Enhancing Safety in Learning from Demonstration Algorithms via Control Barrier Function Shielding

Yue Yang*
Letian Chen*
Zulfiqar Zaidi*
letian.chen@gatech.edu
Georgia Institute of Technology
Atlanta, GA, USA

Sanne van Waveren
Arjun Krishna
Matthew Gombolay
matthew.gombolay@cc.gatech.edu
Georgia Institute of Technology
Atlanta, GA, USA

ABSTRACT

Learning from Demonstration (LfD) is a powerful method for non-roboticists end-users to teach robots new tasks, enabling them to customize the robot behavior. However, modern LfD techniques do not explicitly synthesize safe robot behavior, which limits the deployability of these approaches in the real world. To enforce safety in LfD without relying on experts, we propose a new framework, **S**hi**E**lding with **C**ontrol barrier f**U**nctions in inverse **RE**inforcement learning (SECURE), which learns a customized Control Barrier Function (CBF) from end-users that prevents robots from taking unsafe actions while imposing little interference with the task completion. We evaluate SECURE in three sets of experiments. First, we empirically validate SECURE learns a high-quality CBF from demonstrations and outperforms conventional LfD methods on simulated robotic and autonomous driving tasks with improvements on safety by up to 100%. Second, we demonstrate that roboticists can leverage SECURE to outperform conventional LfD approaches on a real-world knife-cutting, meal-preparation task by 12.5% in task completion while driving the number of safety violations to zero. Finally, we demonstrate in a user study that non-roboticists can use SECURE to effectively teach the robot safe policies that avoid collisions with the person and prevent coffee from spilling.

CCS CONCEPTS

• **Computing methodologies** → **Learning from demonstrations**; • **Theory of computation** → **Inverse reinforcement learning**; • **Software and its engineering** → **Software safety**.

KEYWORDS

Learning from Demonstration, Control Barrier Function, Safety

ACM Reference Format:
Yue Yang, Letian Chen, Zulfiqar Zaidi, Sanne van Waveren, Arjun Krishna, and Matthew Gombolay. 2024. Enhancing Safety in Learning from Demonstration Algorithms via Control Barrier Function Shielding. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24), March 11–14, 2024, Boulder, CO, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3610977.3635002

*Authors contributed equally.

HRI '24, March 11–14, 2024, Boulder, CO, USA

ACM ISBN 979-8-4007-0322-5/24/03.
https://doi.org/10.1145/3610977.3635002

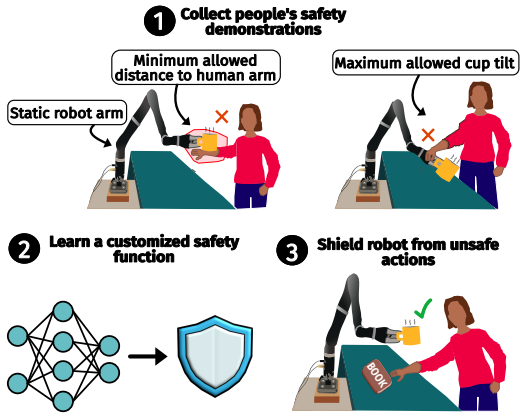


Figure 1: This figure shows an example of a person providing safety demonstrations from which the robot learns a customized safety function that shields it from unsafe actions.

1 INTRODUCTION

Recent advances in robot learning have offered the potential to aid people in a range of applications, including driving [47], manufacturing [48], and household tasks [10], like tidying up or serving someone a drink. Reinforcement learning (RL) has become a ubiquitous approach to develop robot controllers; however, defining the reward function to elicit desired behaviors can be difficult, and engineered reward functions might overfit to particular RL algorithms [7]. Instead, the field of Learning from Demonstration (LfD) seeks to empower non-roboticist end-users to teach robots skills and customized behaviors through demonstrations [13, 14, 23, 39].

Like RL, LfD research has yielded strong results in laboratory settings [13, 14, 36], but few techniques exist for LfD that enable robots to learn safe policies, hindering the deployment of LfD with end-users in the real world. Recently, Brown et al. [8] provided high-confidence bounds for quality of the inferred human intention as a proxy of safety. While promising, such approaches do not allow specifying constraints on the learned policy to explicitly prevent the robot from taking unsafe actions.

To ensure safety, Control Barrier Functions (CBFs) are a state-of-the-art method for designing safe robotic controllers that adhere to explicit safety constraints. CBFs have successfully been applied in RL and HRI settings [3, 4, 16, 29, 30, 35, 46], and we hypothesize that CBFs could similarly help learned LfD policies to avoid unsafe states. However, conventional CBF approaches would still require experts to formally define and construct such constraints. Instead,

we aim to enforce safety in LfD settings without relying on experts by allowing users to define safety via demonstration.

We present SECURE, a novel Safe Learning from Demonstrations (LfD) framework that learns personalized CBFs from end-user demonstrations. In contrast to approaches solely focusing on physical safety, SECURE acknowledges the variability in individuals' safety preferences [24, 38]. This user-centric approach not only enhances perceived safety but also ensures physical safety, as demonstrated in a coffee serving task where safety demonstrations define minimum distance and maximum cup angle to avoid spills (see Figure 1). Our contributions in this work are four-fold:

(1) We propose a new framework named ShiElding with Control barrier fUnctions in inverse REinforcement learning (SECURE), that learns a CBF from human demonstrations. We then develop two techniques, namely *CBF Shield* and *Adaptive Resampling*, which shield the LfD policy to be safe and enhance the sample efficiency of SECURE for improved usability in HRI;
(2) We demonstrate SECURE's ability to learn a high-quality CBF, in comparison to an expert-designed CBF in 2D Double Integrator system. Empirical evaluation on simulated robot control tasks showcases SECURE's task performance on par or exceeding Learning from Demonstrations (LfD) baselines, while significantly reducing safety constraint violations by up to 100%.
(3) We demonstrate that roboticists can leverage SECURE to synthesize safe policies from demonstrations on a real-world knife-cutting, meal-preparation task. SECURE outperforms conventional LfD approaches by 12.5% in task completion and eliminates **100%** unsafe cases (i.e., "cut" human arms);
(4) We further conduct a user study in which participants first provide demonstrations in a coffee-cup placing task and then work on a secondary task in the robot's proximity. SECURE can effectively learn user-specific safe policies from provided demonstrations to enable the robot to complete its task while being perceived as safe by users operating in its proximity.

2 RELATED WORK

Ensuring safe and reliable robot operation, particularly in interactions with human users, is of paramount importance [9]. In the RL realm, safety challenges arise due to the learning process's exploration in unknown environments, where various safety approaches tailored to RL have emerged, including constrained policy optimization [1, 17, 32, 40, 43], safe exploration [20, 33, 34], learning a safety critic [5, 41, 44], risk-averse RL [45, 51], and shielding [2, 11]. Shielding, in particular, is a framework that ensures the safety of a control policy by verifying that each action applied keeps the system within a predefined safe set of states [6]. CBFs are mathematical functions utilized in control theory to enforce safety constraints by defining a safe set of states [3, 4]. CBFs are a popular technique to shield robots from unsafe actions, as they enforce the system to always remain within a set of safe states.

