathbf{o}_T^N, \mathbf{a}_T^N)\}_{i=1}^N$ for N vehicles and $T = 80$ time steps, for a set of K traffic scenarios in the Waymo Open Motion Dataset (WOMD) (Ettinger et al., 2021). The human driver ("expert") actions (acceleration, steering) are inferred from the positions and velocity of the observed positions using a dynamic bicycle model (Gulino et al., 2023). As the scenarios are recorded by fusing sensors onboard an autonomous vehicle (AV), the inferred positions of the AV are of higher quality compared to those of surrounding non-AV vehicles, which tend to have more noise. Therefore, we only use the demonstrations from the AV vehicles. To illustrate the difference between AV and non-AV demonstrations, Table 5 contrasts the performance under different conditions, and Figures 10, 11, and 12 show several randomly sampled trajectories in the dataset.

Imitation Learning We train a Behavioral Cloning (BC) policy on the shuffled dataset of observation-action pairs to an open-loop accuracy of 97-99%. The dataset, $\mathcal{D} = \{(o_i, a_i)\}_{i=1}^{(T \cdot K)}$ is obtained from $K = 200$ scenarios with $T = 90$ time steps, which is equal to just **30 minutes of driving data**. We obtain the behavioral reference policy τ using the negative log-likelihood objective to the expert demonstrations:

$$\tau_{\text{NLL}} = \arg\min_{\tau \in \mathcal{T}} \sum_{i=1}^{N} -\log \tau(a_i \mid o_i)$$

and implement the algorithm using the `imitation` package (Gleave et al., 2022). Table 1 compares the performance of BC policies trained and evaluated on randomly assigned vehicles to only AV vehicles. We also show the performance obtained with the discretized expert actions (top-row), which is an upper bound on performance with this action space. Our BC policy trained on only the AV demonstrations performs better when used to control either the AVs or the random (non-AV) vehicles in the scenarios. Therefore, we select this policy as a regularizer in the multi-agent human-regularized PPO setting.

Table 1: Imitation Learning (IL) performance.

Agent	Action Space	Generate data from	Evaluate on	Off-road Rate (%)	Collision Rate (%)	Goal Rate (%)
Expert-*actions*	21×31	AV only	AV only	9.2	3.3	78.0
BC	21×31	AV only	AV only	11.0	4.0	73.1
BC	21×31	AV only	Random vehicle	16.0	10.4	51.0
BC	21×31	Random vehicle	AV only	17.8	9.0	48.4
BC	21×31	Random vehicle	Random vehicle	17.2	7.6	46.2

2.2 Environment details

2.2.1 Dataset and simulator

We use `Nocturne` (Vinitsky et al., 2022), a 2D multi-agent driving simulator that runs at 2000+ FPS built on top of the Waymo Open Motion Dataset (WOMDB; (Ettinger et al., 2021)) for training and evaluation. For the training dataset, we partition 10,200 randomly chosen traffic scenarios into 200 for training and 10,000 for testing. Each traffic scenario is 9 seconds, which is discretized at 10 hertz. We use the first second as a warmup period that provides agents with context, so each episode has

80 steps. Details on the dataset, such as the number of vehicles per scene and the interactivity of the scenarios can be found in Appendix A.

2.2.2 Partially observable driving navigation tasks

At initialization, every vehicle in a scenario starts at a fixed position $\mathbf{x}_0^i = (x_0^i, y_0^i)$ and is assigned a fixed goal position $\mathbf{x}_g^i = (x_g^i, y_g^i)$. A vehicle obtains the sparse reward when its center is within a tolerance region of its goal position: $\|\mathbf{x}_t^i - \mathbf{x}_g\|_2 < \delta$ before the end of the episode, which is at most 80 steps. The goal positions are fixed and set to the last point from every logged vehicle trajectory. We set the tolerance region to $\delta = 2$ meters. Vehicles are removed from the scene when they go off-road or collide with another agent.

2.2.3 State space

A vehicle i has two main sources of information about the environment. The first is the **ego state**, $\mathbf{s}^i \in \mathbb{R}^{10}$, which includes the speed, the vehicle length, and width, its current speed, the distance to the goal position, the angle to the goal position (target azimuth), the heading and speed at goal position from the logged trajectory, the current acceleration and the current steering position. Secondly, the vehicle has a **partial view of the traffic scene** which is constructed by parameterizing the view distance, head angle, and cone radius of the driver $\mathbf{v}^i \in \mathbb{R}^{6720}$ and contains the road graph information, vehicle objects and the positions and speeds of the other vehicles that are within its field of view. Figure 1 shows an example scene in Nocturne with the obstructed vehicle view. We denote the full observation for a vehicle i as $\mathbf{o}^i = [\mathbf{s}^i, \mathbf{v}^i]$. The observations are all relative to every agent's own ego-centric frame. In this work the cone radius is always 180 degrees and the radius of the cone is 80 meters.

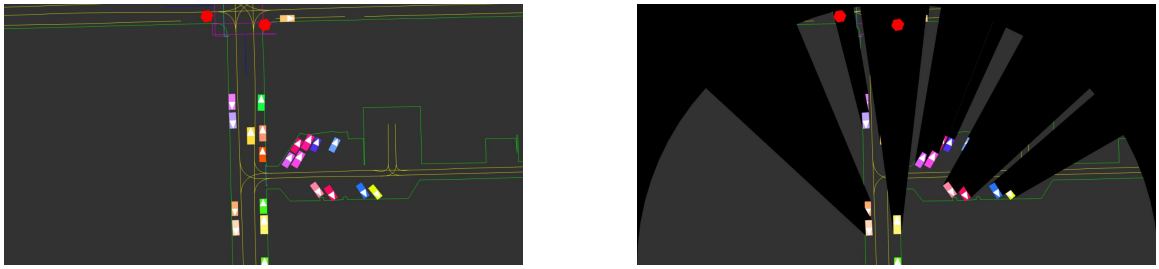

Figure 1: LHS: A bird's eye view of an example scenario in the training dataset from the perspective of the green agent in the bottom center. RHS: Agents only have a partial view of the environment and must plan under uncertainty.

2.2.4 Action space

At each time step, all agents simultaneously take actions. An action is a 2-dimensional tuple with the vehicle's acceleration and steering wheel angle. We create a joint action space by discretizing the actions (acceleration, and steering) into a grid of 21 x 31 = 651 actions. The steering wheel angle lower bound is set to -0.3 radians and the upper bound to 0.3 radians. The acceleration bounds are -4 and 4 m/s^2.

2.3 Reward function

In our agent-based simulation, we provide sparse rewards to agents when they reach their goal position before the end of the 80-step episode. If an agent reaches its goal, it receives a reward of +1. Otherwise, it receives a reward of 0. The goal-achieved condition is satisfied when the vehicle is within a tolerance region of 2 meters from the target position. If a vehicle collides with another vehicle, goes off the road, or achieves its goal, it is removed from the scene. The reward function

is intentionally simplified, omitting common additions such as reducing the distance to the goal, maintaining a safe distance from other vehicles, or following road rules. This is done so that all of these components can emerge from imitation regularization, rather than being hardcoded in.

3 Experiments and results

3.1 Baselines and implementation details

We use self-play to train HR-PPO agents in scenarios where we control all the vehicles in the scene, with a maximum of 43 controlled vehicles. Full implementation details, including the architecture, hyperparameters, and compute used, are found in Appendix D. We compare HR-PPO agents with four different baseline training methods:

- Multi-agent PPO: Self-play while controlling all vehicles in the scene, without regularization.
- Single-agent PPO: Sample a random agent at reset to control, step the rest of the agents in log-replay.
- Single-agent HR-PPO: Add regularization but all but one random agent is in log-replay.
- Behavioral Cloning: The behavioral reference policy.

3.2 Evaluation metrics

We evaluate our driving agents based on two classes of metrics, as shown in Figure 2. We refer to the first category as *Effectiveness*, which measures how well driving agents can achieve their goal safely, without colliding or going off-road. The second category, *Realism*, assesses how closely the driving behavior of the agents matches that of human drivers in the dataset. We use a variation of the Average Displacement Error (ADE) to measure the deviation from the logged human trajectories. In contrast to the trajectory prediction setting, our agents are goal-conditioned and thus they don't have to do inference over their own target goal positions. To distinguish this from the metric used in trajectory prediction, we refer to the metric as the **Goal Condtioned ADE (GC-ADE)**. Additionally, we examine the absolute differences between the human expert actions and the policy-predicted steering wheel angle and acceleration at each time step. Full details on the metrics are in Appendix E.

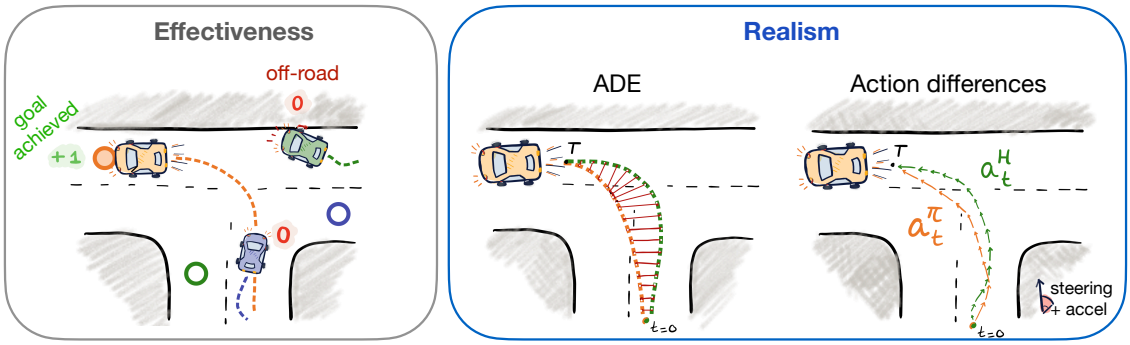

Figure 2: Overview of metrics used for evaluation. Left: Agents achieve their goal if they reach the target (color-coded circles) without collisions before the episode ends (80 steps). In this example, the goal rate is 1/3 (only the yellow car reaches its goal), the off-road rate is 1/3 (the green car hits a road edge) and the collision rate is 0 (no vehicle crashes with another vehicle). Right: Realism metrics concern *how* agents navigate to their goal positions, that is, the extent to which the policy-generated trajectories (orange) resemble the logged human ones (green).

