Imitation Is Not Enough: Robustifying Imitation with Reinforcement Learning for Challenging Driving Scenarios

Yiren Lu[1], **Justin Fu**[1], **George Tucker**[2], **Xinlei Pan**[1], **Eli Bronstein**[1], **Becca Roelofs**[2], **Benjamin Sapp**[1], **Brandyn White**[1], **Aleksandra Faust**[2], **Shimon Whiteson**[1], **Dragomir Anguelov**[1], **Sergey Levine**[2]
[1]Waymo Research [2]Google Research
{maxlu,justinfu,xinleipan,ebronstein,bensapp,bran,
shimonw,dragomir}@waymo.com[1]
{gjt,rofls,sandrafaust,slevine}@google.com[2]

Abstract

Imitation learning (IL) is a simple and powerful way to use high-quality human driving data, which can be collected at scale, to identify driving preferences and produce human-like behavior. However, policies based on imitation learning alone often fail to sufficiently account for safety and reliability concerns. In this paper, we show how imitation learning combined with reinforcement learning using simple rewards can substantially improve the safety and reliability of driving policies over those learned from imitation alone. In particular, we use a combination of imitation and reinforcement learning to train a policy on over 100k miles of urban driving data, and measure its effectiveness in test scenarios grouped by different levels of collision risk. To our knowledge, this is the first application of a combined imitation and reinforcement learning approach in autonomous driving that utilizes large amounts of real-world human driving data.

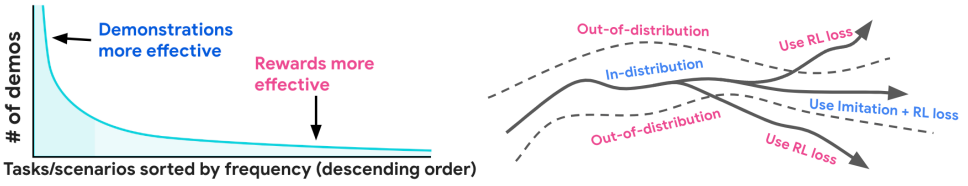
1 INTRODUCTION

Building an autonomous driving system that is deployable at scale has many challenges. First and foremost is the problem of handling the numerous rare and challenging edge cases that occur in real-world driving. To this end, imitative learning based approaches have been proposed that allow the performance of the method to scale with the amount of data available [Pomerleau, 1988, Bojarski et al., 2016, Codevilla et al., 2018]. However, the resulting policies often fail to sufficiently account for safety and reliability concerns. The problem is compounded by complex interactions, where human expert driving data in similar scenarios may be scarce and sub-optimal [Zhou et al., 2022].

Reinforcement Learning (RL) has the potential to resolve this by leveraging explicit reward functions to increase safety awareness. Furthermore, because RL methods train in closed-loop, RL policies can establish causal relationships between observations, actions, and outcomes. This yields policies that are 1) less vulnerable to covariate shifts and spurious correlations commonly seen in open loop IL [Ross et al., 2011, De Haan et al., 2019], and 2) aware of safety considerations that are only implicit in the demonstrations.

However, relying on RL alone, e.g., [Pan et al., 2017, Liang et al., 2018, Zhang et al., 2021a], is also problematic because it heavily depends on reward design, which is an open challenge in autonomous driving [Knox et al., 2021]. Without accounting for imitation fidelity, driving policies trained with RL may be safe but unnatural and may have a hard time making forward progress. IL and RL offer

Machine Learning for Autonomous Driving Workshop at the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, USA.



(a) (b)

Figure 1: **(a)** The demonstration-reward trade-off. As the amount of data for a particular scenario decreases, reward signals become more important for learning. **(b)** The influence of different objectives in our method. For in-distribution states (according to the demonstration data), both IL and RL objectives provide learning signal. For out-of-distribution states, the RL objective dominates, since the IL objective is only applied in-distribution.

complementary strengths: IL increases realism and eases the reward design burden and RL improves safety and robustness, especially in rare and challenging scenarios in the absence of abundant data (Fig. 1a). Prior work has combined IL and RL (e.g., DQfD [Hester et al., 2018], DDPGfD [Vecerik et al., 2017], and DAPG [Rajeswaran et al., 2017]), however, the focus has primarily been on overcoming exploration challenges in environments with known reward functions.

In this paper, we show that by combining IL and RL with *simple* reward functions, we can substantially improve the safety and reliability of policies learned over imitation alone without compromising on human-like behavior. Prior work [Bronstein et al., 2022b] showed that training on a subset of the most "challenging" scenarios improves performance over using the entire dataset. We leverage this insight to 1) restrict training to the most challenging scenarios and 2) to focus our evaluation on the scenarios most likely to exhibit safety and reliability concerns. In particular, we rank the real-world driving segments using a difficulty classifier [Bronstein et al., 2022b] that predicts whether a run segment will result in a collision or near-miss when re-simulated with an internal autonomous driving policy. Our systematic evaluation across varying slices of the dataset by challenge level demonstrates the superiority of this approach over baselines that use only IL or RL.

