# HG-DAgger: Interactive Imitation Learning with Human Experts

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J. Kochenderfer

***Abstract*— Imitation learning has proven to be useful for many real-world problems, but approaches such as behavioral cloning suffer from data mismatch and compounding error issues. One attempt to address these limitations is the DAgger algorithm, which uses the state distribution induced by the novice to sample corrective actions from the expert. Such sampling schemes, however, require the expert to provide action labels without being fully in control of the system. This can decrease safety and, when using humans as experts, is likely to degrade the quality of the collected labels due to perceived actuator lag. In this work, we propose HG-DAgger, a variant of DAgger that is more suitable for interactive imitation learning from human experts in real-world systems. In addition to training a novice policy, HG-DAgger also learns a safety threshold for a model-uncertainty-based risk metric that can be used to predict the performance of the fully trained novice in different regions of the state space. We evaluate our method on both a simulated and real-world autonomous driving task, and demonstrate improved performance over both DAgger and behavioral cloning.**

## I. Introduction

Imitation learning is often posed as a supervised learning problem. Data is gathered from an expert policy and is used to train a novice policy [1], [2]. While this approach, known as behavioral cloning, can be an effective way to learn policies in scenarios where there is sufficiently broad data coverage, in practice it often suffers from data mismatch and compounding errors [3].

Online sampling frameworks such as the DAgger algorithm have been proposed to address the drawbacks inherent in naive behavioral cloning [3]. DAgger trains a sequence of novice policies using corrective action labels provided by the expert at states sampled by a mixture of the expert and the novice policies. At each time-step in a data-gathering rollout, a gating function determines which of the two policies' choice of action will actually be executed on the combined system; in the case of DAgger, this gating function amounts to a weighted coin toss that executes the expert's choice of action with some probability $\beta \in [0,1]$ and the novice's choice of action with probability $1-\beta$.

By allowing the novice to influence the sampling distribution used to acquire expert action labels (a practice known as "Robot-Centric" (RC) sampling), DAgger trains a more robust policy that is capable of handling perturbations from

This material is based upon work supported by SAIC Innovation Center, a subsidiary of SAIC Motors and by AFRL and DARPA under contract FA8750-18-C-0099.

M. Kelly is with the Computer Science Department, Stanford University, Stanford, CA, USA (e-mail: mkelly2@stanford.edu). C. Sidrane and M.J. Kochenderfer are with the Aeronautics and Astronautics Department, Stanford University, Stanford, CA, USA (e-mail: {csidrane,mykel}@stanford.edu). K. Driggs-Campbell is with the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign, Urbana, IL (e-mail: krdc@illinois.edu).

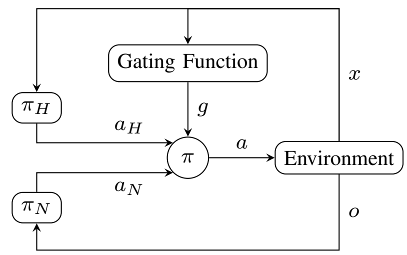


Fig. 1: Control loop for HG-DAgger

the nominal trajectories of the expert [4]. While DAgger has many appealing properties, including theoretical performance guarantees, its use of RC sampling can compromise training-time safety because it involves performing data-gathering rollouts under the partial or complete control of an incompletely trained novice. Furthermore, the shared control scheme can alter the behavior of human experts, degrading the quality of the sampled action labels and potentially even destabilizing the combined system.

The central problem with RC sampling methods is that they may not provide the human expert with sufficient control authority during the sampling process [4]. Humans rely on feedback from the system under control to perform optimally; however, many RC sampling based methods (for example, DAgger with $\beta = 0$) provide the human expert with no feedback signal at all, since the human's choice of action is never actually executed.

RC sampling methods that do allow the expert to influence the sampling process, such as DAgger with nonzero $\beta$, switch control authority back and forth between the novice and the expert during training, which can degrade the feedback signal to the user and cause a perceived actuator lag. Humans tend to be sensitive to small changes in execution, like time delay [5], [6]. These issues can drive the human expert to adapt and change their behaviors over time [7]. In imitation learning, this can degrade learning stability and may ultimately cause the novice to learn behaviors that are significantly different from the unbiased expert behavior [6].

Furthermore, even for two policies that are individually stable, there may exist some switching procedures that destabilize the combined system [8]. For example, in the aeronautical domain, pilot-induced oscillations can arise from a series of (over) corrections from a pilot attempting to stabilize an aircraft [9]. When shared control systems directly affect the pilot's inputs, they are often aware that their authority over the control of the system is reduced. This awareness often leads to a greater response (or an overcorrection) from the pilot, in turn causing conflicting efforts from the control system and the human [10].