3.3 Aggregate performance

Table 2 shows our aggregate performance. We compare the performance of HR-PPO agents to the baselines on the full *train* dataset, which consists of 200 traffic scenarios, and the *test* dataset,

consisting of $10,000$ unseen traffic scenarios. Scenarios have between 1 and 58 vehicles, with an average of 12. We consider two evaluation modes:

- *Self-replay* indicates the setting where we are using a trained policy to control all vehicles in the scenario.
- *Log-replay* indicates that we sample a single, random, vehicle in the scene to control, and the rest of the vehicles are stepped using the **static human replay logs**. To reduce randomness in the performance, we sample each scenario in the dataset 15 times, given that an average of 13 vehicles are included in each scenario. This is distinct from the definition of log-replay in other works (Gulino et al., 2023) where only the AV vehicle (the vehicle used to collect data) is controlled.

We highlight our main findings below.

Agents trained in self-play exhibit the highest performance across all modes: In closed-loop self-play, the HR-PPO and PPO agents trained in multi-agent mode using self-play achieve the highest performance overall: HR-PPO has a goal rate of 93.35 %, an off-road rate of 3.51%, and a collision rate of 2.98 %. PPO has a similar goal rate and off-road rate, with a slightly higher collision rate of 3.97 %. The standard errors across scenarios are small, typically between 0.5 and 1%. Further, we observe that training in a multi-agent self-play setting is more effective than training in single-agent settings across all test conditions. We find that self-play HR-PPO and PPO agents both outperform their single-agent variants by 10-14%. Surprisingly, even in log-replay evaluation mode, where the self-play agents encounter previously unseen human driving agents, the HR-PPO self-play agents still achieve a 3% improvement over agents *trained directly against the human driving logs*.

Agent-generalization gap decreases using HR-PPO: Agents trained in self-play typically overfit their training partner. To assess how well the agents can generalize to the unseen human drivers, we compare the change in performance when we switch from self-play to log-replay. Table 2 shows that HR-PPO agents have the highest log-replay performance overall and show an improvement of 11% in goal rate and a 14% improvement in collision rate to PPO. Separately, we notice that the train - test gap, which combines both agent generalization and scene generalization, is negligible for BC and small for both PPO and HR-PPO, especially given that we train on 200 and evaluate on 10,000 scenes. Overall the performance decreases by approximately 1-8%.

3.4 Driving in a human-like way

Human-like and effective driving agents. We aim to construct useful driving agents that can navigate effectively and resemble human driving behavior. To test whether these two properties can be achieved simultaneously, we contrast several existing realism metrics against the effectiveness of agents (Details of the metrics in Section 3.2). Across all four human similarity metrics, we observe that significantly more human-like behavior can be achieved for a minimal or even no trade-off in performance. For instance, Figure 3 shows that HR-PPO with a regularization weight of $\lambda = 0.06$ has a Goal-Conditioned Average Displacement Error (GC-ADE) of 0.54, which is a 60% improvement to PPO (GC-ADE is 1.32), for a decrease in goal rate of 1%, and increase in off-road rate of less than 1%. We observe the same pattern when we compare the policy-predicted actions to the logged human driving logs, as shown in Figures 4,20, and 19. These measures hold when evaluated in a single-agent setting where we control only the AV vehicles (shown in Table 4) as well as the setting where we control all vehicles in the scene (Table 3).

Natural correction for bad actions. Datasets of human driving may contain noise or undesirable actions. For instance, in our dataset, the off-road rate of replaying the expert actions is quite high ($>$ 12%). However, we observe that HR-PPO agents, which are trained with these imperfect behavioral cloning actions, learn to ignore a large fraction of them and instead achieve an off-road rate between 2-4%. This finding suggests that it may not be necessary to have a near-perfect BC policy as the regularizer as RL can compensate for some of the weaknesses of the regularization policy.

Table 2: HR-PPO performance compared to baselines. We report the aggregate mean performance and standard errors across scenarios. *Log-replay* indicates that the agent is evaluated in a single-agent setting where all the other agents are replaying static human driving logs. *Self-play* indicates that all agents in the environment are controlled. The performance means and deviations across *seeds* are shown in Figure 22 and Table 8 in the Appendix.

Agent	Train mode	Dataset	Eval mode	Goal Rate (%)	Off-road Rate (%)	Collision Rate (%)
BC	-	Test	Log-replay	43.95 ± 0.57	19.05 ± 0.51	14.40 ± 0.41
			Self-play	49.22 ± 0.12	15.45 ± 0.11	14.11 ± 0.09
		Train	Log-replay	51.65 ± 0.58	14.55 ± 0.44	12.00 ± 0.41
			Self-play	50.23 ± 0.59	13.13 ± 0.40	13.97 ± 0.40
HR-PPO	Single-agent	Test	Log-replay	72.65 ± 0.45	11.90 ± 0.34	11.35 ± 0.34
			Self-play	76.50 ± 0.09	9.44 ± 0.07	10.32 ± 0.07
		Train	Log-replay	80.15 ± 0.38	8.75 ± 0.29	7.70 ± 0.25
			Self-play	80.15 ± 0.32	6.18 ± 0.23	9.85 ± 0.22
	Multi-agent	Test	Log-replay	76.30 ± 0.45	9.25 ± 0.34	14.65 ± 0.34
			Self-play	86.73 ± 0.09	6.66 ± 0.07	6.40 ± 0.07
		Train	Log-replay	83.75 ± 0.38	5.55 ± 0.29	10.10 ± 0.25
			Self-play	93.35 ± 0.32	3.51 ± 0.23	2.98 ± 0.22
PPO	Single-agent	Test	Log-replay	71.70 ± 0.44	10.25 ± 0.32	19.50 ± 0.36
			Self-play	77.50 ± 0.09	9.99 ± 0.07	13.20 ± 0.08
		Train	Log-replay	81.10 ± 0.40	7.55 ± 0.27	12.55 ± 0.33
			Self-play	83.44 ± 0.38	6.49 ± 0.23	10.61 ± 0.31
	Multi-agent	Test	Log-replay	67.40 ± 0.44	7.00 ± 0.32	27.30 ± 0.36
			Self-play	85.70 ± 0.09	5.93 ± 0.07	8.94 ± 0.08
		Train	Log-replay	72.80 ± 0.40	4.30 ± 0.27	24.20 ± 0.33
			Self-play	93.44 ± 0.38	3.13 ± 0.23	3.97 ± 0.31

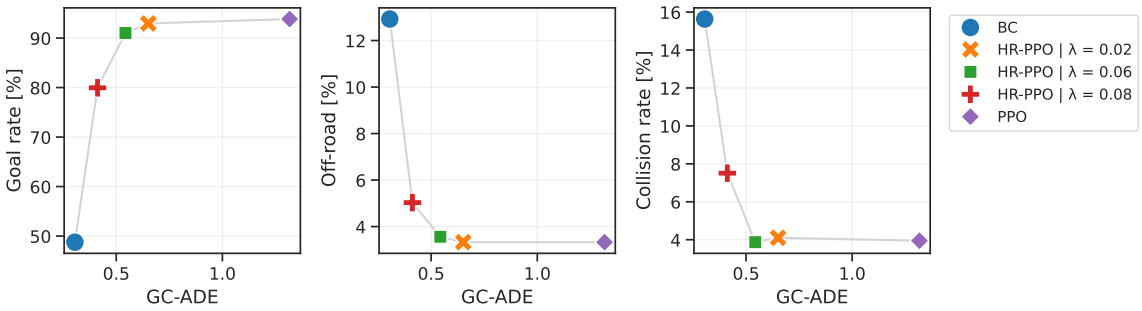

Figure 3: Goal-Conditioned Average Displacement Error (GC-ADE) to logged human driver positions against effectiveness metrics conditioned on knowing the goal. Policies are evaluated on the training dataset of 200 scenarios.

Table 3: Mean and standard error across the 200 scenarios in the training dataset, **controlling all vehicles in every scenario** (Self-play). The reported HR-PPO performance is with $\lambda = 0.06$ (green square in the Figures above).

Agent	GC-ADE	Accel MAE	Action Acc. (%)	Speed MAE	Steer MAE
BC	0.31 ± 0.01	1.71 ± 0.02	5.61 ± 0.02	0.84 ± 0.02	0.02 ± 0.00
HR-PPO	0.54 ± 0.01	2.09 ± 0.02	3.25 ± 0.01	1.82 ± 0.03	0.02 ± 0.00
PPO	1.32 ± 0.03	3.93 ± 0.02	0.20 ± 0.00	5.07 ± 0.08	0.08 ± 0.00

Table 4: Mean performance and standard errors across the training dataset of 200 scenarios, controlling **only the AV vehicle** in every scenario. This is distinct from the **log-replay** setting where a random vehicle is set as controlled. The reported HR-PPO performance is with $\lambda = 0.06$.

Agent	GC-ADE	Accel MAE	Action Acc. (%)	Speed MAE	Steer MAE	Goal Rate (%)	Off-Road Rate (%)	Collision Rate (%)
BC	0.08 ± 0.01	0.41 ± 0.02	0.22 ± 0.01	0.09 ± 0.01	0.01 ± 0.00	69.50 ± 1.68	11.00 ± 2.21	6.00 ± 1.68
HR-PPO	0.56 ± 0.03	1.15 ± 0.06	0.10 ± 0.01	1.83 ± 0.08	0.01 ± 0.00	90.00 ± 2.12	1.50 ± 0.86	8.50 ± 1.97
PPO	1.22 ± 0.06	3.92 ± 0.05	0.00 ± 0.00	4.77 ± 0.19	0.09 ± 0.00	71.50 ± 3.19	2.00 ± 0.99	28.00 ± 3.17

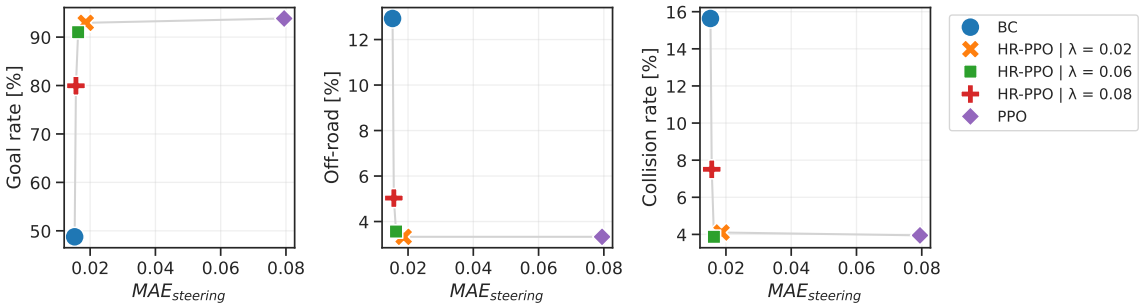

Figure 4: Steering MAE against effectiveness metrics.