To summarize, our contributions are: 1) we conduct the first large-scale application of a combined IL and RL approach in autonomous driving utilizing large amounts of real-world human driving data (over 100k miles of real-world urban driving data), and 2) we systematically evaluate its performance and baseline performance by slicing the dataset by difficulty demonstrating that combined IL and RL improves safety and reliability of policies over those learned from imitation alone.

2 RELATED WORK

Learning-based approaches in autonomous driving. We briefly summarize key properties of different learning-based algorithms for planning in Table 1. IL was among the earliest and most popular learning-based approaches adopted for deriving driving policies [Pomerleau, 1988, Bojarski et al., 2016, Zhang et al., 2021b, Vitelli et al., 2022]. Controllable models trained with either IL [Codevilla et al., 2018, Rhinehart et al., 2018] or RL [Liang et al., 2018] allow the user to specify high-level commands in the form of goals or control signals (e.g., left, right, straight) to combine higher-level route planning with low-level control.

Two drawbacks of IL methods are: 1) open-loop IL (such as the widely used behavioral cloning approach [Chai et al., 2019, Salzmann et al., 2020, Rhinehart et al., 2019, Gilles et al., 2021, Liang et al., 2020, Ngiam et al., 2021]) suffers from covariate shift [Ross et al., 2011] (which can be addressed with closed-loop training [Ng and Russell, 2000, Ho and Ermon, 2016]), 2) IL methods lack *explicit* knowledge of what constitutes good driving, such as collision avoidance. RL methods have been proposed that allow the policy to learn from explicit reward signals with closed-loop training and have been applied to tasks such as lane-keeping [Kendall et al., 2019], intersection traversal [Isele et al., 2018], and lane changing [Wang et al., 2018]. While these works show the efficacy of RL on specific scenarios, our work analyzes both the large-scale, aggregate performance *and* challenging and safety-critical edge cases that make autonomous driving difficult to deploy in a real-world system.

RL and other closed-loop methods work within a simulated environment. There are a number of such public environments, which vary in how realistic they are, in particular what drives the simulated

	Offline Demo	Closed-loop	Rewards	Example Methods
Behavior Cloning (BC)	Expert Demos	No	No	Multipath, Precog, Trajectron++
Adversarial Imitation/IRL	Expert Demos	Yes	No	IRL, GAIL, MGAIL
RL	No	Yes	Yes	DQN, SAC
Offline RL	Behavioral Data	No	Yes	CQL, TD3+BC
"Imitative" RL	Expert Demos	Yes	Yes	DQfD, DAPG, BC-SAC (ours)

Table 1: A comparison of learning-based approaches to robotic control and autonomous driving.

agents (e.g., expert-following/log playback [Vinitsky et al., 2022, Kothari et al., 2021, Li et al., 2022], intelligent driving model (IDM) [Caesar et al., 2021], or other rule based systems [Dosovitskiy et al., 2017] and ML-based agents [Caesar et al., 2021, Ramamohanarao et al., 2016]), and whether scenarios are procedurally generated (e.g., [Dosovitskiy et al., 2017, Leurent, 2018, Ramamohanarao et al., 2016]) or initialized from real-world driving scenes [Zhan et al., 2019, Li et al., 2022, Vinitsky et al., 2022]. In our experiments, we develop and evaluate in closed-loop on real-world data with other agents following logs.

Combining imitation and reinforcement learning. Methods such as DQfD [Hester et al., 2018], DDPGfD [Vecerik et al., 2017], and DAPG [Rajeswaran et al., 2017] have shown that IL can help RL overcome exploration challenges in domains with sparse rewards. Offline RL approaches, such as TD3+BC [Fujimoto and Gu, 2021] and CQL [Kumar et al., 2020] combine RL objectives with IL ones to regularize Q-learning updates and avoid overestimating out-of-distribution values. The goal of our work is not to propose a novel algorithmic combination of IL and RL, but rather to demonstrate the potential of this general approach to addressing challenges in autonomous driving at scale.

Addressing challenging and safety-critical scenarios for autonomous vehicles. Zhou et al. [2022] learn policies that address rare and long-tail scenarios in autonomous driving by using an ensemble of IL planners combined with model-predictive control. Another approach to improving safety is to augment a learned planner with a non-learned fallback layer that guarantees safety [Shalev-Shwartz et al., 2016, Vitelli et al., 2022]. Our work differs from these approaches, in that we directly incorporate safety awareness into the model learning process through a reward function. Our method is also compatible with a non-learned fallback layer if needed, although we do not explore this direction in this work. Another way to improve robustness of polices is to increase the exposure of policies to negative data during training. Gandhi et al. [2017] collects failure data that covers various ways an unmanned aerial vehicle can crash, and the negative data combined with positive data helps to train robust policies that succeed in challenging environments. Bronstein et al. [2022b] investigates the use of curriculum training to improve performance on challenging edge cases in autonomous driving. While we also increase the exposure of the policy to challenging scenarios during training, we extend the findings of these prior works by showing how RL yields disproportionate improvement on the hardest scenarios.