In the case of the DAGGER algorithm, tuning the $\beta$ parameter schedule may mitigate these issues. A larger $\beta$-value that is reduced more slowly over training epochs will provide greater control authority to the human expert, which should allow for the collection of higher-quality action labels and improved safety during training, since the potentially unsafe novice has a lesser degree of control. However, increasing $\beta$ too much will likely degrade test-time performance, particularly in the extreme case of $\beta = 1$, where DAGGER reduces to behavioral cloning and is subject to compounding error issues. It is therefore not clear how or if a good choice of $\beta$ for an arbitrary imitation learning task can be made *a priori*, and tuning $\beta$ on a physical system is likely to be challenging due both to the expense of collecting samples from real-world systems and due to $\beta$'s effect on training-time safety [6]. Furthermore, it is not necessarily clear that there exists a $\beta$ that provides a satisfactory trade-off between learning performance and training-time safety.

Recent prior work using human experts has not fully addressed these issues, and has focused instead primarily on reducing the work load of the expert by minimizing the number of times the expert is queried [11]–[13]. Instead of querying at every time step, the expert only gives corrective actions when there is a significant discrepancy between the anticipated behaviors of the novice and the expert. These approaches effectively reduce the number of demonstrations and expert queries required while preserving strong theoretical guarantees on performance and often improving safety.

In this work, we consider a probabilistic extension of DAGGER called ENSEMBLEDAGGER. This method takes a Bayesian approach to determine whether it is safe for the novice to act by approximating risk with familiarity [14]. To do this, the novice is represented as an ensemble of neural networks; such ensembles can efficiently approximate Gaussian processes, which scale poorly with the amount of training data. Using the mean and variance of the policy's actions, the *confidence* of the novice can be assessed and used to determine whether or not the expert should intervene. In short, this method aims to maximize the novice's share of actions during data-gathering rollouts, while constraining the probability of failure in a model-free manner. However, identifying measures of risk and appropriate thresholds on the variance remains an open problem in both model-free and model-based communities [14]–[16].

We expand upon the concepts proposed by ENSEMBLEDAGGER and develop a methodology for learning from humans in scenarios where running a novice policy in parallel with a human policy is not intuitive for the user. We present the following contributions:

1) We propose Human-Gated DAGGER (HG-DAGGER), a DAGGER-variant designed for more effective imitation learning from human experts.
2) We propose a data-driven approach for learning a safety threshold for our accompanying risk metric and show that the metric produced is meaningful.
3) We demonstrate the efficacy of our method on an autonomous driving task, showing improved sample efficiency, greater training stability, and more human-like behavior relative to DAGGER and behavioral cloning.

This paper is organized as follows. Section II presents the methodology for our proposed framework, HG-DAGGER. The experimental setup for collecting human data and training novice policies is presented in Section III. The performance of the resulting policies is analyzed in Section IV. Section V discusses our findings and outlines future work.

## II. METHODS

We are motivated by the assumption that higher quality action labels can be acquired when the human expert is given stretches of uninterrupted control authority. Other methods hand off control stochastically at each time-step, or require that the human expert retroactively provide corrective action labels to states visited by the novice. Instead, we allow the human expert to take control when they deem it necessary and to maintain exclusive control authority until they manually hand control back to the novice policy.

Using nomenclature borrowed from the hierarchical RL literature, we refer to this human supervisor as a gating function, which decides whether the expert or novice "sub-policy" should be in control at a given moment. For this reason, we call our algorithm Human-Gated DAGGER (HG-DAGGER). Fig. 1 illustrates this control scheme.

Like other DAGGER variants, HG-DAGGER trains a sequence of novice policies on a training data set $\mathcal{D}$, which is iteratively augmented with additional expert labels collected during repeated data-gathering rollouts of a combined expert-novice system. Unlike other DAGGER variants, however, the gating function employed by HG-DAGGER is controlled directly by the expert. In HG-DAGGER, the novice policy is rolled out until the expert observes that the novice has entered an unsafe region of the state space. The expert takes control and guides the system back to a safe and stable region of the state space. Expert action labels are only collected and added to $\mathcal{D}$ during these recovery trajectories, during which the human expert has uninterrupted control of the system. Once in a safe region, control is returned to the novice.