3.5 Coordinating with human drivers

We explore the ability of HR-PPO agents to coordinate with human drivers in interactive scenarios. Since we cannot directly interact with human drivers, we use the available driving logs as a proxy instead. We compare the collision rates between self-play mode, where all agents are controlled by a single policy, and log-replay mode, where a single random agent is controlled by our policy, and the rest of the agents are controlled by human driving logs. By swapping out only the agents in identical scenarios, we can isolate errors caused by the inability to anticipate other agents' actions.

Figure 5 compares the effectiveness of BC, PPO, and HR-PPO agents in different evaluation modes. PPO performs well when interacting with agents of the same kind but struggles when facing unseen human driver replay agents. Overall, there's a significant increase in collision rates, exceeding 20%, when switching from self-play mode to log-replay mode. HR-PPO also experiences a rise in collision rates, but to a lesser extent, with an increase of 7%. In log replay, HR-PPO outperforms the base BC agent in terms of collision rates while also achieving a much higher goal rate.

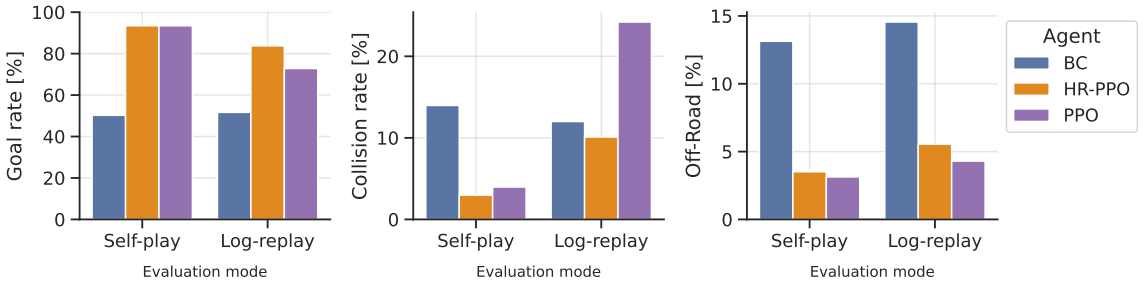

Figure 5: Overall performance gap between evaluating in self-play vs. log-replay settings across the 200 training scenarios.

The effectiveness of HR-PPO agents in coordinating is more visible when we examine the collision rate as a function of the number of intersecting paths vehicles encounter (Details in Section E.3), which is shown in Figure 6. Notably, the collision rate for PPO consistently increases as trajectories become more interactive, with collisions occurring between 40-65% of vehicles when encountering one

or more intersecting paths. In contrast, the collision rate for HR-PPO shows only a slight increase of approximately 5-8% compared to its self-play collision rate, remaining relatively stable regardless of scene interactivity. It is worth noting that this improvement is not quite evident based on the aggregated performance metrics because more than 70% of all agent trajectories in the dataset do not intersect with other vehicles. Altogether, our results suggest that HR-PPO agents are more compatible with human driving behavior.

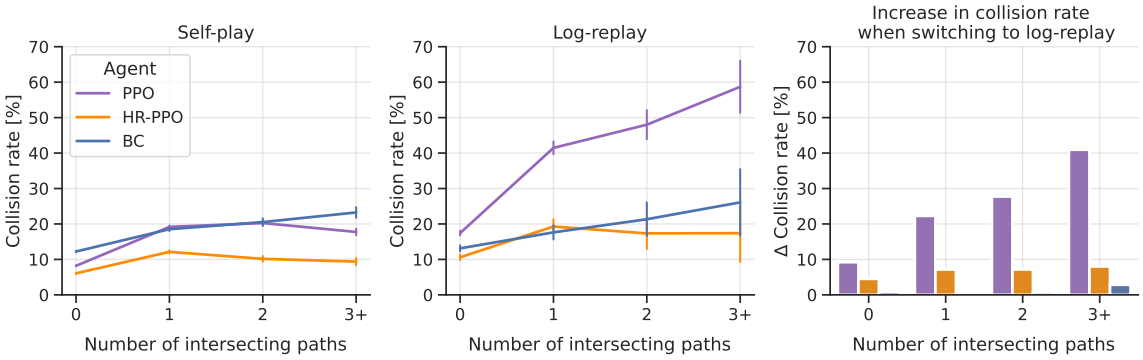

Figure 6: Collision rate as a function of the number of intersecting paths (a proxy for interactivity) of a vehicle trajectory on the training dataset.

What makes HR-PPO agents more compatible with the human logs? To find out, we conduct a qualitative analysis. After analyzing the driving behavior of PPO and HR-PPO agents in 50 randomly sampled scenarios, we conclude that the lower collision rates can be attributed to two main factors. First, the HR-PPO agent's driving style aligns better with human logs, enabling a higher level of anticipation of other agents' actions. Secondly, HR-PPO agents maintain more distance from other vehicles, which reduces the risk of collisions. A subset of videos are available at https://sites.google.com/view/driving-partners

Regularization ensures that policies are more consistent with a reference distribution, in our case the human driving logs. This is also evident when we plot the statistical divergence between policies during training as shown in Figure 7. On the left side, we see that the PPO and HR-PPO learning curves are similar, indicating that both agents learn to navigate effectively. On the right side, we plot the KL divergence between the human and RL policies across training. In the case of PPO, the divergence increases indefinitely, while for HR-PPO, the divergence remains small. Although both policies seem to converge from the reward curves, the resulting driving behaviors are fundamentally different.

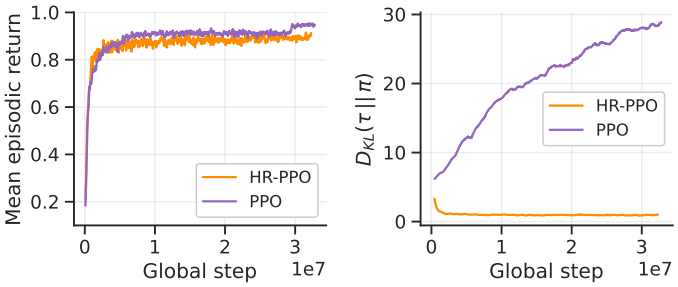

Figure 7: Comparison between a sampled PPO run (purple) and an HR-PPO run with a regularization parameter $\lambda = 0.06$ (orange). Left: Episodic returns averaged over rollouts. Right: The KL-divergence between the human reference policy $\tau(\cdot \mid o)$ and the RL policy $\pi(\cdot \mid o)$ and the entropy of the human reference policy evaluated over the distribution of states visited by π: $H(\tau(\cdot \mid o))$.

3.6 HR-PPO failure cases

We analyzed 100 scenarios each from the train and test datasets in log-replay mode to understand the type of errors HR-PPO agents make. We identify three types of failure modes and describe them below. Example videos for each group are shared on our project page under "HR-PPO failure cases". Of the 200 sampled scenarios, 6% of the training dataset and 21% of the test dataset has a failure. The failures are broken down as follows:

1. **Sharp turns**: About 25% of failures result from off-road events due to challenging turns or target positions.
2. **Coordination**: Approximately 35% of failures are due to collisions resulting from the failure to anticipate human driving log behavior.
3. **Setting-related bugs/failures**: Around 35% of failures are caused by unreachable target positions or other noise/errors in the dataset.

This indicates that while there is room for improvement in coordination, the majority of failure cases (approximately 60%) are due to dataset bugs or kinematically challenging goal positions or routes. Due to the high collision rate of the BC policy (26-33% combined, of which 12-14% vehicle collisions and 14-19% off-road events), it is difficult to see if it is actually experiencing coordination failures. Therefore, studying the qualitative differences between coordination failures between HR-PPO and BC is left for future work.

4 Related work

Driving agents in simulation. There are four major approaches used in existing traffic simulators to model human drivers. One class of methods uses **low-dimensional car following models** to describe the dynamics of vehicle movement through a small number of variables or parameters (Kreutz & Eggert, 2021; Kesting et al., 2007; Treiber et al., 2000). **Rule-based agents** have a fixed set of behaviors. Examples of rule-based agents in driving simulators include car-following agents (Gulino et al., 2023; Caesar et al., 2021; Lopez et al., 2018; Casas et al., 2010) such as the IDM model and behavior agents that can be parameterized to drive more cautiously or aggressively such as CARLA's TrafficManager (Dosovitskiy et al., 2017). While car-following and rule-based agents can respond to other agents and thus provide interactivity, it can be challenging for them to capture the full complexity of human driving behavior and these agents frequently experience non-physical accelerations or come to a deadlock in complex interactions. Some simulators provide the **recorded human driving logs** which can be replayed to allow for interactions (Lu et al., 2023; Vinitsky et al., 2022; Gulino et al., 2023; , FAIR; Caesar et al., 2021). Although these static models produce realistic trajectories, they cannot respond to changes in the environment, such as other drivers. Finally, some driving simulators include **learning-based agents** using reinforcement learning (Li et al., 2022), however, these agents likely do not resemble human behavior. Our Human-Regularized PPO approach aims to produce simulation agents that meet all these criteria to allow for the controlled generation of challenging real-life interactions in simulation.