3 BACKGROUND

3.1 Markov Decision Processes

In this work, we cast the autonomous driving policies learning problem as a Markov decision process (MDP). Following standard formalism, we define an MDP as a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, \rho_0\}$. $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively. $\mathcal{T}$ denotes to transition model. $\mathcal{R}$ represents the reward function, and γ represents the discount factor. ρ_0 represents the initial state distribution. The objective is to find a policy π, a (stochastic) mapping from $\mathcal{S}$ to $\mathcal{A}$, that maximizes the expected discounted sum of rewards,

$$\mathbb{E}_{\mathcal{T},\pi,\rho_0}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t, a_t)\right].$$

3.2 Imitation Learning

IL constructs an optimal policy by mimicking an expert. The typical setup is that we assume an expert (an optimal policy), denoted as π_β, produces a dataset of trajectories $\mathcal{D} = \{s_0, a_0, \cdots, s_N, a_N\}$ through interaction with the environment. The learner then uses an algorithm to train a policy π that imitates the expert. In practice, we only observe the expert states, so we greedily extract

expert actions by minimizing the discrepancy in the state after applying the transition dynamics. For example, behavioral cloning (BC) trains the policy to maximize the log-likelihood of the expert dataset, $\mathbb{E}_{s,a\sim\mathcal{D}}\left[\log \pi(a|s)\right]$. Alternatively, closed loop imitation approaches include inverse RL (IRL) [Ng and Russell, 2000] and adversarial imitation learning (GAIL [Ho and Ermon, 2016], MGAIL [Baram et al., 2016]), which instead aim to more directly match the occupancy measure or state-action visitation distribution between the policy and the expert, rather than indirectly through the conditional action distribution. In principle, this can resolve the covariate shift issue that affects open loop imitation [Ross et al., 2011].

3.3 Reinforcement Learning

RL aims to learn an optimal policy through an iterative, online trial and error process. In this work we use off-policy, value-based RL algorithms such as Q-learning. These methods aim to learn the state-action value function, defined as the expected future return when starting from a particular state and action:

$$Q^{\pi}(s,a) = \mathbb{E}_{\mathcal{T},\pi,\rho_0}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t)|s_0=s,a_0=a\right].$$

In this work, we use an actor-critic method for training continuous control policies. Typical actor-critic methods alternate between training a critic Q to minimize the Bellman error and an actor π to maximize the value function. We use the entropy-regularized updates of Soft Actor-Critic (SAC) [Haarnoja et al., 2018]:

$$\min_{Q}\ \mathbb{E}_{s,a,s'\sim\pi}[(Q(s,a)-\hat{Q}(s,a,s'))^2], \tag{1}$$

where

$$\hat{Q}(s,a,s') = r(s,a) + \gamma\mathbb{E}_{a'\sim\pi}\left[\bar{Q}(s',a') - \log\pi(a'|s')\right] \tag{2}$$

$$\max_{\pi}\ \mathbb{E}_{s,a\sim\pi}\left[Q(s,a)+\mathcal{H}(\pi(\cdot|s))\right], \tag{3}$$

where $\bar{Q}$ denotes a target network that is a copy of the critic through which gradients do not pass.

4 Learning to Drive with RL-Augmented BC

We wish to design an approach that benefits from the complementary strengths of IL and RL. Imitation provides an abundant source of learning signal without the need for reward design, and RL addresses the weaknesses of IL in rare and challenging scenarios where data is scarce. Following this intuition, we formulate an objective that utilizes the learning signal from demonstrations where data is abundant and the reward signal where data is scarce. Specifically, we utilize a weighted mixture of the IL and RL objectives:

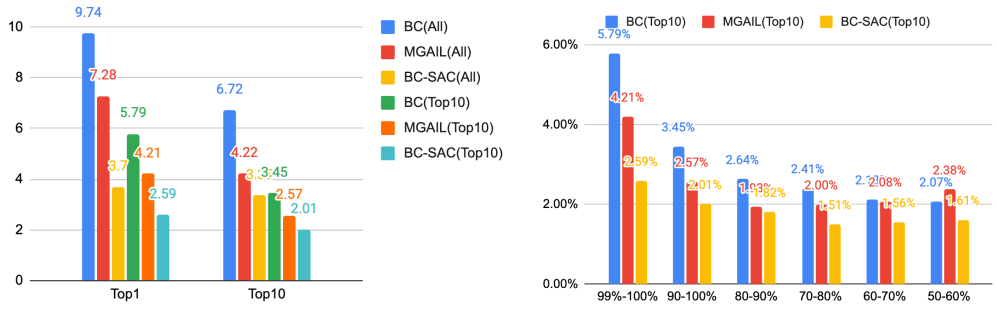
$$\max_{\pi}\ \mathbb{E}_{\mathcal{T},\pi,\rho_0}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t)\right] + \lambda\mathbb{E}_{s,a\sim\mathcal{D}}[\log\pi(a|s)]. \tag{4}$$