We refer to the set of "safe" states (as judged by the human expert) as the *permitted set* $x_t \in \mathcal{P}$ and formalize the human-controlled gating function accordingly as $g(x_t) = \mathbb{1}[x_t \notin \mathcal{P}]$ Given a human expert $\pi_H$ and the current instantiation of the novice $\pi_{N_i}$, we can then express the HG-DAGGER rollout policy for the $i$th training epoch, $\pi_i$, as:

$$\pi_i(x_t) = g(x_t)\pi_H(x_t) + (1 - g(x_t))\pi_{N_i}(o_t) \quad (1)$$

where $o_t$ is the observation received by the novice at the current state $x_t$, generated by the observation function $\mathcal{O}(\cdot)$. A human expert has access to many channels of information that are unavailable to the novice policy; we acknowledge this by explicitly denoting that the expert has access to the full state, $x_t$, but that the novice only has access to an observation, $o_t = \mathcal{O}(x_t)$. Letting $\xi_i$ represent the concatenation of all rollouts performed with $\pi_i$, we can represent the data collected in the $i$th training epoch as

$$\mathcal{D}_i = \{(\mathcal{O}(x_t), \pi_H(x_t)) \mid g(x_t) = 1, x_t \in \xi_i\} \quad (2)$$

At the end of epoch $i$, $\mathcal{D}_i$ is added to the training data set $\mathcal{D}$ and the next novice policy is trained on the aggregated data. The dataset is initialized with a set of samples $\mathcal{D}_{BC}$ gathered using behavioral cloning.

In theory, the data collected with HG-DAGGER is used to teach the novice to stabilize itself about the nominal expert trajectories demonstrated by the expert during the initial behavioral cloning step. Since both behavioral cloning and HG-DAGGER collect data from the human expert only while it has uninterrupted control, we can expect to acquire high-quality action labels both along nominal expert trajectories and along recovery trajectories using this method.

At training time, HG-DAGGER relies on the human expert to ensure safety by intervening in dangerous situations.[1] At test time, the policy trained with these additional demonstrations is allowed to act without any human intervention. Meanwhile, HG-DAGGER also learns a risk metric derived from the novice policy's "doubt" that can be used to understand and assess the performance of the final trained policy.

We use a risk approximation method inspired by [14] and represent the novice as an ensemble of neural networks. The covariance matrix, $C_t$, of the ensemble's outputs given input $o_t$ contains useful measures of policy confidence. We use the $\ell_2$-norm of the main diagonal of $C_t$ as the doubt, $d_N(o_t)$.

$$d_N(o_t) = \|\text{diag}(C_t)\|_2 \tag{3}$$

Following the assumption that a neural network ensemble approximates a Gaussian process [17], we expect that doubt will be high in regions of the state space that are poorly represented in the data set. The novice is expected to perform poorly in these regions both because they are inadequately sampled and because they are likely to be more intrinsically risky, as the expert is likely to bias sampling away from more intrinsically risky regions when possible. ENSEMBLEDAGGER uses this risk heuristic to improve training-time safety by designing a gating function that only permits the novice to act when its doubt falls below some threshold, but the authors do not provide a means of selecting that threshold.

Rather than attempting to select a "safe" threshold value for doubt *a priori*, we instead learn a threshold, $\tau$, from human data. We record the novice's doubt at the point of human intervention in a doubt intervention logfile, $\mathcal{I}$. We compute $\tau$ as the mean of the final 25% of the entries in $\mathcal{I}$:

$$\tau = \frac{1}{\text{len}(\mathcal{I})/4} \sum_{i=\lfloor .75N \rfloor}^{N} (\mathcal{I}[i]) \tag{4}$$

We chose this to balance learning $\tau$ from a larger number of human interventions and learning $\tau$ from only the most relevant human interventions. The most relevant interventions are those made during rollouts of novice policies trained on more data. These rollouts are made with a policy which more closely resemble the fully trained policy.

The current work uses this threshold to evaluate and understand the fully-trained policy's performance, but we propose

[1] As a result, HG-DAGGER is not suitable for application in those real-world domains where the human expert cannot quickly identify and react to unsafe situations.

**Algorithm 1** HG-DAGGER

**procedure** HG-DAGGER($\pi_H, \pi_{N_1}, \mathcal{D}_{BC}$)
  $\mathcal{D} \leftarrow \mathcal{D}_{BC}$
  $\mathcal{I} \leftarrow []$
  **for** epoch $i = 1 : K$
    **for** rollout $j = 1 : M$
      **for** timestep $t \in T$ of rollout j
        **if** expert has control
          record expert labels into $\mathcal{D}_j$
        **if** expert is taking control
          record doubt into $I_j$
      $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_j$
      append $\mathcal{I}_j$ to $\mathcal{I}$
    train $\pi_{N_{i+1}}$ on $\mathcal{D}$
  $\tau \leftarrow f(\mathcal{I})$
  **return** $\pi_{N_{K+1}}, \tau$

in future work to use this risk metric at test time to determine when control of the system should be returned from the trained policy to a safer, more conservative controller. See Algorithm 1 for a summary of the HG-DAGGER algorithm.