Imitation Learning and Supervised Learning. A canonical approach for developing learning-based driving policies for autonomous driving has been through Imitation Learning (IL) (Pomerleau, 1988; Bojarski et al., 2016; Xu et al., 2023) and other supervised methods such as trajectory prediction (Philion et al., 2023) and language-conditioned traffic scene generation (Tan et al., 2023). IL works by mimicking expert behavior using recorded actions from human drivers. There are two broad classes of IL: *open-loop* and *closed-loop*. Open-loop methods, like Behavioral Cloning (BC), learn a policy without taking into account real-time feedback. As such, one limitation of open-loop IL methods is that they suffer from compounding errors once deployed in closed-loop systems (Ross et al., 2011). Closed-loop IL (Ng et al., 2000; Ho & Ermon, 2016; Fu et al., 2017; Igl et al., 2022; Baram et al., 2017; Suo et al., 2021) improves upon this by letting the system adjust its actions through ongoing interaction with the environment during training. While these methods provide

enhanced robustness, they have not yet achieved high closed-loop performance when all agents are controlled. In addition, our approach does not rely on large, high-quality datasets of human driving data.

Multi-Agent Reinforcement Learning. Reinforcement learning techniques have been effective in developing capable agents without requiring human data (Silver et al., 2016; 2018; Vinyals et al., 2019) in zero-sum and collaborative games. While this approach has worked in a range of games (Strouse et al., 2021; Bard et al., 2020), many games have multiple equilibria such that agents trained in self-play do not perform well when matched with human-partners (Bakhtin et al., 2021; Hu et al., 2020). In the driving setting, this challenge can partly be ameliorated through the design of reward functions that encode how people drive and behave in traffic interactions (Pan et al., 2017; Liang et al., 2018). However, it is not entirely clear what reward function corresponds to human driving and the inclusion of this type of reward shaping can create undesired behaviors (Knox et al., 2023). An alternate approach tries to create human compatibility through the design of training procedures that restrict the set of possible equilibria (Hu et al., 2020; 2021) by ruling out equilibria that humans are unlikely to play.

Combined IL + (MA)RL. Recent work has shown that augmenting IL with penalties for driving mistakes can create more reliable policies. This has been demonstrated in both closed-loop (Zhang et al., 2023; Wu et al., 2023) and open-loop (Lu et al., 2023) settings. Outside of the driving domain, augmenting goal-conditioned single-agent reinforcement learning has been found to enhance performance in the Arcade Learning Environment (ALE) Hester et al. (2018) and improve the likelihood of convergence to the equilibrium in certain multi-agent learning settings (Lerer & Peysakhovich, 2019; Hu et al., 2022). In multi-agent settings, it has empirically been shown to yield policies more compatible with existing social conventions of the human reference group (Jacob et al., 2022; , FAIR; Bakhtin et al., 2022). Our approach extends these works to the driving setting where it has not yet been investigated in prior work if this type of data-driven regularization is sufficient to enable convergence to a human-compatible policy.

5 Conclusion and future work

We presented Human-Regularized PPO (HR-PPO), a multi-agent RL-first approach that yields effective goal-reaching agents that are more aligned with human driving conventions. We show that HR-PPO agents achieve a high goal rate and low collision rate in a variety of multi-agent traffic scenarios and exhibit human-like driving behavior according to several proxy measures. They also demonstrate significant advancements in coordinating with human drivers compared to BC policies trained directly on human demonstrations or PPO without regularization.

Several interesting challenges remain for future work. Firstly, due to computational constraints, we limit training to a dataset of 200 traffic scenarios. We expect that scaling our approach to more scenarios will enhance the generalization capabilities of agents and close the observed generalization gap between train and test scenes. We also note that reported performance was from policies that were still learning, indicating that better performance can be achieved with a faster simulation setup or training for more steps. Furthermore, we expect that by improving the quality of the behavioral cloning policies, the performance of the HR-PPO agents can be significantly enhanced. Although the agents ignore many of the bad actions output by the BC model, they still imitate some of the suboptimal actions, which can be observed by the increase in off-road rate as regularization increases. Additionally, it is still to be seen if the agent generalization gap can be closed simply by increasing the capability of the BC policy using more complex imitation methods such as GAIL (Ho & Ermon, 2016) or better architectures such as Diffusion Policies (Chi et al., 2023).

There are also opportunities for improving the evaluation of human-like driving agents. The desired measure of performance is compatibility with human drivers, which can only be truly assessed via real-world driving. Our current proxy measure for this real-world performance, testing in log-replay, is imperfect as these drivers are not reactive. This both limits our ability to coordinate with them

and also does not illuminate potential failure modes that could occur under reactivity. Alternative proxy measures that could be considered in future work include testing across multiple seeds (referred to as cross-play in the zero-shot coordination literature), testing with a variety of reactive agents such as the IDM agents included in Waymax (Gulino et al., 2023) and NuPlan (Caesar et al., 2021), or driving alongside humans operating in virtual reality.

Finally, there remain unresolved theoretical questions about the soundness of this approach. In contrast to other works applying this type of regularization in the game literature, we do not have access to the ground truth reward function. As such, we are relying on imitation learning to implicitly complete these portions of the reward. It is not clear if the KL loss used can compensate for these missing terms. Additionally, it would be interesting to understand whether there are settings under which the inclusion of data drawn from the equilibrium can guarantee approximate convergence to the equilibrium.

Acknowledgments

This work is funded by the C2SMARTER Center under Grant Number 69A3552348326 from the U.S. Department of Transportation's University Transportation Centers Program. This work was also supported in part through the NYU IT High-Performance Computing resources, services, and staff expertise. We are thankful to Mert Çelikok, Aditya Makkar, Sam Sokota, Graham Todd, Xieyuan Zhang, and Yutai Zhou, for their feedback on early versions of this draft. We thank Cole Gulino for taking the time to answer our questions. We also thank Franklin Yiu, Alex Tang, and Aarav Pandya for visualizations, last-minute debugging, and being all-around helpful.

References

Anton Bakhtin, David Wu, Adam Lerer, and Noam Brown. No-press diplomacy from scratch. *Advances in Neural Information Processing Systems*, 34:18063–18074, 2021.

Anton Bakhtin, David J Wu, Adam Lerer, Jonathan Gray, Athul Paul Jacob, Gabriele Farina, Alexander H Miller, and Noam Brown. Mastering the game of no-press diplomacy via human-regularized reinforcement learning and planning. *arXiv preprint arXiv:2210.05492*, 2022.

Nir Baram, Oron Anschel, Itai Caspi, and Shie Mannor. End-to-end differentiable adversarial imitation learning. In *International Conference on Machine Learning*, pp. 390–399. PMLR, 2017.

Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Jordi Casas, Jaime L Ferrer, David Garcia, Josep Perarnau, and Alex Torday. Traffic simulation with aimsun. *Fundamentals of traffic simulation*, pp. 173–232, 2010.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9710–9719, 2021.

Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

Adam Gleave, Mohammad Taufeeque, Juan Rocamonde, Erik Jenner, Steven H. Wang, Sam Toyer, Maximilian Ernestus, Nora Belrose, Scott Emmons, and Stuart Russell. imitation: Clean imitation learning implementations. arXiv:2211.11972v1 [cs.LG], 2022. URL https://arxiv.org/abs/2211.11972.

Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, et al. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. *arXiv preprint arXiv:2310.08710*, 2023.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pp. 4399–4410. PMLR, 2020.

Hengyuan Hu, Adam Lerer, Brandon Cui, Luis Pineda, Noam Brown, and Jakob Foerster. Off-belief learning. In *International Conference on Machine Learning*, pp. 4369–4379. PMLR, 2021.

Hengyuan Hu, David J Wu, Adam Lerer, Jakob Foerster, and Noam Brown. Human-ai coordination via human-regularized search and learning. *arXiv preprint arXiv:2210.05125*, 2022.

Maximilian Igl, Daewoo Kim, Alex Kuefler, Paul Mougin, Punit Shah, Kyriacos Shiarlis, Dragomir Anguelov, Mark Palatucci, Brandyn White, and Shimon Whiteson. Symphony: Learning realistic and diverse agents for autonomous driving simulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2445–2451. IEEE, 2022.

Athul Paul Jacob, David J Wu, Gabriele Farina, Adam Lerer, Hengyuan Hu, Anton Bakhtin, Jacob Andreas, and Noam Brown. Modeling strong and human-like gameplay with kl-regularized search. In *International Conference on Machine Learning*, pp. 9695–9728. PMLR, 2022.

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

W Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. Reward (mis) design for autonomous driving. *Artificial Intelligence*, 316:103829, 2023.

Karsten Kreutz and Julian Eggert. Analysis of the generalized intelligent driver model (gidm) for uncontrolled intersections. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 3223–3230. IEEE, 2021.

Adam Lerer and Alexander Peysakhovich. Learning existing social conventions via observationally augmented self-play. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 107–114, 2019.

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475, 2022.

Xiaodan Liang, Tairui Wang, Luona Yang, and Eric Xing. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 584–599, 2018.

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. URL https://elib.dlr.de/124092/.

Yiren Lu, Justin Fu, George Tucker, Xinlei Pan, Eli Bronstein, Rebecca Roelofs, Benjamin Sapp, Brandyn White, Aleksandra Faust, Shimon Whiteson, et al. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7553–7560. IEEE, 2023.

Nico Montali, John Lambert, Paul Mougin, Alex Kuefler, Nicholas Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Yang, Shimon Whiteson, et al. The waymo open sim agents challenge. *Advances in Neural Information Processing Systems*, 36, 2024.

Khan Muhammad, Amin Ullah, Jaime Lloret, Javier Del Ser, and Victor Hugo C de Albuquerque. Deep learning for safe autonomous driving: Current challenges and future directions. *IEEE Transactions on Intelligent Transportation Systems*, 22(7):4316–4336, 2020.

Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2980–2987. IEEE, 2023.

Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, pp. 2, 2000.

Xinlei Pan, Yurong You, Ziyan Wang, and Cewu Lu. Virtual to real reinforcement learning for autonomous driving. *arXiv preprint arXiv:1704.03952*, 2017.

Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Learning the language of driving scenarios. *arXiv preprint arXiv:2312.04535*, 2023.

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

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Stéphane 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*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34:14502–14515, 2021.

Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10400–10409, 2021.

Shuhan Tan, Boris Ivanovic, Xinshuo Weng, Marco Pavone, and Philipp Kraehenbuehl. Language conditioned traffic generation. *arXiv preprint arXiv:2307.07947*, 2023.

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

Eugene Vinitsky, Nathan Lichtlé, Xiaomeng Yang, Brandon Amos, and Jakob Foerster. Nocturne: a scalable driving benchmark for bringing multi-agent learning one step closer to the real world. *Advances in Neural Information Processing Systems*, 35:3962–3974, 2022.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Jingda Wu, Yanxin Zhou, Haohan Yang, Zhiyu Huang, and Chen Lv. Human-guided reinforcement learning with sim-to-real transfer for autonomous navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Danfei Xu, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Bits: Bi-level imitation for traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2929–2936. IEEE, 2023.

Chris Zhang, James Tu, Lunjun Zhang, Kelvin Wong, Simon Suo, and Raquel Urtasun. Learning realistic traffic agents in closed-loop. *arXiv preprint arXiv:2311.01394*, 2023.

A Data distribution and scene information

Train dataset. The left side of Figure 8 displays the distribution of the number of vehicles in our training dataset of 200 traffic scenarios. On average, a scenario has 12 vehicles, with a maximum of 43. During training we control all vehicles in a scene up to a maximum of 50 controlled vehicles. Therefore, we always control all vehicles in the scene. On the right-hand side, we plot the distribution of intersecting paths, where we have a total of 3,489 vehicle trajectories. We observe that in most cases, expert vehicle trajectories do not intersect, which means 73% of the expert vehicles can reach their target position without crossing the path of another vehicle. Of the remaining 27% of vehicles whose paths intersect, most intersect once (19%, which is 667 vehicles), and a small set has two (5%; 182 vehicles) or three or more (3%; 104 vehicles) intersections.

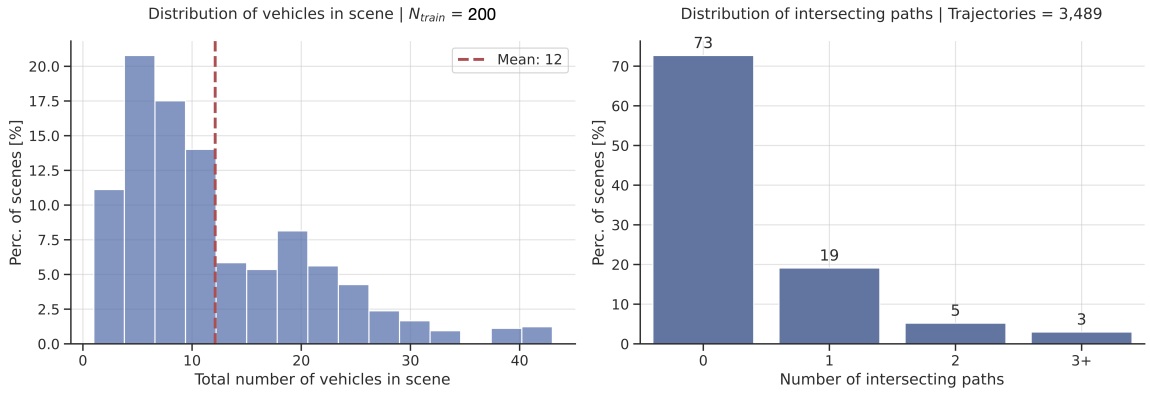

Figure 8: Train data distribution; 200 scenarios.

Test dataset. Our test dataset consists of 10,000 scenarios. Figure 9 shows the distribution of vehicles and intersecting paths in the test set, which is similar to the train dataset.

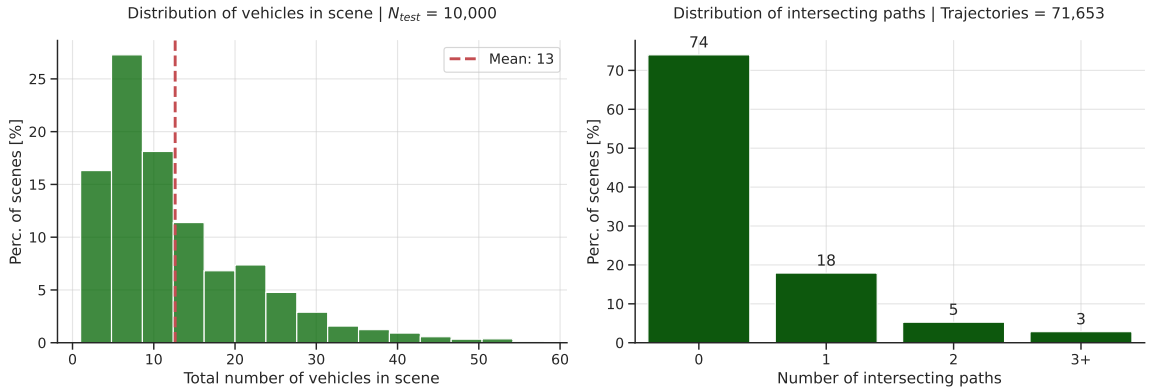

Figure 9: Test data distribution; 10,000 scenarios.

Test dataset | Vehicles per scene	
count	71653.00
mean	12.63
std	9.46
min	1.00
25%	6.00
50%	10.00
75%	17.00
max	58.00

B Expert demonstrations

Table 5 contrasts the performance of the expert agents under different conditions: Expert-*teleport* indicates the performance of agents that are stepped using the recorded position logs, Expert-*actions* the performance of agents stepped using the inferred expert actions.

Table 5: Expert performance and effect of discretization. Tested in 2,000 random traffic scenarios. We control a single vehicle and step the remaining vehicles in the scene in log-replay mode.

Agent	**Action space**	**Action dim**	**Controlled vehicle**	Off-road Rate (%)	Collision Rate (%)	Goal Rate (%)
Expert-*teleport*	-	-	AV only	0	0	100
Expert-*actions*	Bicycle Continuous	-	AV only	5.1	1.1	85.7
Expert-*actions*	Bicycle Continuous	-	Random	6	1.8	84
Expert-*actions*	Bicycle Discrete	31 x 101	AV only	5.1	1.2	83.5
Expert-*actions*	Bicycle Discrete	21 x 31	AV only	9.2	3.3	78.0
Expert-*actions*	Bicycle Discrete	21 x 31	Random	12.2	4.3	67.9

Several randomly sampled trajectories from the dataset. The green circle represents the tolerance region around the goal position.