To develop safe controllers, prior work has explored synthesizing CBFs from data, including expert demonstrations [26, 27, 37, 42]. However, these approaches work with expert demonstrations, limiting their applicability with end-users, which is central in LfD. Researchers have also explored tuning specific CBF parameters according to user data [18, 25, 31, 46]. In the context of RL safety, researchers have investigated the utilization of expert-designed CBFs to synthesize control policies that confine the system within safe states [15, 16, 29, 30, 35]. Recent efforts have also focused on leveraging data-driven methods to learn CBFs within the RL framework for safety assurance [50]. However, these approaches have been limited to RL and have not been extended to LfD methods where robots directly learn from and interact with humans.

While a recent method extended CBF to the domain of imitation learning [19], it requires a manually-designed CBF to supplement the Behavioral Cloning (BC) policy, which is not practical for real-world LfD settings. Castañeda et al. [12] proposes to construct a CBF from data to detect out-of-safe-distribution cases. Still, the approach risks being overly conservative. To the best of our knowledge, our study is the first to successfully integrate CBFs with IRL algorithms and effectively increase policy performance while mitigating potential safety concerns.

3 PRELIMINARIES

In this section, we introduce three building blocks of SECURE: Markov Decision Process, Inverse Reinforcement Learning, and Control Barrier Function.

Markov Decision Process: We model the environment as a Markov Decision Process (MDP) [49], $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, R, T, \gamma, \rho_0 \rangle$. $\mathcal{S}$ and $\mathcal{A}$ denote the state and action space, respectively. $R : \mathcal{S} \to \mathbb{R}$ is the reward of a given state. $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a deterministic transition function that gives the next state, s', for applying the action, a, in state, s. $\gamma \in (0, 1)$ is the temporal discount factor. $\rho_0 : \mathcal{S} \to \mathbb{R}$ denotes the initial state probability distribution. A stochastic policy $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a mapping from states to probabilities over actions. A trajectory, $\tau = (s_0, a_0, \cdots, s_t, a_t, \cdots)$, is generated by executing the policy within the environment: $s_0 \sim \rho_0, a_t \sim \pi(s_t), s_{t+1} = T(s_t, a_t)\ \forall t \geq 0$. The expected discounted return of a policy, π, is calculated by $J(\pi) = \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t)\right]$. The objective for RL is to find the optimal policy, $\pi^* = \arg\max_\pi J(\pi)$.

Inverse Reinforcement Learning (IRL) infers a reward function, $\hat{R}$, from a set of demonstration trajectories, $\mathcal{D} = \{\tau_i\}_{i=1}^N$. Our method is based on adversarial IRL (AIRL) [21], which consists of a generator (i.e., a policy) to imitate the demonstrator and a discriminator to distinguish the generator's behaviors from the demonstrator's. The discriminator D is trained to minimize the cross entropy loss, $\mathcal{L}_{\text{Discriminator}} = -\mathbb{E}_{\tau \sim \mathcal{D}, (s,a,s') \sim \tau}[\log D(s, a, s')] - \mathbb{E}_{\tau \sim \pi_\phi, (s,a,s') \sim \tau}[\log(1 - D(s, a, s'))]$. The generator policy, $\pi_\phi(a|s)$, is trained by optimizing the policy loss, $\mathcal{L}_{\text{policy}} = -J_\theta(\pi_\phi)$, to maximize the pseudo reward function which is given by $r_\theta(s, a, s') \triangleq \log D_\theta(s, a, s') - \log(1 - D_\theta(s, a, s'))$.

Control Barrier Functions (CBFs) define a set of safe states, $\mathcal{S}_s$, and a set of unsafe (or dangerous) states, $\mathcal{S}_d$. A CBF, h, needs to satisfy the following three requirements (R1-R3) [3, 28]: **R1**: $\forall s \in \mathcal{S}_s$, $h(s) \geq 0$; **R2**: $\forall s \in \mathcal{S}_d$, $h(s) < 0$; **R3**: $\forall s \in \{s | h(s) \geq 0\}$, $\frac{h(T(s, \pi_\phi(s))) - h(s)}{\Delta t} + \alpha(h(s)) \geq 0$, where $\alpha(\cdot)$ is a class-$\mathcal{K}$ function, i.e., $\alpha(\cdot)$ is strictly increasing and $\alpha(0) = 0$. Intuitively, the three requirements ensure trajectories to stay inside the superset, $C_h = \{s \in \mathcal{S} : h(s) \geq 0\}$, and never visit unsafe states where $h(s) < 0$. In order to obtain a CBF, $h(\cdot)$, and a safe policy, $\pi_\phi(\cdot)$, that meet the three requirements, we formulate an objective similar to Qin et al. [35], as shown in Equation 1. **R1**-**R3** are satisfied when we find $h(\cdot)$ and $\pi_\phi(\cdot)$ such that $y(h, \pi_\phi) > 0$, i.e., our optimization

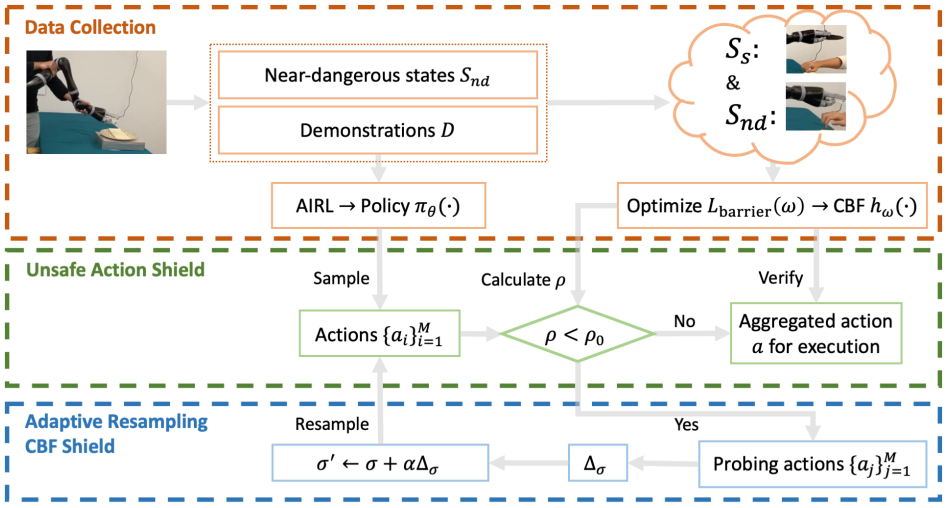


Figure 2: This figure illustrates SECURE's architecture. End-users contribute demonstrations and near-dangerous states to train the policy, $\pi_\theta(\cdot)$, and CBF, $h_\omega(\cdot)$. *CBF Shield* prevents the IRL policy from entering dangerous states while minimizing interference with task completion. *Adaptive sampling* introduced in *CBF Shield* generates safe and task-aware actions efficiently.

objective is to maximize y.

$$y(h,\pi_\phi) \triangleq \min\Big\{\inf_{s\in\mathcal{S}_s} h(s),\quad \inf_{s\in\mathcal{S}_d} -h(s),\quad \inf_{\{s|h(s)\geq 0\}} \frac{h(T(s,\pi_\phi(s))) - h(s)}{\Delta t} + \alpha(h(s))\Big\} \tag{1}$$

4 METHOD

We describe SECURE in three steps: In Section 4.1, we first describe how SECURE learns a CBF, represented by a neural network, from user-provided safety demonstrations (Figure 2, top). Second, Section 4.2 describes how SECURE utilizes a shielding mechanism with the learned neural CBF to prevent the robot from entering dangerous states while still allowing for task completion (Figure 2, middle). Finally, in Section 4.3, we introduce a novel *adaptive sampling* method for SECURE that improves the efficiency in finding safe and task-aware actions (Figure 2, bottom).