The approach most similar to ours is the Confidence-Based Autonomy algorithm proposed in [18]. Like HG-DAGGER, this algorithm allows the expert to provide corrective demonstrations to the novice whenever the expert deems it necessary. However, the method also allows the novice to request demonstrations from the expert when a measure of its confidence falls below some threshold, allowing for a switching behavior similar to that seen in DAGGER, which we have argued is problematic in the human-in-the-loop setting. Our approach also differs from [18] in that we use an ensemble of neural network learners instead of Gaussian mixture models. This distinction is significant because having a sufficiently expressive learner is important if we are to adequately mimic expert behavior on complex tasks [4], [19]. Another important difference between the two works is that HG-DAGGER learns the doubt threshold from the expert (via training-time interventions) rather than calculating it in an ad-hoc manner (i.e. by setting it as three times the average nearest neighbor distance in the dataset).

## III. EXPERIMENTAL SETUP

We apply our method to a complex, real-world task: learning autonomous driving policies from human drivers. Early autonomous driving researchers in the 1980s applied behavioral cloning to the lane keeping task [20]. More recently, end-to-end learning approaches have made use of both behavioral cloning and DAGGER frameworks [21], [22]. This work specifically targets human-in-the-loop learning, which, as previously discussed, is typically a difficult problem.

In this section, we present the experimental setup used to collect data and train our policy both in simulation and on a physical test vehicle. We compare the performance of HG-DAGGER to that of DAGGER and behavioral cloning.

## A. Experimental Task

The driving task involves an ego vehicle moving along a two-lane, single direction roadway populated with stationary cars. The ego vehicle must weave between the cars safely, without leaving the road. We perform this experiment both in simulation and on a real automobile.

In training, the obstacle cars are initialized on the roadway at 30 meter intervals. The spacing of the obstacles is randomized by $\pm$ 5 meters and the lane that the obstacle appears in (left or right) is randomized as well. The dimensions of the vehicles are $4\,\text{m}\times 1.5\,\text{m}$, and each lane is $3\,\text{m}$ across.

The novice policy receives an observation of the ego vehicle's state consisting of distance from the median $y$, orientation $\theta$, speed $s$, distances to each edge of the current lane $(l_l, l_r)$, and distances to the nearest leading obstacle in each lane $(d_l, d_r)$. The policy then issues steering angle and speed commands to the ego vehicle.

Performance is evaluated primarily on road departure and collision rates. A road departure is when the center of mass of the ego vehicle leaves the road while under control of the novice policy. Rates were calculated on a per meter basis.

*1) Training:* For both the simulated and real-world experiments, we first trained a policy using behavioral cloning on 10,000 action labels collected from the expert. This policy was employed as an initialization for the three methods tested.[2] Each method then refined that initial policy over an additional five training epochs, each of which involved the accumulation of an additional 2,000 expert labels.[3] For DAGGER, we initialized the $\beta$ parameter at 0.85 and decayed it by a factor of 0.85 at the end of each training epoch.

*2) Vehicle Experiments:* We trained and tested policies with each method using an MG-GS vehicle provided by SAIC as the ego vehicle, and simulated vehicles as the stationary obstacles. The MG-GS vehicle was equipped with LiDAR and high-fidelity localization. A two-lane road with static obstacle cars was simulated for testing. A safety driver monitored the vehicle at all times during testing and training, while another human driver who could see the simulated obstacles and road used an auxiliary steering wheel and pedal set to send control inputs to the system. For HG-DAGGER data collection, the expert could retake control of the system by turning the steering wheel and could revert control to the novice by pressing a button. The vehicle and testing setup are shown in Figure 2.

*3) Simulation:* We trained additional policies in simulation in order to perform various evaluations that would be prohibitively time-consuming or dangerous to perform on a physical car. Ego vehicle dynamics were approximated using a bicycle model with parameters such as distance from center of gravity to axles determined by the physical test vehicle.[4]

[2]A behavioral cloning initialization was used because both DAGGER and HG-DAGGER rely on the novice to shape the sampling distribution used to acquire action labels from the expert. The state distribution induced by a completely untrained novice policy places density on regions of the state space that will not be visited once the task is learned.