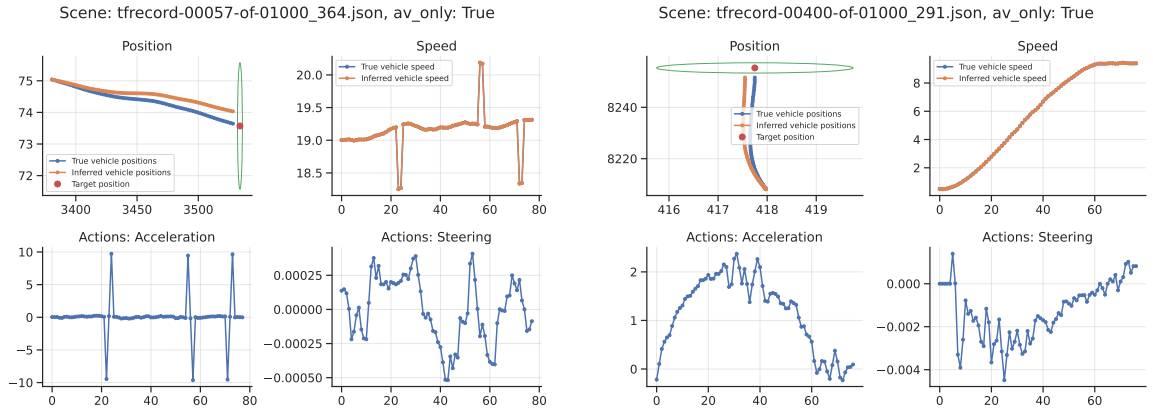

Figure 10: AV trajectories

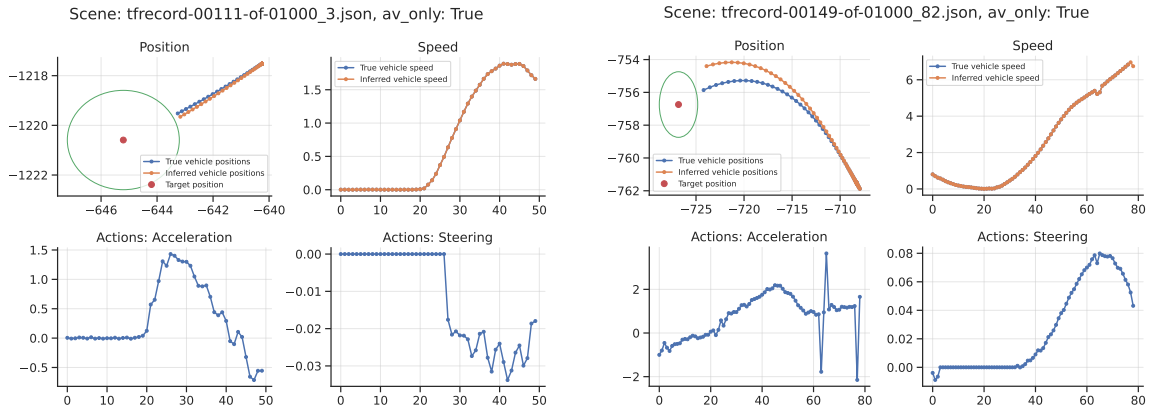

Figure 11: AV trajectories

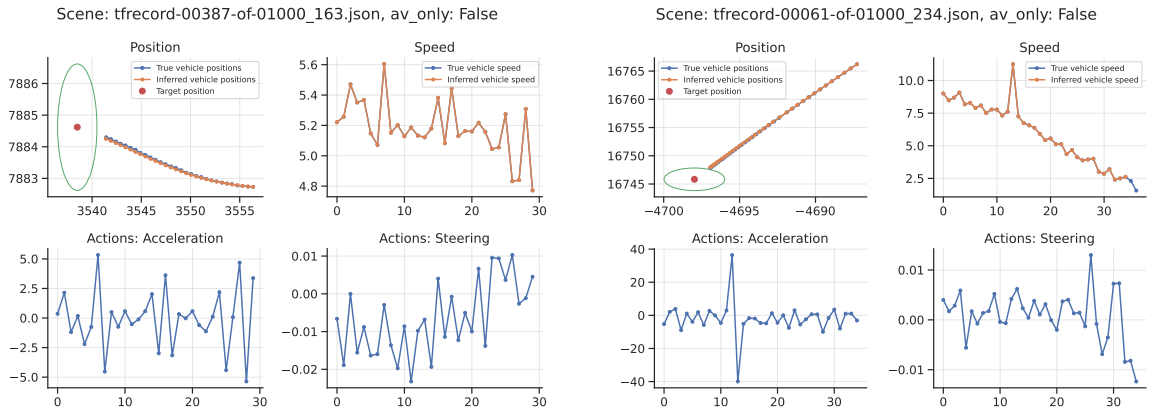

Figure 12: non-AV trajectories

C Learned human reference policy distributions

The KL divergences obtained during the training process of Human-Regularized PPO are influenced by the form of the observation-conditioned pre-trained human-policy distributions: $\tau(a \mid o_t)$. This section examines these distributions in detail. We use the entropy, or average Shannon information H (Shannon, 1948), to quantify the level of uncertainty in a distribution:

$$H(\tau(a \mid o_t)) = -\sum_{a \in \mathcal{A}} p(a) \ln(p(a))$$

with $\mathcal{A} = \{0, 1, 2, \ldots, 651\}$ being our chosen joint action space where every integer points to an acceleration, steering pair. For instance, the integer 325 points to the acceleration value 0 and the steering wheel angle of 0 radians, meaning that the vehicle is moving straight at a constant speed.

Table 6 presents the entropy and probability of the sampled actions for the human reference policy trained solely on AV demonstrations and Figure 13 (Left) displays the boxenplots. As expected, we observe that the entropy for the *seen* AV instances in the training dataset ($H = 0.10 \pm 0.27$) is notably lower than the entropy observed for *unseen instances*, namely the test set and/or non-AV vehicles ($H \approx 0.26 \pm 0.41$). To put these values into perspective, note that the upper bound on the entropy is given by the entropy of a perfectly uniform distribution with the size of our action space:

$$H = \ln(n = 651) \approx 6.48$$

As such, the imitation learning policy yields high-certainty distributions overall, particularly for the AV vehicles. This is also evident when we look at a few example action distributions for AV vehicles in Figure 14 and for non-AV vehicles in Figure 15.

Table 6: Entropy of the human reference policy τ trained on only the AV demonstrations. Estimates are based on $\sim 20,000$ samples from 200 random traffic scenarios.

Dataset	Vehicle type	Entropy	Avg. Prob. of sampled action
Train	AV	0.10 ± 0.27	0.69 ± 0.45
	Non-AV	0.26 ± 0.41	0.68 ± 0.42
Test	AV	0.27 ± 0.41	0.66 ± 0.43
	Non-AV	0.26 ± 0.41	0.68 ± 0.42

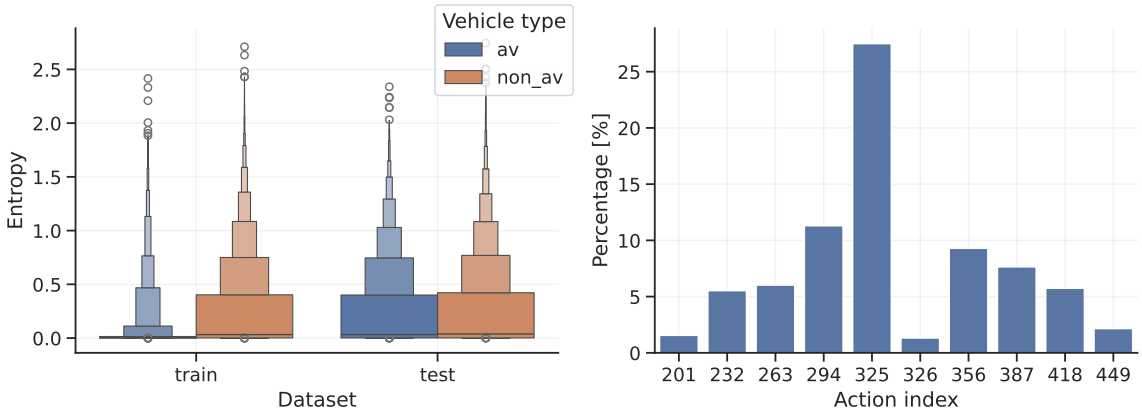

Figure 13: Left: Entropy for action probability distributions, $\tau(a \mid o_t)$; Right: The top 10 most occurring action indices in the human policy predictions. Together they make up 78 % of all predictions.

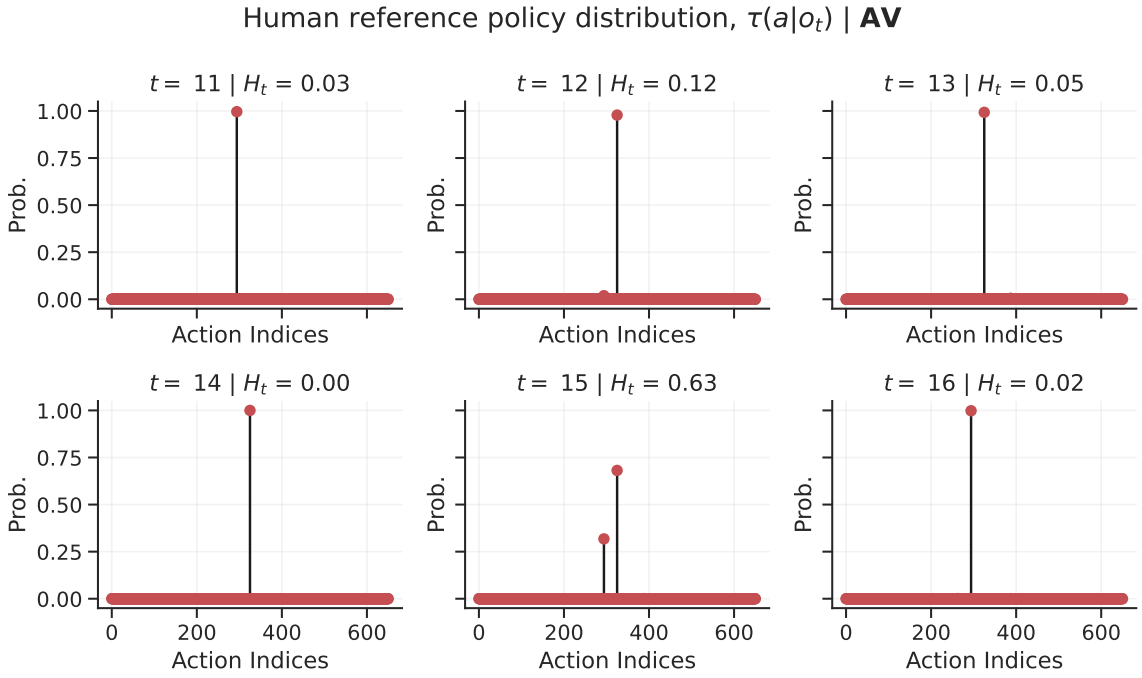

Figure 14: Probability distributions from the human reference policy trained on AV data only.

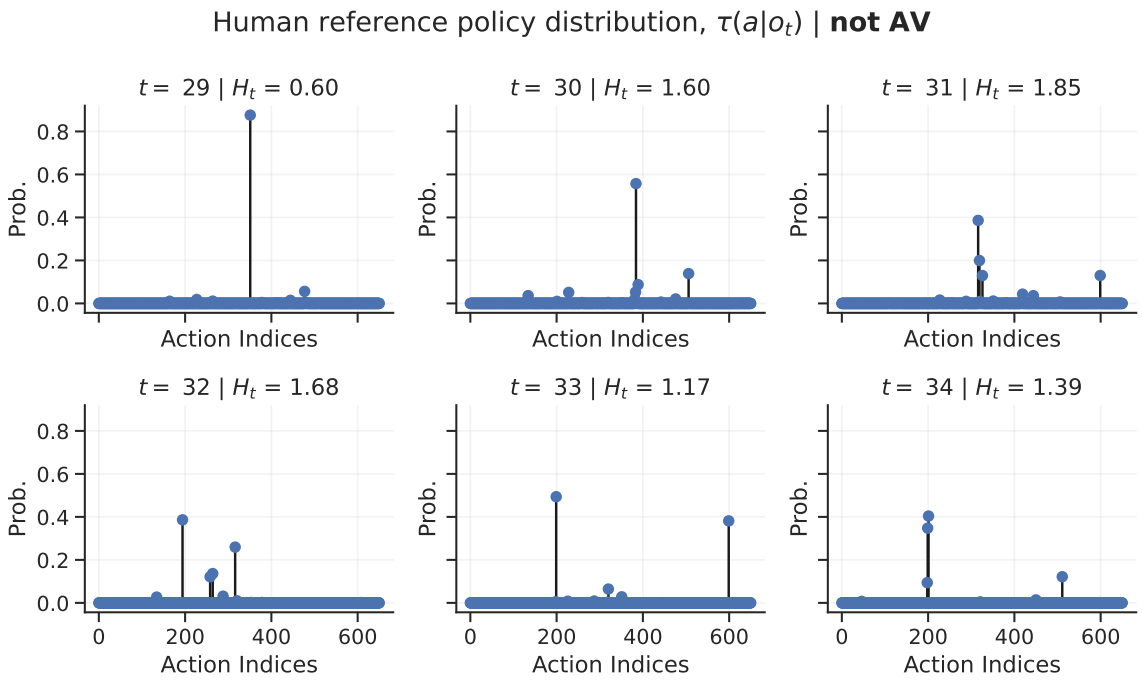

Figure 15: Probability distributions from the human reference policy trained on AV data only.