4.1 Behavior Cloned Soft Actor-Critic (BC-SAC)

While in principle a variety of RL methods could be combined with IL to optimize Eq. (4), a convenient choice for efficient training is to use actor-critic algorithms, in which case the policy can be optimized with respect to Eq. (4) simply by adding the imitation learning objective to the expected value of the Q-function (i.e., the critic), similarly to DAPG [Rajeswaran et al., 2018] or TD3+BC [Fujimoto and Gu, 2021]. Building on the widely used SAC framework, which further adds an entropy regularization objective to the actor, we obtain our full actor objective:

$$\mathbb{E}_{s,a\sim\pi}[Q(s,a)+\mathcal{H}(\pi(\cdot|s))] + \lambda\mathbb{E}_{s,a\sim\mathcal{D}}[\log\pi(a|s)]. \tag{5}$$

The critic update remains the same as in SAC, outlined in Eq 1. With the appropriate setting of λ, this objective encourages the policy to mimic the expert data when it is within the data distribution $\mathcal{D}$. However, in regions of the state space visited by the policy that are outside of the dataset, the RL term is active, and the policy primarily relies on reward to learn. Fig. 1b visualizes this concept.



(a) (b)

Figure 2: **(a)** Failure rates on the most challenging evaluation sets: Top1 and Top10 (lower is better, with training on All and Top10). BC-SAC consistently achieves the lowest error rates. **(b)** Failure rates of BC, MGAIL, and BC-SAC across scenarios of varying difficulty levels (50%-100%, lower is better). While all methods perform worse as the evaluation dataset becomes more challenging, BC-SAC always performs best and shows the least degradation.

4.2 Reward Function

While designing a reward function to capture "good" driving behavior is an open-challenge [Knox et al., 2021], we can side-step this issue by relying on the imitation learning loss to primarily guide the policy, while the simple reward function only needs to encode safety constraints. To this end, we use a combination of collision and off-road distances as our reward signal. The collision reward is

$$R_{\text{collision}} = \min(d_{\text{collision}} - 1.0, 0),$$

where $d_{\text{collision}}$ is the Euclidean distance in meters of the closest points between the ego vehicle and a nearest bounding box of other vehicles. This reward encourages the vehicle to keep a certain distance from nearby objects. The off-road reward is

$$R_{\text{off-road}} = \text{clip}(-1.0 - d_{\text{to-edge}}, 0.0, 2.0),$$

where $d_{\text{to-edge}}$ is the distance in meters of the vehicle to the road edge (negative being on-road, positive being off-road). This reward encourages the vehicle to keep a negative distance to road edge (off-road status). We combine the rewards additively, such that

$$R = R_{\text{collision}} + R_{\text{off-road}}.$$

4.3 Forward and Inverse Vehicle Dynamics Models

We update the vehicle's state using the kinematic bicycle dynamics model [Rajamani, 2011], which computes the vehicle's next (x, y) position and heading given a 2-dimensional input steering and acceleration action $a = (a_{\text{steer}}, a_{\text{accel}})$. In order to obtain expert actions for imitation learning, we use an inverse dynamics model to solve for the actions that would have achieved the same states as the logged trajectories in our dataset. These expert actions are found using trajectory optimization by minimizing the mean squared errors of the corners' (x, y) positions between the inferred state $\mathcal{T}(s_t, a_t)$ and ground-truth next state s_{t+1}:

$$a^*_{1:T} = \underset{a_{1:T}}{\text{argmin}} \sum_{t=0}^{T} ||s_{t+1} - \mathcal{T}(s_t, a_t)||^2.$$

4.4 Model Architecture

We use a dual actor-critic architecture similar to TD3 and SAC [Fujimoto et al., 2018, Haarnoja et al., 2018]: the main components are an actor network $\pi(a|s)$, a double Q-critic network $Q(s, a)$ and a target double Q-critic network $\bar{Q}(s, a)$. Each network has a separate Transformer observation encoder described in Bronstein et al. [2022a] that encodes features including all vehicle states, road-graph points, traffic lights signals, and route goals. The actor network outputs a $\tanh$-squashed diagonal Gaussian distribution parameterized by a mean μ and variance σ. Training architecture and hyper-parameters can be found in Appendix A.1 and A.2 respectively.