4.1 Safe LfD with CBF

To enable end-users to define customized safety boundaries, we seek to learn user-specific safety constraints, represented by a CBF, from user demonstrations. To learn the CBF, we need access to the safe states set, $\mathcal{S}_s$, and the unsafe states set, $\mathcal{S}_d$. While we can construct the safe state set with demonstrations: $\mathcal{S}_s = \{s|s \in \tau \in \mathcal{D}\}$, we should not request demonstrators to take the risk of hurting themselves to provide unsafe demonstrations. Instead, we define the *near dangerous* state set, $\mathcal{S}_{nd}$, as a set that the robot has to pass before entering $\mathcal{S}_d$, shown in Equation 2.

$$\forall\tau \text{ with } s_0 \in \mathcal{S}_s, t>0 \quad \nexists s_t \in \mathcal{S}_d \quad \text{s.t.} \quad \forall 0<t'<t, s_{t'} \notin \mathcal{S}_{nd} \tag{2}$$

Intuitively, $\mathcal{S}_{nd}$ would be a set that "wraps" the actual physically unsafe states, e.g. collisions. For instance, if a robot helps a person with serving a cup of coffee, the person can demonstrate near-dangerous states by moving their arms around the static robot arm holding the cup of coffee at distances that they perceive as near-dangerous. Note that one user may define a large distance as "near" dangerous even if the expected harm may be low, and SECURE respects such user-defined safety concepts.

Having defined $\mathcal{S}_{nd}$, we amend the CBF's second requirement as **R2′**: For $\forall s \in \mathcal{S}_{nd}$, $h(s) < 0$. As a corollary of the CBF property introduced in Section 3, if **R1**, **R2′**, and **R3** are satisfied, the policy cannot enter $\mathcal{S}_{nd}$, which further means the policy cannot enter the dangerous state set, $\mathcal{S}_d$, according to the definition of $\mathcal{S}_{nd}$. While **R2′** is a stricter requirement than **R2**, it allows people to personally demonstrate what they deem as unsafe. We replace $\mathcal{S}_d$ in Equation 1 to be $\mathcal{S}_{nd}$, resulting in Equation 3.

$$y'(h,\pi_\phi) \triangleq \min\Big\{\inf_{s\in\mathcal{S}_s} h(s),\quad \inf_{s\in\mathcal{S}_{nd}} -h(s),\quad \inf_{\{s|h(s)\geq 0\}} \frac{h(T(s,\pi_\phi(s))) - h(s)}{\Delta t} + \alpha(h(s))\Big\} \tag{3}$$

Finding a solution of h and π for $y' > 0$ will satisfy CBF requirements and ensure that the agent does not enter dangerous states or near dangerous states. One observation to maximize y is that the first two terms are only dependent on the CBF, h, while the third term relies on π_ϕ. Although one can jointly optimize h and π_ϕ, such an optimization suffers from empirical difficulty because π_ϕ is chasing the moving h. To show this, we conduct an empirical experiment in the demolition derby domain (see Section 6). Joint optimization of h and π_ϕ yields a 32.3% ± 11.0% success rate with a high 77.7% ± 3.4% occurrence of dangerous cases. SECURE instead takes a two-stage approach: 1) optimize the CBF, h, to satisfy **R1** and **R2′**; 2) modulate π_ϕ to satisfy **R3** by the *CBF shield* we introduce in Section 4.2. As a result, SECURE achieves a high 52.3% ± 2.5% success rate and a low 3.3% ± 1.2% occurrence of dangerous cases.

For Stage 1, we formulate the loss function $\mathcal{L}_{\text{barrier}}$ as shown in Equation 4, where $h_\omega(\cdot)$ is a neural network parameterized by ω. Intuitively, minimizing $\mathcal{L}_{\text{barrier}}$ provides an $h_\omega(\cdot)$ that can discriminate safe states which have positive h values and near-dangerous

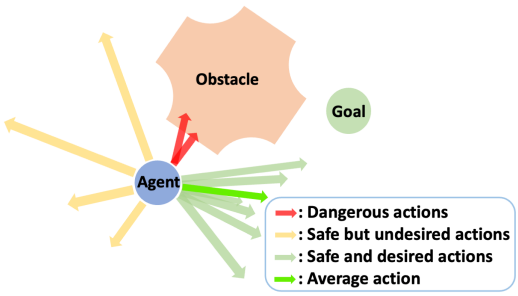


Figure 3: This figure shows that *CBF Shield* identifies an action that is safe and does not hinder task completion.

Algorithm 1: *CBF shield* Action Choice

Input : Learned CBF $h_\omega(\cdot)$, Policy $\pi_\phi(\cdot|s)$, Current state s, Sampling batch size M, Safe action percentage requirement ρ_0

1 $\mu, \sigma \leftarrow \pi_\phi(\cdot|s)$
2 $\{a_i\}_{i=1}^M \sim \mathcal{N}(\mu, \sigma)$
3 **while** $\frac{\sum_{i=1}^M \mathbb{I}(g(a)>0)}{M} \leq \rho_0$ **do**
4 $\quad \mu, \sigma \leftarrow$ `AdaptiveResampling` (μ, σ)
5 $\quad \{a_i\}_{i=1}^M \sim \mathcal{N}(\mu, \sigma)$
6 $\overline{a} \leftarrow \frac{1}{M} \sum_{i=1}^M [\mathbb{I}(g(a) > 0) \cdot a_i]$
7 **if** $g(\overline{a}) > 0$ **then**
$\quad$ **Output :** $\overline{a}$
8 **else**
9 $\quad \widetilde{a} \leftarrow \min_{a \in \{a_i | g(a_i) \geq 0\}_{i=1}^M} \|a - \overline{a}\|$
$\quad$ **Output :** $\widetilde{a}$

states which have negative h values, when trained on the safe and near-dangerous states specified through demonstrations.

$$\mathcal{L}_{\text{barrier}}(\omega) = \sum_{s \in \mathcal{S}_s} \max(-h_\omega(s), 0) + \sum_{s \in \mathcal{S}_{nd}} \max(h_\omega(s), 0) \quad (4)$$

4.2 Shielding Unsafe Actions

After learning the CBF, $h_\omega(\cdot)$, from human demonstrations for encoding safe and near-dangerous states, one naïve way to avoid danger is to choose actions with $h_\omega > 0$. However, this approach is myopic which can lead to danger. Consider a scenario where a fast-moving vehicle approaches unsafe states: merely choosing actions with $h_\omega > 0$ results in the vehicle approaching the unsafe boundary and inevitably entering an unsafe state. In contrast, CBF **R3** (Equation 5, where $a \sim \pi_\phi(\cdot|s)$) enables SECURE to assess the gradual decline of h_ω from safe to unsafe states, ensuring the agent never enters unrecoverable states. Therefore, SECURE employs the *CBF Shield* to find actions aligned with **R3**.