D Implementation details

D.1 PPO

Proximal Policy Optimization (PPO) (Schulman et al., 2017) optimizes the following surrogate objective:

$$\mathcal{L}_t^{\text{PPO}}(\theta) = \hat{\mathbb{E}}\left[L_t^{\text{CLIP}}(\theta) - c_1 L_t^{\text{VF}} + c_2 S[\pi_\theta](o_t)\right]$$

where we use a value function coefficient of $c_1 = 0.5$ and an entropy coefficient of $c_2 = 0.001$ during training. Here, $L_t^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}\left[\min(r^t(\theta)\hat{A}^t), \text{clip}(r^t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}^t\right]$ is a lower bound on the clipped advantage, S denotes an entropy value to encourage exploration, and $L^{\text{VF}} = (v - \hat{v})^2$ is the squared error between the target and predicted state-values.

D.2 Network architecture

The agent observations contain multi-modal data. To process different types of data efficiently, we initially process them separately and then combine them using a late-fusion architecture (Nayakanti et al., 2023). We first process every modality independently and then apply a max-pool operation to flatten the embeddings. This ensures permutation invariance, meaning that the network is insensitive to the rearrangement of objects, such as road vehicles or road graph points, in the input. Figure 16 depicts our network architecture.

D.3 Hyperparameters

See Table 7 for an overview of the hyperparameters used for PPO and HR-PPO. For HR-PPO, we experimented with human regularization weights $\lambda \in \{0.001, 0.005, 0.02, 0.04, 0.05, 0.06, 0.08, 0.1, 0.2\}$ and use most of the default parameters from stable baselines. The overall best HR-PPO model was trained with a regularization weight of 0.06. All other hyper-parameters are identical between the

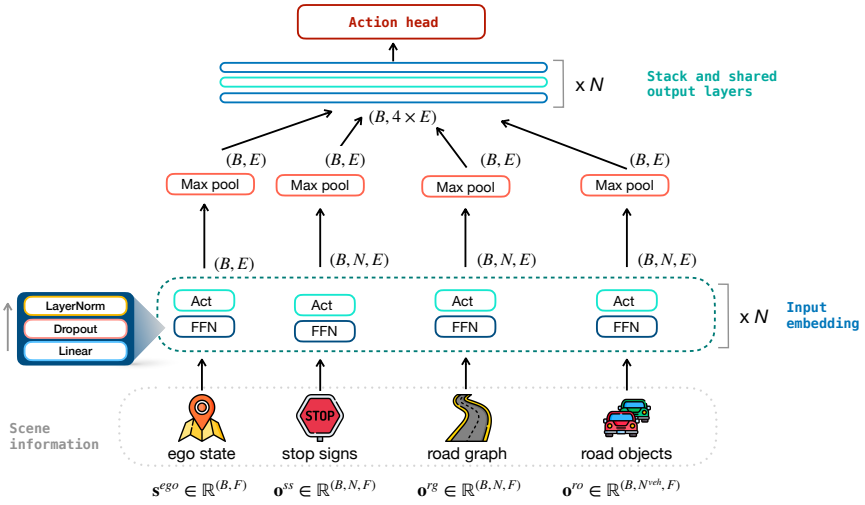

Figure 16: PPO and HR-PPO network architecture.

settings. For the single-agent training runs, we multiply the rollout length by five to increase the batch size.

Table 7: Hyperparameters used for training in `Nocturne` scenarios.

Parameter	PPO	HR-PPO
γ	0.99	0.99
λ_{GAE}	0.95	0.95
PPO rollout length	4096	4096
PPO epochs	10	10
PPO mini-batch size	512	512
PPO clip range	0.2	0.2
Adam learning rate	3e-4	3e-4
Adam ϵ	1e-5	1e-5
normalize advantage	yes	yes
entropy bonus coefficient	0.001	0.001
value loss coefficient	0.5	0.5
human regularization coefficient λ	0.0	0.06
total timesteps	140 M	140 M
seed	42	42

D.4 Compute

We ran all experiments on a training dataset of 200 scenarios for 140 million steps. Every run took approximately 5 days on a single GPU (A100 or NVIDIA Quadro RTX 8000).

E Evaluation metrics

E.1 Realism metrics

Goal-Conditioned Average Displacement Error (GC-ADE). Measures how far the trained driving policy deviates from the logged human driving behavior conditioned on knowing the agent goal. Let $\mathbf{x}^{\text{H}} = ((x_0, y_0), \ldots, (x_T^{\text{H}}, y_T^{\text{H}}))$ be a vector with the logged step-wise (x,y) positions of a human driver and $\mathbf{x}^{\pi} = ((x_0, y_0), \ldots, (x_T^{\pi}, y_T^{\pi}))$ trajectory resulting from the predicted policy actions

in closed-loop. Since the end times T^{H}, T^{π} can be different, we define $T = \min(T^{\mathrm{H}}, T^{\pi})$ and compute the GC-ADE as follows:

$$\text{GC-ADE}(\mathbf{x}^{\mathrm{H}}, \mathbf{x}^{\pi}) = T^{-1}\sqrt{\sum_{t=1}^{T}(\mathbf{x}_t^{\mathrm{H}} - \mathbf{x}_t^{\pi})^2}$$

Mean Absolute Steering Error. Measures how much the trained driving policy steering wheel action values deviate from the inferred human driving actions. Let $\mathbf{a}^{\mathrm{H}} = (s_0, \ldots, s_T^{\mathrm{H}})$ be a vector with the logged steering wheel angles from a human driver and $\mathbf{a}^{\pi} = (s_0, \ldots, s_T)$ be the policy-predicted acceleration values. Since the end times T^{H}, T^{π} can be different, we define $T = \min(T^{\mathrm{H}}, T^{\pi})$ and compute the MAE as follows:

$$\text{MAE}_{\text{steer}} = \frac{1}{T}\sum_{t=1}^{T}|s_t^{\mathrm{H}} - s_t^{\pi}|$$

Mean Absolute Acceleration Error. Measures how much the trained driving policy acceleration action values deviate from the inferred human driving actions. Let $a^{\mathrm{H}} = (a_0, \ldots, a_T^{\mathrm{H}})$ be a vector with the logged acceleration values from a human driver and $a^{\pi} = (a_0, \ldots, a_T)$ be the predicted acceleration values. We define $T = \min(T^{\mathrm{H}}, T^{\pi})$ and compute the MAE as follows:

$$\text{MAE}_{\text{accel}} = \frac{1}{T}\sum_{t=1}^{T}|a_t^{\mathrm{H}} - a_t^{\pi}|$$

Accuracy to discretized human driver actions. Measures the ratio of the policy-predicted action tuples (acceleration, steering) that matches the discretized human driver action tuple. Note that our action space is size 651.

E.2 Effectiveness metrics

- *Off-Road Rate*: Percentage of vehicles that hit a road edge or barrier.
- *Collision Rate*: Percentage of vehicles that collided with another agent.
- *Goal-Rate*: Percentage of total vehicles that achieved their goal position within an episode.

To calculate the aggregate percentages, we take the total number of agents that meet a given criteria (such as colliding or achieving a goal) across all scenarios and divide it by the total number of agents. For instance, if we have two scenarios with 3 and 2 agents respectively, and in scenario one, 2 agents met their goal and in scenario 2, one agent met their goal, then the goal rate is calculated as $3/5 = 0.6$.

Since the outcomes are binary (either the agent meets the criteria or not), we can estimate the variance by first aggregating the data across scenarios. The standard error for the goal rate is calculated by first computing the scene-le goal ratio for each scenario and then taking the standard deviation across them. For instance, using the example given above, the scene-level goal rates would be 2/3 and 1/2. The standard error would be $\sigma/\sqrt{n} = 0.083/1.414 = 0.0589$ or 5.89 %.

E.3 Interactivity: Computing the intersecting paths for a vehicle

We use the number of intersecting paths as a proxy metric for the level of interactiveness in a scene. To compute the number of intersecting paths for a vehicle i, we follow these steps: We pair vehicle i with every other vehicle in a scenario. For every pair of vehicles (i, j), we step the both in expert-replay mode. If the line segments touch and the time difference between them is less than 5 seconds (50 steps), we increase the intersection count for vehicle i by one. To illustrate various trajectories and scenarios with different numbers of intersecting paths, Figure 17 displays two scenarios with low levels of interactivity (0-1 intersecting paths) and Figure 18 depicts two scenarios with medium to high levels of interactivity.

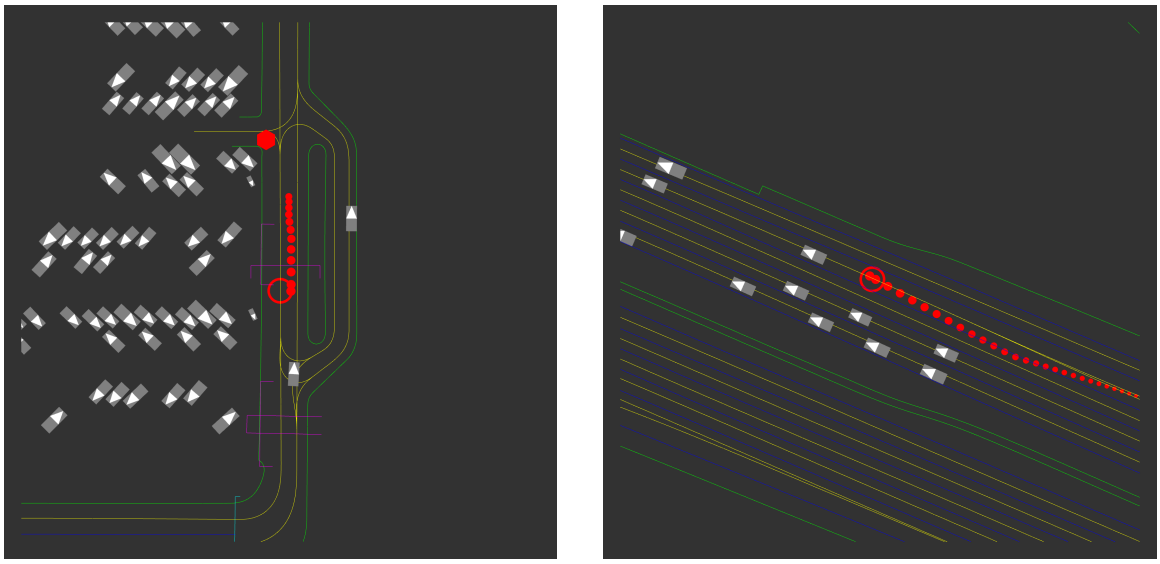

Figure 17: **Example scenarios with a relatively low level of interactivity.** We control the ***red*** vehicle and the *grey* vehicles are stepped using the replayed human logs. Left: The red vehicle, has no intersecting paths. This means that the vehicle can reach its target destination without encountering another vehicle. Right: This vehicle has one intersecting path because its trace touches the trace of the grey vehicle in front of it. Overall, this scenario is more interactive than the left scenario because the controlled vehicle has to consider the moving vehicles around it.