Method	Training	Top1 (%)	Top10 (%)	Top50 (%)	All (%)	Route Progress Ratio, All(%)
BC	All	9.74±0.49	6.72±0.47	5.14±0.39	4.35±0.27	99.00±0.39
MGAIL	All	7.28±0.98	4.22±0.77	3.40±0.97	**2.48±0.29**	**99.55±1.91**
SAC	All	5.29±0.66	4.64±1.08	4.12±0.74	6.66±0.44	77.82±8.21
BC-SAC	All	**3.72±0.62**	**2.88±0.23**	**2.64±0.21**	3.35±0.31	95.26±8.64
BC	Top10	5.79±0.82	3.45±0.72	2.71±0.57	3.64±0.31	**98.06±0.18**
MGAIL	Top10	4.21±0.95	2.57±0.52	2.20±0.52	**2.45±0.35**	96.57±1.19
SAC	Top10	4.33±0.47	4.11±0.63	3.66±0.47	5.60±0.86	71.05±2.47
BC-SAC	Top10	**2.59±0.31**	**2.01±0.29**	**1.76±0.20**	2.81±0.26	87.63±0.58
BC	Top1	7.66±1.13	7.84±0.92	6.63±0.78	6.85±0.65	**94.10±1.00**
MGAIL	Top1	4.24±0.95	3.16±0.43	2.74±0.46	3.79±0.46	93.10±11.72
SAC	Top1	4.15±0.31	3.87±0.12	3.46±0.16	5.98±1.03	75.63±2.19
BC-SAC	Top1	**3.61±0.87**	**2.96±1.11**	**2.69±0.87**	**3.38±0.48**	75.00±17.21

Table 2: Failure rates (lower is better) and progress ratios (higher is better) of BC-SAC and baselines on different training/evaluation subsets.

4.5 Training on Difficult Examples

The performance of learning-based methods strongly depends on the distribution of the training data. This is a particularly important factor in settings with a long-tail distribution of safety-critical examples [Frank et al., 2008, Kalra and Paddock, 2016, Shalev-Shwartz et al., 2016, Paul et al., 2018, 2019]). Autonomous driving falls in this category: most examples could be easily navigated by a variety of policies, but a small minority contains challenging scenarios with potentially adverse safety outcomes. As demonstrated in Bronstein et al. [2022b], training on more difficult examples results in better performance than using all the available data, both in aggregate and in challenging scenarios. We explore how the composition of the dataset affects the performance of our method and the baselines in the experiments.

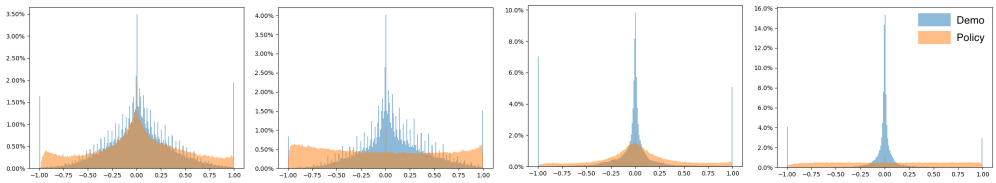
5 EXPERIMENTS

5.1 Experimental Setup

Datasets. We use a dataset (denoted **All**) consisting of over 100k hours of expert driving trajectories, split into 10 second segments, collected from a fleet of vehicles operating in San Francisco (SF) [Bronstein et al., 2022b]. We divide these segments into 6.4 million for training and 10k for testing. Trajectories from the same vehicle operating on the same day are stored in the same partition to avoid train-test leakage. The trajectories, which are sampled at 15 Hz, contain features describing the AV state and the state of the environment as measured by the AV's perception system. Using the *difficulty model* described in Bronstein et al. [2022b] to rank segments, we also filter the dataset to contain more challenging scenarios in order to 1) evaluate our method's performance on the long-tail of driving scenarios, and 2) train policy variants on more difficult examples. The difficulty model is trained on an additional 14k hours of expert trajectories in the same manner as the All dataset [Bronstein et al., 2022b]. This model predicts whether a segment will result in a collision or near-miss when re-simulated with an internal AV planner, as determined by human labelers. We create the **Top1**, **Top10**, and **Top50** subsets by selecting the top 1% (40k train, 1.2k test), 10% (400k train, 19k test), and 50% (2 million train, 66k test) of segments with the highest difficulty model scores, respectively.

Simulation. As mentioned in Sec. 4.3, vehicle dynamics are modeled using a 2D bicycle dynamics model. The behavior of other vehicles and pedestrians in the scene are replayed from the logs (log-playback), similarly to Vinitsky et al. [2022], Kothari et al. [2021], Li et al. [2022]. While this means that agents are non-reactive, it ensures that the behavior of other agents is human-like, and the inclusion of imitative losses discourages the learned policy to deviate too far from the logs, which would cause the log-playback agents to become unrealistic. We also use short segments of 10s to mitigate pose divergence.

Baselines. As a representative open loop imitation method, we use behavioral cloning (*BC*) [Pomerleau, 1988] and as a representative closed loop imitation method, we use *MGAIL* [Baram et al., 2016, Bronstein et al., 2022a]. The latter takes advantage of closed loop training and the differentiability of the simulator dynamics. For completeness, we also include a SAC baseline to represent RL-only approaches.



(a) BC-SAC acceleration (b) SAC acceleration (c) BC-SAC tire angles (d) SAC tire angles

Figure 4: Marginal action distributions of SAC/BC-SAC (orange) vs logs (blue).