$$\mathcal{L}_{\text{derivative}}(\phi) = g(a) \triangleq \frac{h_\omega(T(s,a)) - h_\omega(s))}{\Delta t} + \alpha(h_\omega(s)) \geq 0 \quad \forall s \ s.t. \ h_\omega(s) \geq 0 \quad (5)$$

CBF shield directly finds safe actions that satisfy **R3**, i.e., $\mathcal{L}_{\text{derivative}} \geq 0$. We summarize the *CBF shield* procedure in Algorithm 1. For each safe action choice, we begin by sampling a batch of actions $\{a_i\}_{i=1}^M$ from the AIRL policy (lines 1-2). Specifically, the policy output is modeled as a Gaussian distribution with $\mu_\omega(s)$ and $\sigma_\omega(s)$, and the action is sampled by $a_i \sim \mathcal{N}(\mu_\omega(s), \sigma_\omega(s))$. Next, a straightforward approach could be randomly selecting one safe action from the batch of actions. However, while the selected action is safe, it is possible that the action interferes with the task completion (yellow arrows in Figure 3). Instead, *CBF Shield* aggregates multiple safe actions (green arrows in Figure 3) to better reflect the policy's intention of accomplishing the task. As such, we calculate the ratio of safe actions within a sampled action batch, $\rho = \frac{\sum_{i=1}^M \mathbb{I}(g(a) \geq 0)}{M}$, where M is the sampled batch size. When the ratio ρ exceeds a threshold, ρ_0, we have more confidence that the average of the safe actions aligns well with the policy mean output (i.e., aims at accomplishing the task). Thus, we aggregate safe actions within this batch (Line 6). When $\rho \leq \rho_0$, it suggests that the current batch does not contain enough safe actions and we resort to the *Adaptive Sampling* method (Section 4.3) to explore and find more safe actions efficiently (Line 4-5).

To ensure the safety of the executed action, we aggregate the safe actions by averaging first, $\overline{a} = \frac{1}{M} \sum_{i=1}^M [\mathbb{I}(g(a) \geq 0) \cdot a_i]$ (Line 6). If the averaged action (brighter green arrow in Figure 3) is deemed safe, $g(\overline{a}) \geq 0$ (Line 7), $\overline{a}$ is returned for execution. Otherwise, we select the closest action from the safe action set, $\widetilde{a} = \min_{a \in \{a_i | g(a_i) \geq 0\}_{i=1}^M} \|a - \overline{a}\|$ (Line 9). In summary, the procedure of *CBF shield* ensures the satisfaction of **R3** (i.e., policy safety) by always returning an action a such that $g(a) \geq 0$ while also being task-aware, which helps the agent to accomplish the task while respecting personalized safety definitions.

4.3 Adaptive Resampling

The *CBF Shield* introduced in the Section 4.2 assumes a minimum percentage of safe actions to be in the sampled action batch in order to obtain an action that is both safe and task-aware. However, the AIRL policy may be overly confident in a task-oriented but unsafe action, and thus it might not sample an action batch containing even a single safe action, let alone enough for safe action aggregation. Therefore, there is a need to devise a strategy for greater exploration within the action space. To address this, SECURE modifies the policy action distribution, $\mathcal{N}(\mu_\omega, \sigma_\omega)$, and conducts resampling from the modified distribution. To preserve the task completion goal represented by the action mean, μ_ω, we refrain from modifying it to avoid disrupting the task. Instead, we amplify the standard deviation in certain directions. To reduce the probability of generating safe but undesired actions, we selectively increase the standard deviation specifically along the directions identified as unsafe.

Algorithm 2 and Figure 4 show how our approach finds unsafe directions and adjusts the standard deviation. First, we sample N probing actions (the blue and green arrows in Figure 4) uniformly from action space (Line 1). To determine the unsafe action direction, we compute a weighted average of unsafe probing actions (i.e., green arrows in Figure 4, identified by $h_\omega(\cdot) < 0$) where the weights are given by the negative h values (Line 2). We can then adjust the standard deviation (i.e., the purple lines) by taking a small step with size α, in the normalized direction of the unsafe actions (Line 3-4). A new batch of actions is sampled for a subsequent verification loop conducted by *CBF shield*. Our *Adaptive Sampling* approach provides an efficient way to find safe and effective actions.

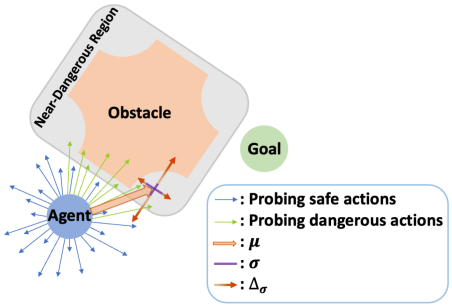


Figure 4: For *Adaptive Resampling*, we amplify the standard deviation, σ, by Δ_σ while keeping the action mean, μ. The amplification is greater in the direction of suspected near-dangerous regions.

Algorithm 2: Adaptive Resampling

Input : Learned CBF h_ω, Current state s, Policy output distribution mean μ and standard deviation σ, Probing extent R, Probing batch size N, Action dimention n, Standard deviation update step size α

$\{a_j\}_{j=1}^N \sim \mathcal{U}_n([-R, R]^n)$

$a_{\text{unsafe}} \leftarrow \sum_{j=1}^N [a_j \cdot \max(0, -h_\omega(T(s, a_j)))]$

$\Delta_\sigma \leftarrow \frac{|a_{\text{unsafe}}|}{\|a_{\text{unsafe}}\|}$, where $|\cdot|$ denotes element-wise absolute value and $||\cdot||$ denotes the two-norm

$\sigma' \leftarrow \sigma + \alpha\Delta_\sigma$

Output : μ, σ'

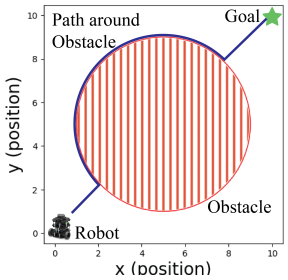


Figure 5: This figure illustrates the 2-D double integrator domain. The robot needs to go to the goal avoiding the obstacle. The blue curve is a feasible path for the robot.

5 VALIDATION OF SECURE'S LEARNED CBF

Notably, a known ground-truth CBF, defined by $h = \gamma[(x-x_{\text{obst}})^2 + (y-y_{\text{obst}})^2 - r_{\text{obst}}^2] + 2[(x-x_{\text{obst}})\cdot\dot{x} + (y-y_{\text{obst}})\cdot\dot{y}]$, serves as a reference to evaluate the performance of learned CBF, where (x, y) is the current coordinate, $(\dot{x}, \dot{y})$ is the current velocity vector, and $(x_{\text{obst}}, y_{\text{obst}}, r_{\text{obst}})$ represents the obstacle's position and radius.

We collect a dataset comprising of 800 safe states and 800 unsafe states by sampling from the state space and labeling each state with the ground-truth CBF to separate the impact of data quality and the CBF learning process itself. To test the learned CBF, we discretize the state space with a grid size of 0.1 within the ranges $[0, 10]$, $[0, 10]$, $[-1.5, 1.5]$, $[-1.5, 1.5]$, for x, y, $\dot{x}$, $\dot{y}$, respectively. As such, we obtain $100 \times 100 \times 30 \times 30 = 9,000,000$ test states. We summarize the evaluation results in Table 1, which shows a low overly-conservative rate (1.9%) and a low under-conservative rate (4.1%). We observe that SECURE is effective in learning a high-quality approximation of the ground-truth CBF with limited data. Additionally, SECURE strikes a good balance between being over-conservative and under-conservative.