[3]Except for the final HG-DAGGER training epoch, which incorporated fewer expert labels due to the novice policy's ability to successfully avoid the unsafe regions of the state space without human intervention.


Fig. 2: Test vehicle (L) and expert driver interface (R).

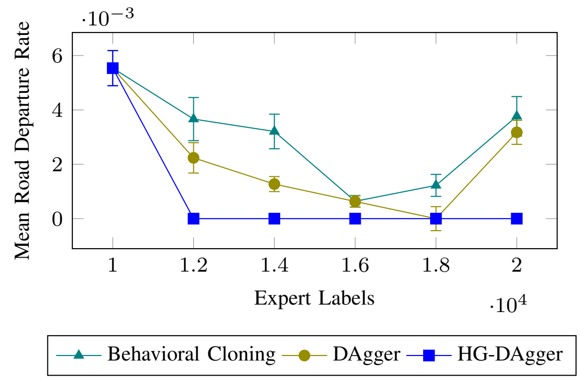


Fig. 3: Mean road departure rate per meter over training epochs. Error bars represent standard deviation.

## IV. RESULTS

Our results demonstrate that novice policies trained with HG-DAGGER outperform novice policies trained with DAGGER and behavioral cloning in terms of sample efficiency, training stability, and similarity to human behavior in our driving task. Additionally, our results demonstrate the significance of the doubt threshold learned by our method.

### A. Simulation: Learning Performance

We evaluate the effectiveness of each of the three methods as a function of the number of expert labels on which each novice policy was trained. The same eight randomly selected obstacle configurations and initializations were used for the evaluation of each policy. Learning curves from these tests are displayed in Figure 3 and Figure 4. The x-axis in each chart shows the number of expert labels on which each intermediate novice policy is trained.

HG-DAGGER demonstrates faster and more stable learning than DAGGER and behavioral cloning as measured by road departure rate and collision rate. All rates are per meter. One interesting feature of the learning curves for DAGGER is the instability demonstrated in later epochs. As the parameter $\beta$ is decayed over training epochs, the novice policy is given control a larger percentage of the time. We hypothesize that in this situation, perceived actuator lag begins to affect the labels provided by the human expert. The observed instabilities may be a result of a concomitant deterioration in the quality of collected expert action labels.

[4]Simulations were implemented using AutomotiveDrivingModels.jl

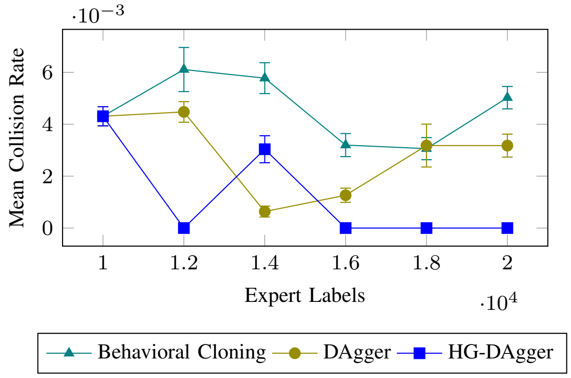


Fig. 4: Mean collision rate per meter over training epochs. Error bars represent standard deviation.

TABLE I: Mean collision and road departure rates per meter, and mean road departure duration in seconds, for rollouts initialized within or outside the permissible set.

| Initialization | Collision Rate | Road Departure Rate | Departure Duration |
|---|---|---|---|
| $\hat{\mathcal{P}}$ | $0.607 \times 10^{-3}$ | $0.607 \times 10^{-3}$ | 1.630 |
| $\hat{\mathcal{P}}'$ | $7.533 \times 10^{-3}$ | $12.092 \times 10^{-3}$ | 3.740 |

### B. Simulation: Safety and Risk Evaluation

We evaluated the adequacy of novice doubt as a risk metric as well as the significance of the learned doubt threshold $\tau$ by examining the novice's performance when initialized inside and outside of the estimated permissible set $\hat{\mathcal{P}}$. The set $\hat{\mathcal{P}}$ is an approximation to the true permitted set $\mathcal{P}$ derived using the doubt approximation of risk and the learned threshold $\tau$:

$$\hat{\mathcal{P}} = \{x_t \mid d_N(\mathcal{O}(x_t)) \leq \tau\} \tag{5}$$

where $d_N(o_t)$ is the novice's doubt given observation $o_T$. The complement of $\hat{\mathcal{P}}$, the estimated unsafe set, is $\hat{\mathcal{P}}'$.