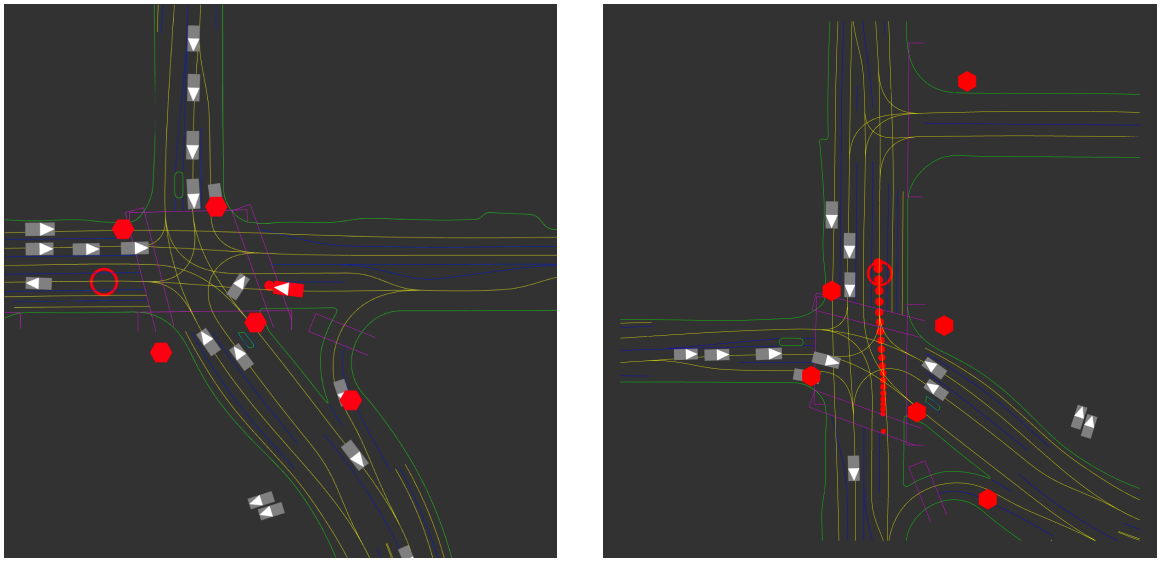

Figure 18: **Example scenarios with medium to high levels of interactivity.** We control the ***red*** vehicle and the *grey* vehicles are stepped using the replayed human logs. Left: The red, controlled, vehicle here has three intersecting paths. Timely coordination between the red vehicle and other vehicles is necessary to reach the goal. When using the log-replay setting, the controlled vehicle must be able to work with the existing trajectories of uncontrolled vehicles that are replayed using static human logs. Right: The red vehicle has five intersecting paths.

F Additional Figures

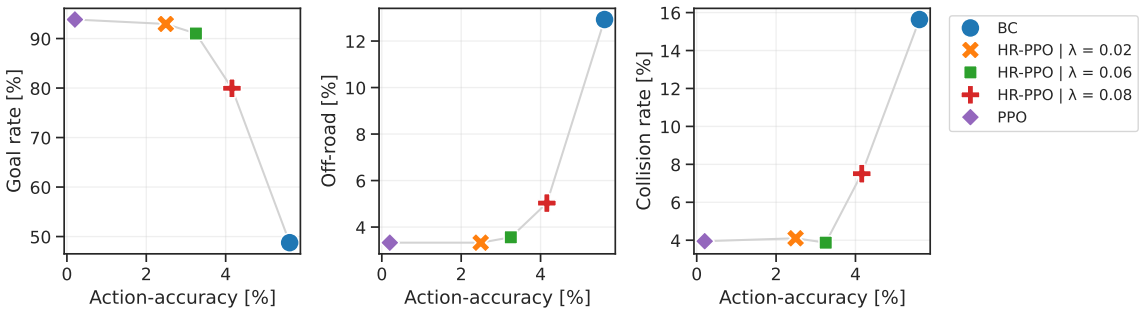

Figure 19: Accuracy to the human actions against effectiveness on 200 scenes in self-play.

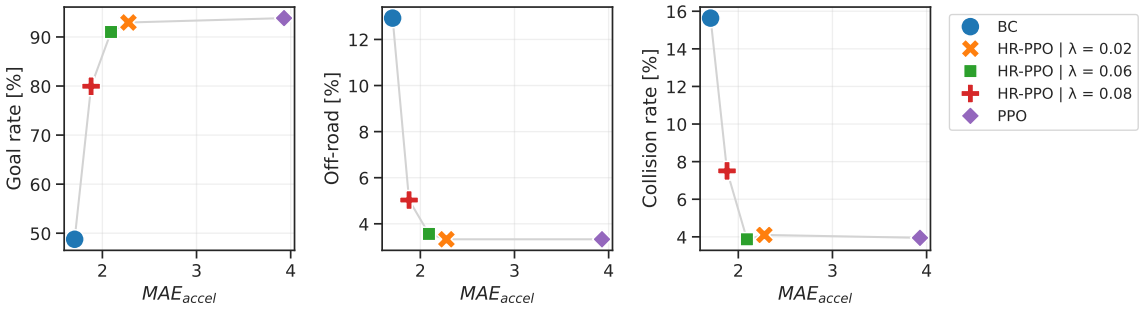

Figure 20: MAE between acceleration values of the logged human drivers and the HR-PPO-predicted acceleration values against effectiveness on 200 scenes in self-play.

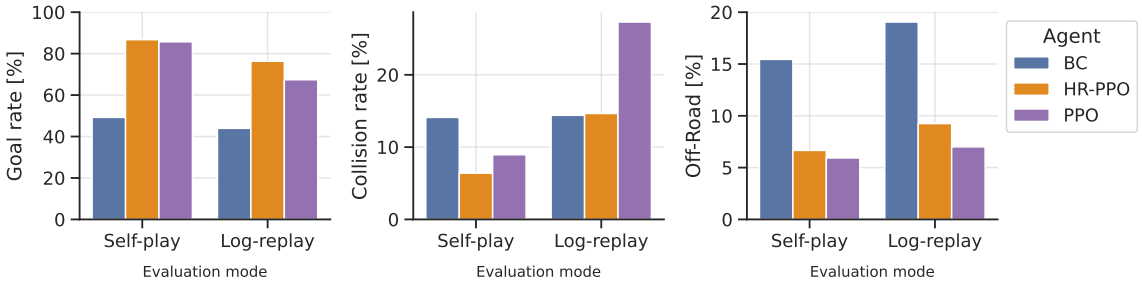

Figure 21: Self-play vs. log-replay performance across the test dataset.

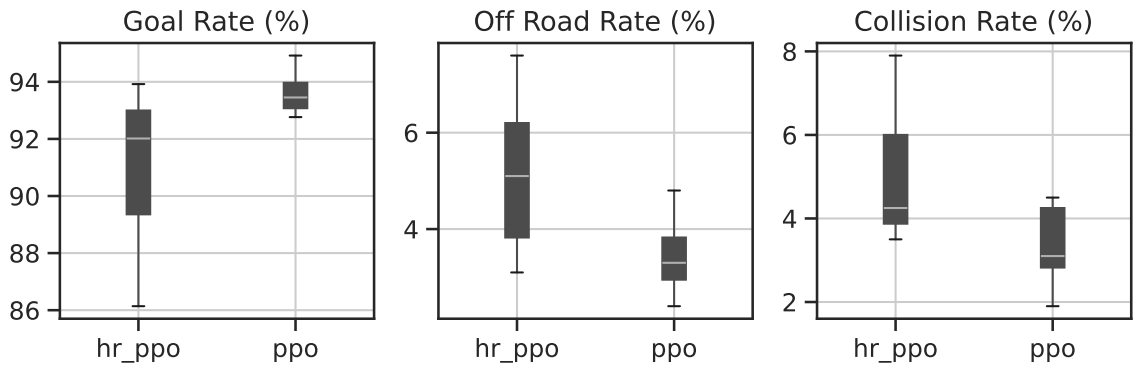

Figure 22: Comparison between PPO and HR-PPO performance across 10 different seeds. Due to computational constraints, we ran these experiments on 50 scenarios instead of the full train dataset of 200 scenarios.

		HR-PPO	**PPO**
Goal Rate (%)	count	10.00	10.00
	mean	91.08	93.58
	std	2.71	0.74
	min	86.14	92.76
	25%	89.36	93.08
	50%	92.01	93.45
	75%	93.00	93.97
	max	93.92	94.92
Off Road (%)	count	10.00	10.00
	mean	5.12	3.48
	std	1.57	0.79
	min	3.10	2.40
	25%	3.83	2.95
	50%	5.10	3.30
	75%	6.20	3.83
	max	7.60	4.80
Collision Rate(%)	count	10.00	10.00
	mean	4.98	3.33
	std	1.59	0.97
	min	3.50	1.90
	25%	3.88	2.82
	50%	4.25	3.10
	75%	6.00	4.25
	max	7.90	4.50

Table 8: PPO and HR-PPO performance across 10 different seeds.