Table 1: The table shows the means and standard deviations of the learned CBF's performance with five different random seeds for training on the 2D double integrator domain.

Predicted \ Ground-truth	Safe States	Unsafe States
Safe States	98.1% (1.0%)	4.1% (2.2%)
Unsafe States	1.9% (1.0%)	95.9% (2.2%)

6 SIMULATION EXPERIMENTS

We evaluate SECURE in the following simulated domains:

Demolition Derby Domain: a car is tasked to reach a target location while avoiding 16 other randomly moving cars (Figure 6). We utilize the approach from Qin et al. [35] to collect safe demonstrations by filtering out trajectories with collisions. We generate near-dangerous states by collecting states where the distance between the car and an obstacle is below a predefined threshold.

Panda Arm Push Domain: the objective is to push a block with a high center of gravity to a target location without toppling it [22] (Figure 7). We collect demonstrations by teleoperation via a keyboard. We collect three near-dangerous scenarios that knock down the block: a) pushing the upper part of the block (count: 442), b) pushing with high velocity (count: 590), and c) pushing the upper part of the block with high velocity (count: 444).

The number of safe and near-dangerous states for training the CBF, the number of demonstrations to train the policy, and the architecture of the neural network CBFs is tabulated in Table 2. Please refer to the supplementary for auxiliary details for the experiments.

6.1 Results

We develop two metrics to evaluate task completion and safety: "Success Rate," which quantifies the rate of successful task completion, and "Dangerous Rate," which is the rate of hazardous scenarios encountered. We evaluate both metrics across 100 trajectories with ten random seeds for both domains. Since SECURE is the first method to address safety issues for IRL, there is no existing benchmark tailored for the same task. Therefore, we select two baselines: 1) behavior cloning (BC), as BC remains a prevalent approach; 2) the state-of-the-art IRL approach, AIRL, as it has strong capability to imitate demonstrated behaviors.

The results are summarized in Table 3, showcasing the exceptional performance of SECURE. With BC displaying the lowest performance, our results analysis focuses on comparing SECURE and AIRL. In the demolition derby domain, AIRL and SECURE have similar success rates (two one-sided t-test with bound=10, $p < .01$) but SECURE achieves significantly less dangerous cases (71.2% less, Mann-Whitney $U = 0, p < .001$). In the Panda Arm Push domain, SECURE not only eliminates all instances of the block toppling over (comparing with AIRL, Mann-Whitney $U = 0, p < .001$) but also achieves a 43.7% improvement in the successful rate, significantly outperforming AIRL (Mann-Whitney $U = 99.5, p < .001$).

Table 2: Number of safe and near-dangerous states for CBF training, number of task demonstration states for policy learning, and neural network CBF's architecture in simulated and real-robot domains. CNN refers to Convolutional Neural Networks and FC refers to Fully-Connected networks with hidden layer node numbers specified in the parentheses.

	Demolition Derby	Panda Arm Push	Coffee Placing	Knife-cutting
Safe states	1024	1476	2500 (per participant)	450
Near-dangerous states	1024	1476	2500 (per participant)	450
Task demo states	52612	246	≈2000 (per participant) (user demonstration lengths vary)	2000
CBF NN	CNN akin to [35]	FC (32, 128, 128, 256, 256, 256, 256, 128, 128, 32)	FC (64, 64)	FC (32, 128, 128, 256, 256 256, 256, 128, 128, 32)

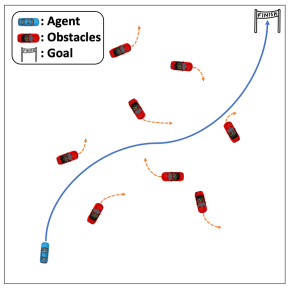


Figure 6: This figure shows the Demolition Derby domain.

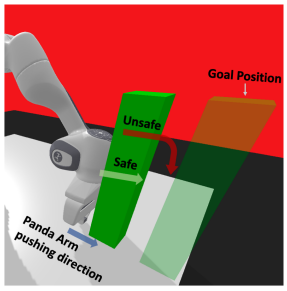


Figure 7: This figure illustrates the Panda Arm Push domain.

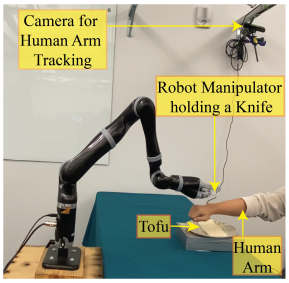


Figure 8: This figure shows the setup for the real-robot banana-cutting task.

Table 3: This table shows the comparison of SECURE (ours) with BC and AIRL in three domains. The standard deviation is calculated with ten runs of different random seeds for each algorithm. Bold denotes best performing algorithm.

		BC	AIRL	SECURE (ours)	SECURE Comparison with AIRL
Demolition Derby Domain (Evaluated on 100 Episodes)	Success Rate (Stdev)	17.9% (3.6%)	46.8% (4.7%)	**49.2%** (5.6%)	+2.4% (TOST $p < .01$ with bound=10)
	Dangerous Rate (Stdev)	65.7% (4.1%)	75.4% (4.9%)	**4.2%** (1.2%)	-71.2% (Mann-Whiteney $U = 0, p < .001$)
Panda Arm Push Domain (Evaluated on 100 Episodes)	Success Rate (Stdev)	22.7% (3.2%)	52.9% (22.6%)	**96.6%** (5.3%)	+43.7% (Mann-Whiteney $U = 99.5, p < .001$)
	Dangerous Rate (Stdev)	72.3% (3.5%)	31.3% (17.9%)	**0.0%** (0.0%)	-31.3% (Mann-Whitney $U = 0, p < .001$)
Kitchen Cutting Domain (Evaluated on 10 Episodes)	Success Rate	70%	80%	**90%**	+10%
	Dangerous Rate	100%	100%	**0%**	-100%

6.2 Ablation Study of Resampling Method

To evaluate each component's contribution in SECURE, we conduct ablation studies in simulated domains. In the first ablation study, to examine the importance of averaging the safe actions within the shield, we randomly select a safe action from the batch instead of averaging all safe actions. For the second ablation study, we removed the *adaptive resampling* approach. Instead, we keep resampling with the policy output until a predetermined resampling limit is reached, upon which a random action is selected. The second ablation allows us to assess the effect of not adapting for resampling.

The results of the ablation study are presented in Figure 9, showing the significant impact of *CBF Shield* and the *adaptive resampling*. In the demolition derby domain, SECURE achieves a significant improvement (18.0% and 68.2%) in safety with respect to the two ablations (Kruskal-Wallis $H(2) = 16.25, p < .001$; pairwise posthoc comparisons using Dunn's test indicates SECURE significantly outperforms both ablations with $p < .01$ and $p < .001$, respectively), while maintaining similar or higher task performance. In the Panda Arm Push domain, SECURE eliminates all unsafe executions (Kruskal-Wallis $H(2) = 17.33, p < .001$, DUNN posthoc shows SECURE significantly outperforms both ablations with $p < .01$ and $p < .001$, respectively) as well as achieves a significant task performance gain of 28.2% and 43.8% with respect to the two ablations (Kruskal-Wallis $H(2) = 14.56, p < .001$, Dunn posthoc shows SECURE significantly outperforms both ablations with $p < .01$ and $p < .001$, respectively). These findings validate our design.