To perform this experiment, we defined a set $S$ of conservative initializations. Given the ego vehicle's pose $(x, y, \theta)$, its speed $s$, and the distance to the nearest leading obstacle in each lane $(d_l, d_r)$, we define $S$ as:

$$\begin{aligned} S = \{(x, y, \theta, s) \mid & y \in [-6, 6] \text{ meters}, && \text{(6a)}\\ & \theta \in [-15, 15] \text{ degrees}, && \text{(6b)}\\ & s \in [4, 5] \text{ m/s}, && \text{(6c)}\\ & \max(d_l, d_r) < 8 \text{ meters}\} && \text{(6d)} \end{aligned}$$

One group of initializations was sampled uniformly from $\hat{\mathcal{P}} \cap S$ and and another group from $\hat{\mathcal{P}}' \cap S$. Using these set intersections rather than sampling uniformly over the permissible set and its complement ensures that the states sampled from $\hat{\mathcal{P}}'$ are not unrealistically dangerous states, which would be unlikely to be encountered in a real rollout. Additionally, sampling from this set shows that our method is capable of distinguishing between similar regions of the state space on the basis of the novice policy's estimated risk within those regions. Performance was evaluated using collision rate and road departure rate.

From the results in Table I, we see that the novice policy performed significantly better when initialized inside of $\hat{\mathcal{P}}$: the mean collision rate was 12 times lower and the mean road departure rate was 20 times lower than when initialized outside of $\hat{\mathcal{P}}$. Furthermore, the average duration of those road departures that did occur when the novice had been initialized inside of $\hat{\mathcal{P}}$ was less than half of the average duration of the road departures that occurred when the novice had been initialized outside of the permitted set.

These results highlight the utility of novice doubt as a model-free risk approximator. Furthermore, they show that HG-DAGGER learns a doubt threshold $\tau$ that can be used to distinguish similar states that are nonetheless distinct in terms of their riskiness.

### C. Test Vehicle: Driving Performance

Policies trained on the test vehicle were evaluated on a fixed set of five random obstacle configurations in the same manner as were the policies trained in simulation. Quantitative results from these tests can be seen in Table II. The novice trained with HG-DAGGER had the fewest collisions and road departures of the three methods. Furthermore, the steering angle distribution induced by the HG-DAGGER policy was 21.1% closer to the human driving data, as measured by Bhattacharyya distance [23], than was the distribution induced by DAGGER, indicating more human-like behavior. However, the limited amount of on-vehicle test data limits the statistical significance of these results, and they should be interpreted primarily as a heuristic that our method could be a good candidate for further real-world evaluation.

The test trajectories themselves are visualized in Fig. 5.[5] The HG-DAGGER novice's trajectories appear qualitatively superior to those of DAGGER and behavioral cloning, as the HG-DAGGER policy maintains a safer distance from the edge of the road than the DAGGER policy and does not deviate from the roadway, like the behavioral cloning policy.

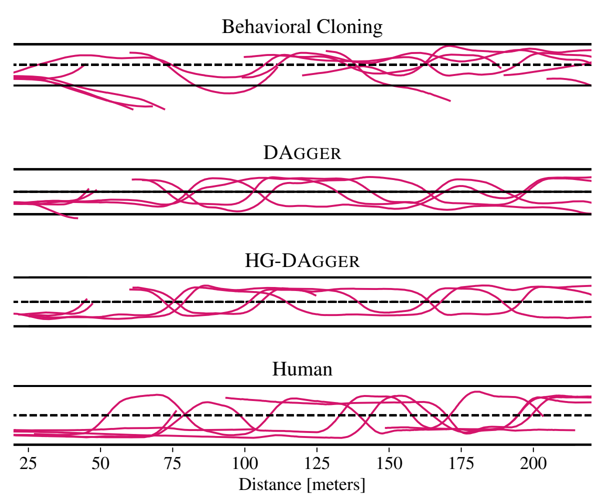


Fig. 5: Trajectory plots of on-vehicle test data.

[5]Discontinuities in Fig. 5 represent points where testing had to be halted due to pedestrians or other vehicles entering the test area.