Metrics. We evaluate agents using two metrics:

1. *Failure Rate*: Percentage of the run segments that have at least one *Collision* or *Off-road* event. *Collision* is true if the bounding box of the ego vehicle intersects with a bounding box of another object. *Off-road* is true if the bounding box of the ego vehicle deviates from the drivable surface according to the map.
2. *Route Progress Ratio*: Ratio of the distance traveled along the route by the policy compared to the expert demonstration.

The first two are computed per timestep and any instance of a collision or off-road event in a segment results in a failure for the entire segment.

5.2 Results

We systematically evaluated the baseline methods (BC, MGAIL, SAC) and BC-SAC when trained on various subsets of the training dataset (All, Top10, and Top1) and evaluated against subsets of the evaluation set (Top1, Top10, Top50, All) in Table 2. All configurations were evaluated with three random seeds, which were used to report mean and standard deviation. Previously, Bronstein et al. [2022b] showed that training MGAIL on Top10 yields similar performance with training on All. Similarly, we find that all methods perform best when trained on Top10. Notably, BC trained on Top1 perform significantly worse compared to training on All or Top10, which reflects the fact that imitation learning methods rely on large amounts of data to implicitly infer driving preferences. In particular, open loop BC tends to fall victim to distribution shifts when not provided with enough data and performs especially poorly. On the other hand, BC-SAC performs robustly when trained on Top1, demonstrating its reliability and robustness. Given that all methods perform best when trained on Top10, we focus on that setting in the following sections.

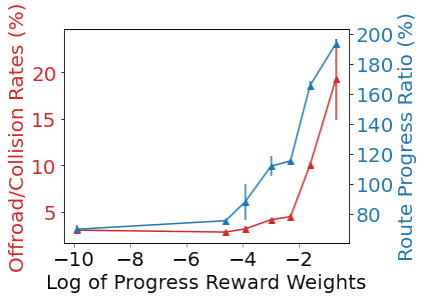


Figure 5: Progress reward weights (log-scale) vs policy evaluation performance (Failure rates and route progress ratios).

How does BC-SAC perform compared to imitation methods (BC, MGAIL) in the challenging scenarios? In Figure 2, we compare BC-SAC against BC and MGAIL across slices of the evaluation dataset according to difficulty levels. We find that BC-SAC achieves better performance overall, especially in the more challenging slices where the performance of both BC and MGAIL substantially degrade. Interestingly, MGAIL's performance degenerates on less difficult data, possibly due to overfitting. Finally, we note that BC-SAC achieves the lowest variance (across scenarios of varying difficulty) in performance ($\sigma = 0.37$) compared to BC ($\sigma = 1.29$) and MGAIL ($\sigma = 0.78$).

How does BC-SAC compare to RL-only training (SAC)? In all training and evaluation configurations, BC-SAC outperforms SAC in terms of safety metrics (Table 2). This could be due to the fact that BC-SAC also leverages the learning signal from large amount of driving demonstrations. In Figure 4, we see that SAC generates actions that deviate significantly from the demonstrations with more boundary action values yielding unnatural (e.g., more swerves) and uncomfortable (e.g., abrupt acceleration/deceleration) driving behavior. With a BC loss, BC-SAC generates an action distribution similar to the logs.

Progress-safety balance. While this work focuses on safety-critical scenarios, we show that introducing a small amount of a progress reward leads to significantly more progress without major regressions in safety metrics. However, large progress rewards lead to degradation in performance as shown in Figure 5.