6.3 Sensitivity Analysis

Due to the data-driven nature of SECURE, performance can be impacted by the data size and quality. As such, we conduct sensitivity analysis for SECURE from three perspectives: 1) dataset size; 2)

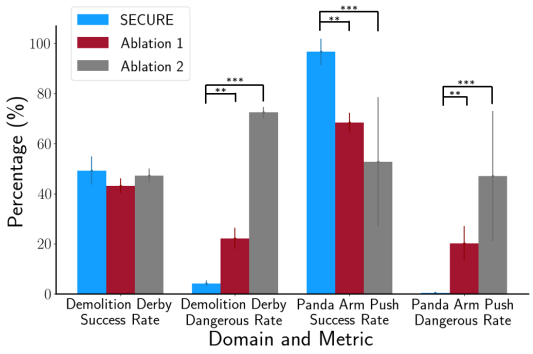


Figure 9: This figure shows the result for the ablation study. The error bars represent standard deviation. ** denotes $p < .01$. * denotes $p < .001$.**

label imbalance; and 3) noisy labels, and show SECURE is robust to non-ideal data.

Dataset Size: In the dataset size sensitivity test, we reduce the overall dataset size for CBF learning while preserving the ratio of safe and unsafe states. We observe SECURE is robust to dataset size in easier tasks, such as Demolition Derby, even with only 1% of the original dataset. The performance drops for harder tasks (e.g., Panda Arm Push) when the dataset size is reduced to 10%.

Label Imbalance: In the label imbalance test, we reduce the number of unsafe states in observance of the relative difficulty in collecting near-dangerous demonstrations. The results demonstrate that SECURE is empirically robust to a data imbalance ratio of 1:2 in Demolition Derby and a ratio of 1:4 in Panda Arm Push. Beyond these ratios, the learned CBF becomes under-conservative due to the overwhelming number of safe states within the dataset.

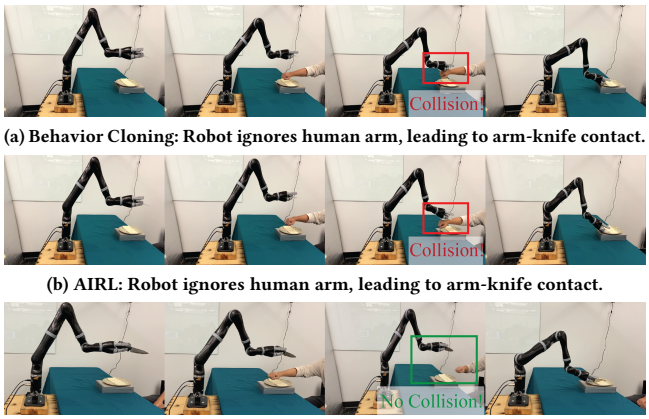
Noisy Data: In the noisy data test, we consider the possible noisy data collection process with naïve user by flipping safe/unsafe labels within the dataset to examine SECURE's robustness. The results show SECURE is robust to noisy data in both domains, exhibiting strong performance even when up to 50% of the labels are wrong.

7 REAL-ROBOT EXPERIMENTS

We conduct two real-robot experiments to demonstrate SECURE's applicability to roboticists and users, respectively. In the first case study, we (roboticists) provide demonstrations for a knife-cutting task and evaluate the success of SECURE in avoiding cutting our arms. In the second user study, we ask users to demonstrate in a coffee placing task and show SECURE's success on users' ratings on task completion, safety, and perceived safety. The number of safe and near-dangerous states for training the CBF for each domain, along with the number of demonstrations used to train the policy, and the size of the neural network CBF are tabulated in Table 2.

7.1 Demonstration with Roboticists

In this demonstration, we compare SECURE with benchmarks in a tofu-cutting task in close proximity to a human. We (roboticists) provide a set of safe demonstrations via kinesthetic teaching. Because of the possible danger the knife may pose, we collect 450 near dangerous states of close proximity of the robot and human arms from experimenters, ensuring they adhere to all necessary safety



(a) Behavior Cloning: Robot ignores human arm, leading to arm-knife contact.

(b) AIRL: Robot ignores human arm, leading to arm-knife contact.

(c) SECURE (ours): Robot yields for human arm, then safely continues.

Figure 10: Timelapse of execution of SECURE and baselines on kitchen cutting task. Unlike baselines, SECURE is able to succesfully finish the task without cutting the nearby human.

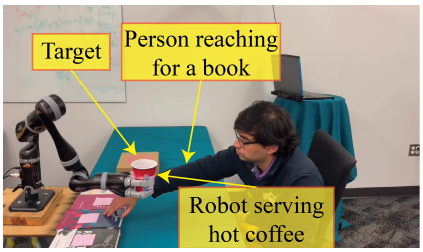


Figure 11: Setup for user study. Robot is tasked to place coffee to pink square, and human is tasked to get a book and turn to certain chapters.

precautions. Following previous CBF literature [35], we assume the robot's forward kinematics model is available.

Similar to the simulated domain experiments, we evaluate SECURE against BC and AIRL with ten episodes and calculate the success rate and dangerous rate metrics. In this cutting task where avoiding collision is of utmost importance, SECURE achieves zero collision cases and 9 successful episodes, surpassing the baseline methods, BC and AIRL (Table 3 and Figure 10). The results demonstrate the safer execution of SECURE, effectively eliminating collisions without compromising task completion. Recordings of SECURE's execution can be found in the supplementary video.

7.2 User Study

We conducted a user study to understand non-roboticist users' abilities to provide helpful demonstrations for SECURE. In this study, we create a context where the user needs to prepare for a lecture by reaching for one out of four books and turning to certain pages, while the robot serves coffee for the user (Figure 11). In the first session of the experiment, human participants first demonstrate how to serve the coffee (i.e., the task) via kinesthetic teaching. The user then provides demonstrations for safe/unsafe human arm positions with respect to the robot and safe/unsafe cup tilt angles. Specifically, to collect safe and unsafe demonstrations, we replay the user's kinesthetic teaching trajectory on the robot, pause at four states,

Table 4: This table shows the task (out of 105), safety (out of 42), and perceived safety (out of 42) ratings in the user study for four conditions. The ratings are reported as mean (standard error). Bold denotes the highest score condition.

Data For	Policy	Individual		Grouped	
	CBF	Individual	Grouped	Individual	Grouped
Metric	Task	73.3 (5.35)	77.1 (4.61)	**81.6** (5.68)	73.6 (5.72)
	Safety	31.3 (3.31)	33.3 (2.74)	**35.4** (2.20)	35.3 (2.48)
	Perceived Safety	33.2 (2.92)	35.4 (2.14)	**36.4** (1.83)	35.8 (2.04)

and invite the participant to provide safe/unsafe demonstrations for arm positions by moving their arm around the robot and for cup tilts by changing the robot end effector tilt angles which is holding the cup. We collect five kinesthetic teaching trajectories and the entire session lasts less than one hour for each participant. As such, we obtain task demonstrations and the user's defined safe/unsafe demonstrations in the first session of the experiment.