TABLE II: Summary of on-vehicle test data. Totals are for the first 5,000 samples collected.

| | # Collisions | Collisions Rate | # Road Departures | Road Departure Rate | Bhattacharyya Metric |
|---|---|---|---|---|---|
| Behavioral Cloning | 1 | $0.973 \times 10^{-3}$ | 6 | $5.837 \times 10^{-3}$ | 0.1173 |
| DAGGER | 1 | $1.020 \times 10^{-3}$ | 1 | $1.020 \times 10^{-3}$ | 0.1057 |
| Human-Gated DAGGER | **0** | **0.0** | **0** | **0.0** | **0.0834** |

### D. Test Vehicle: Safety and Risk Evaluation

A good doubt threshold $\tau$ should be low enough to exclude dangerous regions from the estimated permitted set $\hat{\mathcal{P}}$, also high enough so that it does not exclude safe regions from $\hat{\mathcal{P}}$. Therefore, to complement the results described in Section IV-B, and to further validate the utility of the doubt metric and the associated threshold $\tau$ learned by HG-DAGGER, we examined the correspondence between $\hat{\mathcal{P}}$ and free space, and between $\hat{\mathcal{P}}'$ and occupied space. Occupied space in this case corresponds to all points off of the road as well as on-road points that fall within any of the obstacle vehicles.[6]

We evaluate this correspondence by discretizing the workspace and using binary classification performance metrics on a pixelwise basis. Individual pixels were assigned to $\hat{\mathcal{P}}$ or $\hat{\mathcal{P}}'$ by sampling novice doubt along constant curvature trajectories and then performing linear interpolation.

Figure 6 is a visualization of three risk maps generated by this process, for a single obstacle configuration, and for the doubt threshold learned by the algorithm (center), compared to two other thresholds. Figure 6 demonstrates how the learned value of $\tau$ is meaningful: the map generated with $\tau$ provides a good approximate characterization of the riskiness of different parts of the workspace. Increasing $\tau$ by a relatively small constant factor, however, causes dangerous regions to be inaccurately characterized as safe, while decreasing $\tau$ creates the opposite problem of an overly-conservative characterization of risk.

Figure 7 demonstrates that these results are not limited to the single obstacle configuration seen in Figure 6. The chart shows various performance metrics for the pixel-wise free space vs. occupied space classification task as functions of the doubt threshold used. Each choice of threshold was evaluated on 40 randomly generated obstacle configurations. The chart shows that the threshold learned from HG-DAGGER is near-optimal for all performance metrics examined.

## V. CONLCUSION

While interactive imitation learning algorithms like DAGGER address some of the limitations of behavioral cloning, it is not straightforward to apply such methods to the task of learning from human experts. By limiting the control authority of the human expert, these methods can degrade the quality of collected action labels and can compromise the safety of the combined expert-novice system.

In response to these challenges, we have presented HG-DAGGER, a DAGGER-variant that enables more effective learning in the human-in-the-loop context. HG-DAGGER learns both a novice policy and a safety threshold for a doubt metric that can be used to estimate risk in different regions of the state space at test-time. We demonstrated the efficacy of our method and of the learned risk threshold in simulated and real-world autonomous driving experiments, showing improved sample efficiency and increased training stability relative to DAGGER and behavioral cloning.

Future work will involve the use of doubt-based risk metrics as inputs to an automated gating mechanism to switch between sub-policies in a hierarchical controller. We also plan to investigate more sophisticated ways of estimating model uncertainty and linking it with execution risk.

[6] We note that free space and occupied space are only an approximation of safe and dangerous regions, since there exist points in free space that still fall within the region of inevitable collision.

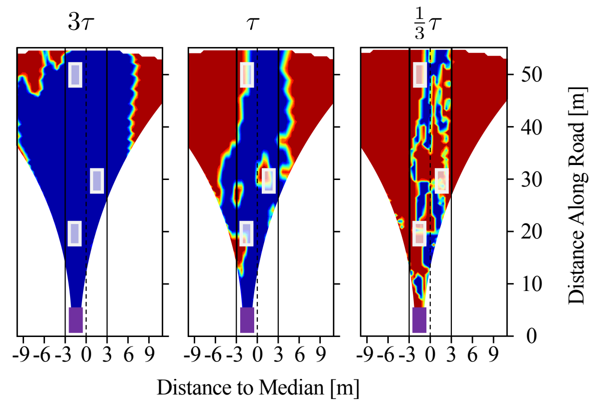


Fig. 6: Risk maps generated for a policy trained on the test vehicle. The center map was generated using the variance threshold $\tau$ learned from human interventions. The purple box represents the ego vehicle, and the white boxes represent other vehicles. Blue is safe and red is unsafe.