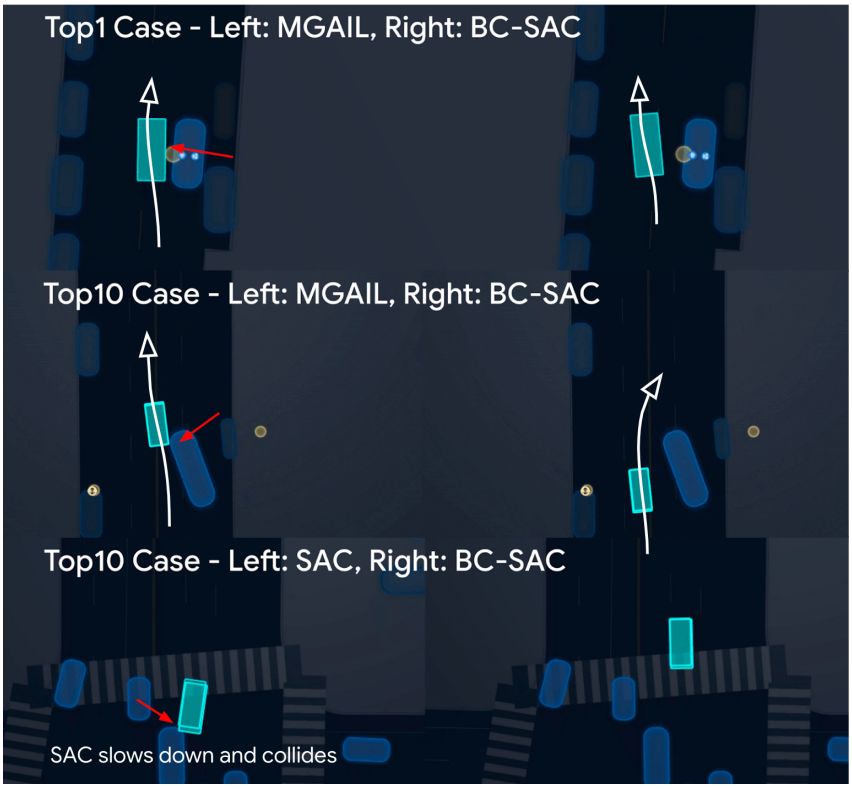


Figure 6: Visualizations of win cases against baseline agents. The cyan car is controlled. **Example 1**: MGAIL collides with a pedestrian coming out of a double parked car while BC-SAC was able to leave an appropriately wide-clearance. **Example 2**: MGAIL does not provide sufficient clearance and collides with the incoming vehicle. **Example 3**: SAC slows down in an intersection resulting in an rear collision. In contrast, BC-SAC keeps an appropriate speed profile through the intersection without a collision.

6 CONCLUSIONS

We showed how augmenting imitation learning with RL and a simple safety reward can significantly improve safety and reliability in challenging scenarios. Our experiments show, both quantitatively and qualitatively, that when training on challenging datasets, the proposed method performs more robustly than IL-only and RL-only methods across driving scenarios with varying difficulty levels, leading to especially large improvements on the hardest slices of the evaluation set. While this work mainly focused on optimizing safety-related rewards, a natural extension is to incorporate other factors into the objective, such as progress, traffic rule adherence, and passenger comfort. Besides the reward function, our approach does not account for unexpected behavior of other agents in response to out-of-distribution actions on the part of the ego vehicle, and it still requires heuristically choosing the tradeoff between the IL and RL objectives. A promising direction for future work would be to extend the approach to enforce safety as an explicit constraint, perhaps in combination with methodology to mitigate distributional shift. Such methods might provide effective safety constraints that combine precise and optimal RL-based behavior with the naturalisitic, human-like driving patterns inherited from imitation.

References

Nir Baram, Oron Anschel, and Shie Mannor. Model-based adversarial imitation learning. *arXiv preprint arXiv:1612.02179*, 2016.

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.

Eli Bronstein, Mark Palatucci, Dominik Notz, Brandyn White, Alex Kuefler, Yiren Lu, Supratik Paul, Payam Nikdel, Paul Mougin, Hongge Chen, et al. Hierarchical model-based imitation learning for planning in autonomous driving. *arXiv preprint arXiv:2210.09539*, 2022a.

Eli Bronstein, Sirish Srinivasan, Supratik Paul, Aman Sinha, Matthew O'Kelly, Payam Nikdel, and Shimon Whiteson. Embedding synthetic off-policy experience for autonomous driving via zero-shot curricula. In *6th Annual Conference on Robot Learning*, 2022b. URL https://openreview.net/forum?id=cF1dxVGxic-.

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.

Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019.

Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

Pim De Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pages 1407–1416. PMLR, 2018.

Jordan Frank, Shie Mannor, and Doina Precup. Reinforcement learning in the presence of rare events. In *Proceedings of the 25th international conference on Machine learning*, pages 336–343, 2008.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

Dhiraj Gandhi, Lerrel Pinto, and Abhinav Gupta. Learning to fly by crashing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3948–3955. IEEE, 2017.

Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Gohome: Graph-oriented heatmap output for future motion estimation. *arXiv preprint arXiv:2109.01827*, 2021.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018. URL http://arxiv.org/abs/1801.01290.

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.

David Isele, Reza Rahimi, Akansel Cosgun, Kaushik Subramanian, and Kikuo Fujimura. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2034–2039. IEEE, 2018.

Nidhi Kalra and Susan M. Paddock. *Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?* RAND Corporation, 2016.

Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8248–8254. IEEE, 2019.

W Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. Reward (mis) design for autonomous driving. *arXiv preprint arXiv:2104.13906*, 2021.

Parth Kothari, Christian Perone, Luca Bergamini, Alexandre Alahi, and Peter Ondruska. Drivergym: Democratising reinforcement learning for autonomous driving. *arXiv preprint arXiv:2111.06889*, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Edouard Leurent. An environment for autonomous driving decision-making. `https://github.com/eleurent/highway-env`, 2018.

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*, 2022.

Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. *arXiv preprint arXiv:2007.13732*, 2020.

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)*, pages 584–599, 2018.

Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of 17th International Conference on Machine Learning, 2000*, pages 663–670, 2000.

Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, David Weiss, Benjamin Sapp, Zhifeng Chen, and Jonathon Shlens. Scene transformer: A unified multi-task model for behavior prediction and planning. *CoRR*, abs/2106.08417, 2021. URL `https://arxiv.org/abs/2106.08417`.

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

Supratik Paul, Konstantinos Chatzilygeroudis, Kamil Ciosek, Jean-Baptiste Mouret, Michael Osborne, and Shimon Whiteson. Alternating optimisation and quadrature for robust control. In *AAAI Conference on Artificial Intelligence*, 2018.