Once we finish the demonstration collection in the first session with all participants, we prepare four different setups of data to train SECURE's policy and CBF. In order to see how different components within SECURE respond to amount of data available and whether data is personalized for each user, we consider a 2 by 2 within-subject design with the two factors being policy training data (grouped vs. individual) and CBF training data (grouped vs. individual). The grouped condition represents pooling all participants' data for training, while the individual condition means only using one participant's own data for training. As such, we obtain two behavior-cloning trained policies and two CBFs.

In the second session of the experiment, the participant is tasked to accomplish the task to reach for a book while the robot places the coffee. We test twelve episodes with each participant, with three episodes corresponding to each of the four conditions. After each episode, the participant evaluates the robot's task completion, safety, and perceived safety via a 10-item Likert Scale. We depict the experiment procedure in the supplementary video for a better visual understanding of the setup.

The user study was approved by the Institutional Review Board and we recruited twelve participants (ten male, two female, three within age range 18-25 and seven within age range 26-35). We summarize the results in Table 4. In all four conditions, we demonstrate SECURE successfully accomplishes the task (i.e., coffee placing) while being safe with the human subjects who reach for books and have close interaction with the robot, evidenced by the high ratings in task, safety, and perceived safety. Comparing the four conditions, the grouped policy and individual CBF yields the highest ratings on all three metrics. We hypothesize the result may suggest the utility to learn policy from larger number of task demonstrations as well as the value of personalized training for CBF. Users commented on executions with individual CBF as "P10: exactly how I defined my comfort zone" and "P12: it is not unsafe nor overly safe" compared with their comments regarding grouped CBF as "P7: it felt like the robot was aiming the coffee cup to my face" and "P2: the robot is overly safe - as long as my arm is visible, it tries to avoid me even if there is large distance". However, due to the limited number of subjects in our study, we could not reach a conclusion regarding the performance of grouped vs. individual SECURE without obtaining statistical significance, but we believe our study still demonstrates that that SECURE is successful in the hands of users.

8 DISCUSSION AND LIMITATIONS

The success of SECURE shown in previous sections is grounded in the novel integration of neural CBFs, IRL, and adaptive sampling. SECURE enables the robot to acquire an effective barrier function, which plays a crucial role in shielding the system from dangerous states. By incorporating *CBF Shield*, SECURE ensures that the system remains within a safe state and avoids potential hazards, and that the action executed is in line with the task objective. Furthermore, our *adaptive sampling* increases the efficiency in finding safe actions. Overall, the proposed SECURE method stands out among all the ablations and design choices and presents a promising paradigm for empowering end-users to teach robots new behaviors while maintaining their definition of safety.

SECURE operates under a foundational set of assumptions. SECURE assumes all states within the task demonstrations are safe, which could be invalid if the user provides demonstrations containing undesirable behaviors. Additionally, SECURE assumes that users can provide a collection of undesired states. Nonetheless, we acknowledge that this presumption might not be feasible in certain domains (e.g., autonomous driving, where demonstrating undesirable states could jeopardize human safety). Therefore, the proposed algorithm, SECURE, offers empirical safety assurances rather than absolute safety guarantees. Additionally, SECURE relies on access to the transition dynamics of the domain to assess the safety of proposed actions. We recognize that establishing these transition dynamics in complex domains can present considerable challenges.

In future work, we aim to explore methods to enable active inquiries about uncertain regions, opening up possibilities for proactive learning and further enhancing safety. Another future direction is to investigate user's perception towards grouped vs. individualized policies and safety modules in a larger-scale user study.

9 CONCLUSION

We introduce a novel Safe LfD framework, SECURE, which combines Control Barrier Functions (CBF) with Inverse Reinforcement Learning (IRL) methods to learn a safe policy from demonstrations. By integrating a learned CBF function from human demonstrations, SECURE establishes a *CBF Shield* that ensures the IRL policy avoids unsafe regions. Through empirical evaluations in two simulated domains and two real robot tasks, we demonstrate the effectiveness of SECURE. SECURE achieves comparable or superior task performance compared to traditional IRL methods while significantly reducing the number of unsafe cases.

ACKNOWLEDGMENTS

We wish to thank our reviewers and area chairs for their valuable feedback in revising our manuscript. This work was supported by the National Science Foundation (NSF) under Grant CNS 2219755, the National Institutes of Health (NIH) under Grant 1R01HL157457, Ford Motor Company under Award 003778, and a gift from the Konica Minolta Foundation.

REFERENCES

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained Policy Optimization. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 22–31.

[2] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. 2018. Safe Reinforcement Learning via Shielding. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (Apr. 2018). https://doi.org/10.1609/aaai.v32i1.11797

[3] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. 2019. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*. IEEE, 3420–3431.

[4] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. 2014. Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE Conference on Decision and Control*. IEEE, 6271–6278.

[5] Homanga Bharadhwaj, Aviral Kumar, Nicholas Rhinehart, Sergey Levine, Florian Shkurti, and Animesh Garg. 2021. Conservative Safety Critics for Exploration. In *International Conference on Learning Representations*. https://openreview.net/forum?id=iaO86DUuKi

[6] Roderick Bloem, Bettina Könighofer, Robert Könighofer, and Chao Wang. 2015. Shield Synthesis: Runtime Enforcement for Reactive Systems. In *International Conference on Tools and Algorithms for Construction and Analysis of Systems*.

[7] Serena Booth, W Bradley Knox, Julie Shah, Scott Niekum, Peter Stone, and Alessandro Allievi. 2023. The Perils of Trial-and-Error Reward Design: Misdesign through Overfitting and Invalid Task Specifications. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[8] Daniel Brown, Russell Coleman, Ravi Srinivasan, and Scott Niekum. 2020. Safe imitation learning via fast bayesian reward inference from preferences. In *International Conference on Machine Learning*. PMLR, 1165–1177.

[9] Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. 2022. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. *Annual Review of Control, Robotics, and Autonomous Systems* 5, 1 (2022), 411–444. https://doi.org/10.1146/annurev-control-042920-020211 arXiv:https://doi.org/10.1146/annurev-control-042920-020211

[10] M. Cakmak and L. Takayama. 2013. Towards a comprehensive chore list for domestic robots. In *Proceedings of the International Conference on Human-Robot Interaction (HRI)*. ACM/IEEE, 93–94.

[11] Steven Carr, Nils Jansen, Sebastian Junges, and Ufuk Topcu. 2022. Safe Reinforcement Learning via Shielding under Partial Observability. arXiv:2204.00755 [cs.AI]

[12] Fernando Castañeda, Haruki Nishimura, Rowan Thomas McAllister, Koushil Sreenath, and Adrien Gaidon. 2023. In-Distribution Barrier Functions: Self-Supervised Policy Filters that Avoid Out-of-Distribution States. In *Learning for Dynamics and Control Conference*. PMLR, 286–299.

[13] Letian Chen, Rohan Paleja, Muyleng Ghuy, and Matthew Gombolay. 2020. Joint goal and strategy inference across heterogeneous demonstrators via reward network distillation. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. 659–668.

[14] Letian Chen, Rohan Paleja, and Matthew Gombolay. 2020. Learning from suboptimal demonstration via self-supervised reward regression. *arXiv preprint arXiv:2010.11723* (2020).

[15] Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. 2019. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 3387–3395.