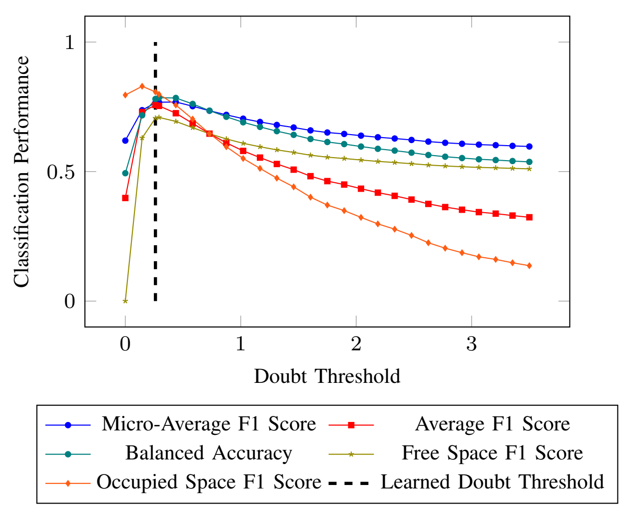


Fig. 7: Performance on the pixelwise free vs. occupied space classification task as a function of the doubt threshold used.

## ACKNOWLEDGMENTS


We thank Kunal Menda for inspiring this work and Xiaobai Ma for his help interfacing with the test vehicle.

# References


[1] B. Price and C. Boutilier, “Accelerating reinforcement learning through implicit imitation,” *Journal of Artificial Intelligence Research*, vol. 19, pp. 569–629, 2003.

[2] J. Kober and J. Peters, “Imitation and reinforcement learning,” *IEEE Robotics Automation Magazine*, vol. 17, no. 2, pp. 55–62, June 2010.

[3] S. Ross and D. Bagnell, “Efficient reductions for imitation learning,” in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 661–668.

[4] M. Laskey, C. Chuck, J. Lee, J. Mahler, S. Krishnan, K. Jamieson, A. Dragan, and K. Goldberg, “Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 358–365.

[5] A. Goil, M. Derry, and B. D. Argall, “Using machine learning to blend human and robot controls for assisted wheelchair navigation,” in *IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2013.

[6] W. S. Kim, B. Hannaford, and A. Fejczy, “Force-reflection and shared compliant control in operating telemanipulators with time delay,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 2, pp. 176–185, 1992.

[7] V. A. Shia, Y. Gao, R. Vasudevan, K. Driggs-Campbell, T. Lin, F. Borrelli, and R. Bajcsy, “Semiautonomous vehicular control using driver modeling,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2696–2709, 2014.

[8] M. S. Branicky, “Multiple Lyapunov functions and other analysis tools for switched and hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 475–482, 1998.

[9] D. T. McRuer, “Pilot-induced oscillations and human dynamic behavior,” *NASA Technical Report*, 1995.

[10] D. McRuer, *Aviation safety and pilot control: Understanding and preventing unfavorable pilot-vehicle interactions*. National Academies Press, 1997.

[11] J. Zhang and K. Cho, “Query-efficient imitation learning for end-to-end autonomous driving,” *arXiv*, no. 1605.06450, 2016.

[12] M. Laskey, S. Staszak, W. Y.-S. Hsieh, J. Mahler, *et al.*, “Shiv: Reducing supervisor burden in DAgger using support vectors for efficient learning from demonstrations in high dimensional state spaces,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 462–469.

[13] B. Kim and J. Pineau, “Maximum mean discrepancy imitation learning,” in *Robotics: Science and Systems*, 2013.

[14] K. Menda, K. Driggs-Campbell, and M. J. Kochenderfer, “EnsembleDAgger: A Bayesian approach to safe imitation learning,” *arXiv*, no. 1807.08364, 2018.

[15] K. Driggs-Campbell, V. Shia, and R. Bajcsy, “Improved driver modeling for human-in-the-loop vehicular control,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[16] V. Govindarajan, K. Driggs-Campbell, and R. Bajcsy, “Data-driven reachability analysis for human-in-the-loop systems,” in *IEEE Conference on Decision and Control (CDC)*, 2017, pp. 2617–2622.

[17] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 6405–6416.

[18] S. Chernova and M. Veloso, “Interactive policy learning through confidence-based autonomy,” in *Journal of Artificial Intelligence Research (JAIR)*, 2009, pp. 1–25.

[19] H. He, H. Daumé III, and J. Eisner, “Imitation learning by coaching,” in *Thirty-second Conference on Neural Information Processing System*, 2012.

[20] D. A. Pomerleau, “ALVINN: An autonomous land vehicle in a neural network,” in *Advances in Neural Information Processing Systems*, 1989, pp. 305–313.

[21] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, “End to end learning for self-driving cars,” *arXiv*, no. 1604.07316, 2016.

[22] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou, and B. Boots, “Agile off-road autonomous driving using end-to-end deep imitation learning,” *arXiv*, no. 1709.07174, 2017.

[23] A. Bhattacharyya, “On a measure of divergence between two statistical populations defined by their probability distributions,” *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943.