

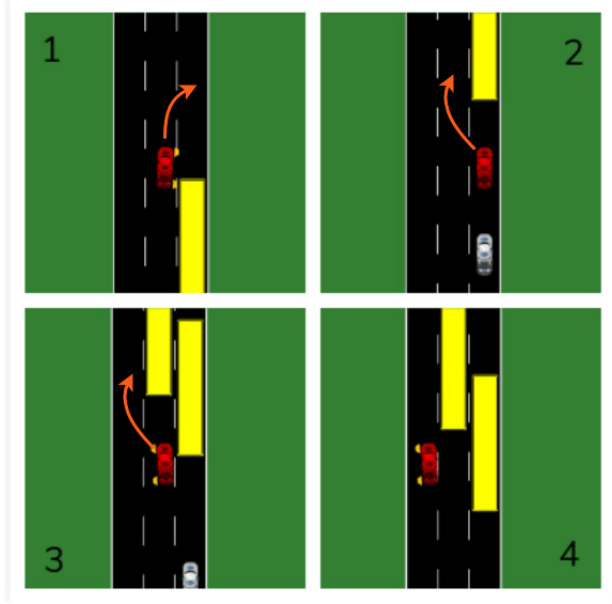


Fig. 1: Top view of a trajectory demonstrated from Alice.

A Highway Use Case

A.1 Expert Demonstrations

As discussed in the previous chapters, we consider trajectories generated from 10 human experts, each left free to drive in the 3D simulator for 50 driving episodes, 40 seconds each. The human drivers were not told what were the driving features that we recorded were, or what goal to follow. This resulted in various driving behaviours generated. In the following figures we show a 2D view from above of some of the behaviors of the different drivers, which indicate the different objectives. Figure 1 shows one of the trajectories demonstrated from the agent Alice, which tries to maintain the right lane of the highway, while also maximizing speed which drives him to perform overtakes and then returning to the right lanes. Carol on the other hand, also demonstrates an interest in maximizing speed which drives him to perform overtakes as in Figure 2, but does not return to the right lanes after performing them, shown not a keen interest in maintaining the right lanes. Finally, Figure 3 shows a typical behavior of Craig, which changes lanes to the right, as soon as possible, and shows no interest in maintaining high speeds since it follows cars in the right lane, without overtaking, hence losing speed to maintain a safe distance.

A.2 Forward RL Training

Here we discuss the details of the training procedure employed in the training of the policies optimizing the reward of the clusters identified by $\Sigma - GIRL$. We employed the techniques proposed in (Likmeta et al., 2020). We trained the agents for 2000 epochs, using PGPE, where each epoch consists in collecting 100 episodes of experience from 5 different rule parameters sampled from the hyperpolicies μ_ρ . The training curves of each of the agents are shown in Figure 4. We can see that each of the agents is optimizing its return, with

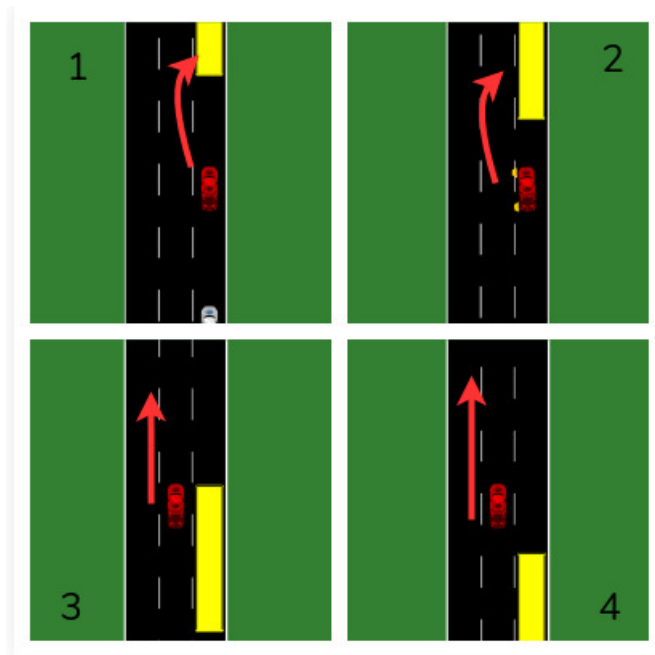


Fig. 2: Top view of a trajectory demonstrated from Carol.

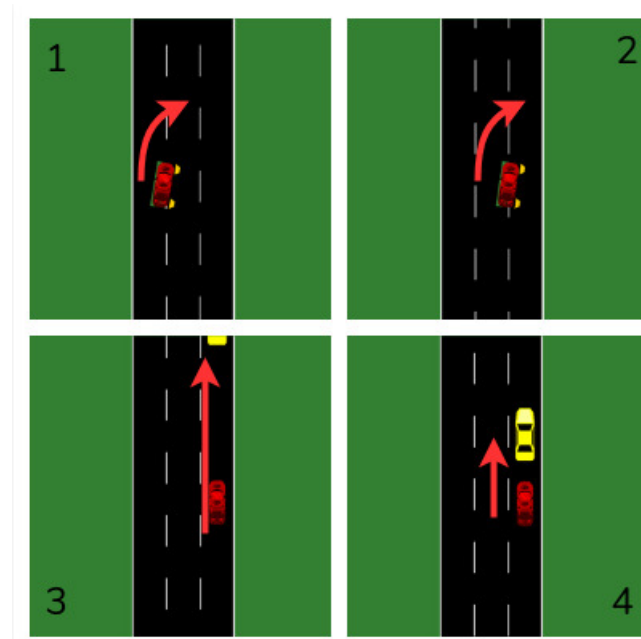


Fig. 3: Top view of a trajectory demonstrated from Craig.