Supratik Paul, Michael A. Osborne, and Shimon Whiteson. Fingerprint policy optimisation for robust reinforcement learning. In *International Conference on Machine Learning*, 2019.

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

Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Robotics: Science and Systems*, 2018.

Kotagiri Ramamohanarao, Hairuo Xie, Lars Kulik, Shanika Karunasekera, Egemen Tanin, Rui Zhang, and Eman Bin Khunayn. Smarts: Scalable microscopic adaptive road traffic simulator. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):1–22, 2016.

Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. Deep imitative models for flexible inference, planning, and control. *arXiv preprint arXiv:1810.06544*, 2018.

Nicholas Rhinehart, Rowan McAllister, Kris M. Kitani, and Sergey Levine. PRECOG: prediction conditioned on goals in visual multi-agent settings. *CoRR*, abs/1905.01296, 2019. URL `http://arxiv.org/abs/1905.01296`.

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

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. *arXiv preprint arXiv:2001.03093*, 2020.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

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. *arXiv preprint arXiv:2206.09889*, 2022.

Matt Vitelli, Yan Chang, Yawei Ye, Ana Ferreira, Maciej Wołczyk, Błażej Osiński, Moritz Niendorf, Hugo Grimmett, Qiangui Huang, Ashesh Jain, et al. Safetynet: Safe planning for real-world self-driving vehicles using machine-learned policies. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 897–904. IEEE, 2022.

Pin Wang, Ching-Yao Chan, and Arnaud de La Fortelle. A reinforcement learning based approach for automated lane change maneuvers. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1379–1384. IEEE, 2018.

Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*, 2019.

Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15222–15232, October 2021a.

Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15222–15232, 2021b.

Weitao Zhou, Zhong Cao, Nanshan Deng, Xiaoyu Liu, Kun Jiang, and Diange Yang. Long-tail prediction uncertainty aware trajectory planning for self-driving vehicles. *arXiv preprint arXiv:2207.00788*, 2022.

A Appendix

A.1 IL + RL Distributed Actor-Learner Training Architecture

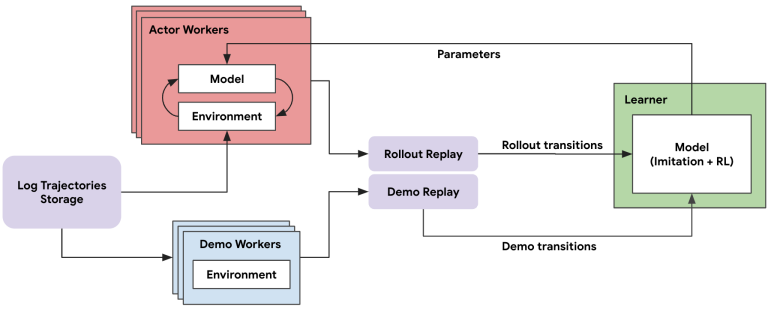


Figure 7: IL + RL distributed actor-learner training architecture. We extend the distributed IMPALA architecture [Espeholt et al., 2018] with additional demo rollout workers and a demo replay buffer, which produce rollout transitions in the same format as the actor workers. The learner worker samples from both the rollout replay buffer and the demo replay buffer to perform training updates in an off-policy manner.

A.2 Additional Details on Model Architectures and Hyper-parameters Settings

We use a dual actor-critic architecture similar to TD3 and SAC [Fujimoto et al., 2018, Haarnoja et al., 2018]: each of the main components, actor network $\pi(a|s)$, double Q-critic network $Q(s,a)$ and target double Q-critic network $\bar{Q}(s,a)$, has a separate Transformer observation encoder described in Bronstein et al. [2022a], and the encoder embedding is fed to a $(256, 256)$ fully connected head. The actor network outputs a $\tanh$-squashed diagonal Gaussian distribution parameterized by a mean μ and variance σ.

We train the BC-SAC algorithm with the following hyper-parameters: the actor learning rate is 1e-4, the critic learning rate is 1e-4, the imitation learning rate is 5e-5, the batch size is 64, and the reward discount ratio is 0.92. The sample-to-insert ratio for replay is 8, which is the average number of times the learner should sample each item in the replay buffer during the item's entire lifetime. In practice, instead of performing a combined gradient step of both the IL and RL objectives, we alternate the training steps between IL and RL with different update frequencies. For every 8 RL updates, we update with IL loss for one time.

For SAC, we use the same network design and hyper-parameters as in BC-SAC, except that it does not perform IL step.

For BC, we discretize the 2d action space (steer, acceleration) in to $31 \times 7 = 217$ actions with the same underlying dynamics model. We use a similar network design for BC as in BC-SAC's actor network with a Softmax prediction head representing probabilities of the discrete actions. We use the cross-entropy loss with a learning rate of 1e-4 and batch size of 256 for training.

For MGAIL, we follow the network design and hyper-parameters setting presented in Bronstein et al. [2022a].