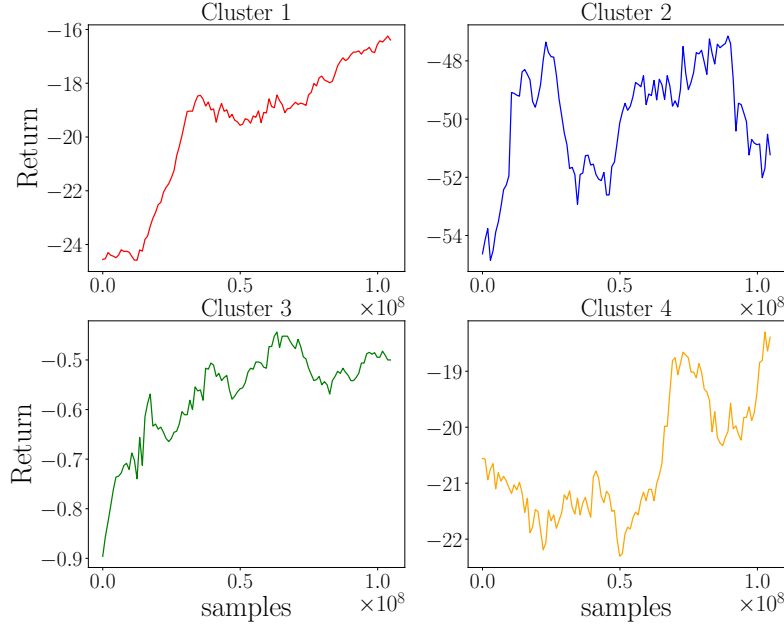


Fig. 4: Training curves of the cluster policies.

some instability, typical of policy gradient methods (as in the case of Cluster 2 or 4). On the other hand, we can see the problem in validating the recovered reward functions by means of forward training with an RL algorithm, as we are not guaranteed that the algorithm will find the optimal policy, as in the case of Cluster 2, where we see that the training procedure is unstable, and the best policy recovered is still suboptimal, performing worse than some of the experts, even though it explicitly optimized that reward function.

B Twitter Use Case

B.1 Clustering Based on Feature Expectations

For completion, we present the results of clustering the twitter experts based on the feature expectations of the expert trajectories. We perform the clustering using a *K-means* with $k = 3$ clusters. Following the clustering, we perform IRL by aggregating the demonstrations of the experts of each cluster, and performing BC on the whole demonstrations of each cluster, as if it was a single expert. Table 1 shows the results of this clustering.

It is not straightforward to compare the results of these clusters with the ones taken from clustering directly based on intentions. Nonetheless, we can notice some differences and commonalities between the two. First cluster 3 in both cases, contains 4 agents (3 of which are the same in both methods). Nonetheless, even though the agents are nearly the same, the recovered weights are fairly different. We believe this is due to the fact of mixing trajectories of different experts and considering them as the same expert when performing BC and then IRL. This aggregation of trajectories might result in a “smoothing” of the policy, that tries

	Reward Features			
	Popularity	N. retweets	δ_{time}	N. agents
Cluster 1	0.48	0.44	0.08	1
Cluster 2	0.72	0.19	0.09	9
Cluster 3	0.54	0.32	0.13	4

Table 1: The reward weights learned by Σ -GIRL on the aggregated trajectories: popularity score of a retweet, number of retweets in a window T , and retweet time (δ_{time}).

to approximate multiple policies and results in an averaging of those policies as discussed in (Almingol and Montesano, 2015). This is also supported by the fact that the weights of the recovered clusters all follow the same “pattern” with a fixed ordering, where the first objective is the most important, followed by the second objective and finally the δ_{time} objective, which again suggests that the BC policies have some kind of “average” behaviour. Finally, another difference compared with the IRL based clustering is that the last objective, δ_{time} always has a really small weight, compared to the other features.

We underline again that these are just some qualitative interpretations that we can do on these results, given that we do not have the ground truth.

References

- Almingol, J. and Montesano, L. (2015). Learning multiple behaviours using hierarchical clustering of rewards. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4608–4613.
- Likmeta, A., Metelli, A. M., Tirinzoni, A., Giol, R., Restelli, M., and Romano, D. (2020). Combining reinforcement learning with rule-based controllers for transparent and general decision-making in autonomous driving. *Robotics and Autonomous Systems*, 131:103568.