[16] Jason Choi, Fernando Castaneda, Claire J Tomlin, and Koushil Sreenath. 2020. Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions. *arXiv preprint arXiv:2004.07584* (2020).

[17] Glen Chou, Dmitry Berenson, and Necmiye Ozay. 2020. Learning constraints from demonstrations. In *Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics 13*. Springer, 228–245.

[18] Ryan K Cosner, Yuxiao Chen, Karen Leung, and Marco Pavone. 2023. Learning Responsibility Allocations for Safe Human-Robot Interaction with Applications to Autonomous Driving. *arXiv preprint arXiv:2303.03504* (2023).

[19] Ryan K. Cosner, Yisong Yue, and A. Ames. 2022. End-to-End Imitation Learning with Safety Guarantees using Control Barrier Functions. *2022 IEEE 61st Conference on Decision and Control (CDC)* (2022), 5316–5322.

[20] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. 2018. Safe Exploration in Continuous Action Spaces. arXiv:1801.08757 [cs.AI]

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

[22] Quentin Gallouédec, Nicolas Cazin, Emmanuel Dellandréa, and Liming Chen. 2021. panda-gym: Open-Source Goal-Conditioned Environments for Robotic Learning. *4th Robot Learning Workshop: Self-Supervised and Lifelong Learning at NeurIPS* (2021).

[23] Erin Hedlund, Michael Johnson, and Matthew Gombolay. 2021. The Effects of a Robot's Performance on Human Teachers for Learning from Demonstration Tasks. In *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. 207–215.

[24] Przemyslaw A Lasota and Julie A Shah. 2015. Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration. *Human factors* 57, 1 (2015), 21–33.

[25] Karen Leung, Sushant Veer, Edward Schmerling, and Marco Pavone. 2023. Learning Autonomous Vehicle Safety Concepts from Demonstrations. In *2023 American Control Conference (ACC)*. IEEE, 3193–3200.

[26] Lars Lindemann, Haimin Hu, Alexander Robey, Hanwen Zhang, Dimos V Dimarogonas, Stephen Tu, and Nikolai Matni. 2020. Learning hybrid control barrier functions from data. *arXiv preprint arXiv:2011.04112* (2020).

[27] Lars Lindemann, Alexander Robey, Lejun Jiang, Stephen Tu, and N. Matni. 2021. Learning Robust Output Control Barrier Functions from Safe Expert Demonstrations. *ArXiv* abs/2111.09971 (2021).

[28] Yuping Luo and Tengyu Ma. 2021. Learning barrier certificates: Towards safe reinforcement learning with zero training-time violations. *Advances in Neural Information Processing Systems* 34 (2021), 25621–25632.

[29] Haitong Ma, Jianyu Chen, Shengbo Eben, Ziyu Lin, Yang Guan, Yangang Ren, and Sifa Zheng. 2021. Model-based constrained reinforcement learning using generalized control barrier function. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4552–4559.

[30] Zahra Marvi and Bahare Kiumarsi. 2021. Safe reinforcement learning: A control barrier function optimization approach. *International Journal of Robust and Nonlinear Control* 31, 6 (2021), 1923–1940.

[31] David L McPherson, Dexter RR Scobee, Joseph Menke, Allen Y Yang, and S Shankar Sastry. 2018. Modeling supervisor safe sets for improving collaboration in human-robot teams. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 861–868.

[32] David L McPherson, Kaylene C Stocking, and S Shankar Sastry. 2021. Maximum likelihood constraint inference from stochastic demonstrations. In *2021 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 1208–1213.

[33] Teodor Mihai Moldovan and Pieter Abbeel. 2012. Safe Exploration in Markov Decision Processes. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress. http://icml.cc/2012/papers/838.pdf

[34] Tu-Hoa Pham, Giovanni De Magistris, and Ryuki Tachibana. 2018. OptLayer - Practical Constrained Optimization for Deep Reinforcement Learning in the Real World. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 6236–6243. https://doi.org/10.1109/ICRA.2018.8460547

[35] Zengyi Qin, Kaiqing Zhang, Yuxiao Chen, Jingkai Chen, and Chuchu Fan. 2021. Learning safe multi-agent control with decentralized neural barrier certificates. *arXiv preprint arXiv:2101.05436* (2021).

[36] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. 2020. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems* 3 (2020), 297–330.

[37] Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V Dimarogonas, Stephen Tu, and Nikolai Matni. 2020. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 3717–3724.

[38] Maha Salem and Kerstin Dautenhahn. 2015. Evaluating trust and safety in HRI: Practical issues and ethical challenges. *Emerging Policy and Ethics of Human-Robot Interaction* (2015).

[39] Mariah L Schrum, Erin Hedlund-Botti, Nina Moorman, and Matthew C Gombolay. 2022. Mind meld: Personalized meta-learning for robot-centric imitation learning. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 157–165.

[40] Dexter RR Scobee and S Shankar Sastry. 2019. Maximum likelihood constraint inference for inverse reinforcement learning. *arXiv preprint arXiv:1909.05477* (2019).

[41] Krishnan Srinivasan, Benjamin Eysenbach, Sehoon Ha, Jie Tan, and Chelsea Finn. 2020. Learning to be Safe: Deep RL with a Safety Critic. arXiv:2010.14603 [cs.LG]

[42] Mohit Srinivasan, Amogh Dabholkar, Samuel Coogan, and Patricio A Vela. 2020. Synthesis of control barrier functions using a supervised machine learning approach. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 7139–7145.

[43] Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. 2019. Reward Constrained Policy Optimization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SkfrvsA9FX

[44] Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E. Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. 2021. Recovery RL: Safe Reinforcement Learning With Learned Recovery Zones. *IEEE Robotics and Automation Letters* 6, 3 (2021), 4915–4922. https://doi.org/10.1109/LRA.2021.3070252

[45] Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E. Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. 2020. Safety Augmented Value Estimation From Demonstrations (SAVED): Safe Deep

Model-Based RL for Sparse Cost Robotic Tasks. *IEEE Robotics and Automation Letters* 5, 2 (2020), 3612–3619. https://doi.org/10.1109/LRA.2020.2976272

[46] Sanne van Waveren, Rasmus Rudling, Iolanda Leite, Patric Jensfelt, and Christian Pek. 2023. Increasing perceived safety in motion planning for human-drone interaction. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*. 446–455.

[47] Sen Wang, Daoyuan Jia, and Xinshuo Weng. 2018. Deep reinforcement learning for autonomous driving. *arXiv preprint arXiv:1811.11329* (2018).

[48] Zheyuan Wang and Matthew Gombolay. 2020. Heterogeneous Graph Attention Networks for Scalable Multi-Robot Scheduling with Temporospatial Constraints. In *Proceedings of Robotics: Science and Systems (RSS)*.

[49] Douglas J White. 1993. A survey of applications of Markov decision processes. *Journal of the operational research society* 44, 11 (1993), 1073–1096.

[50] Chenlin Zhang, Shaochen Wang, Shaofeng Meng, and Zhen Kan. 2022. Safe Exploration of Reinforcement Learning with Data-Driven Control Barrier Function. In *2022 China Automation Congress (CAC)*. 1008–1013. https://doi.org/10.1109/CAC57257.2022.10055848

[51] Jesse Zhang, Brian Cheung, Chelsea Finn, Sergey Levine, and Dinesh Jayaraman. 2020. Cautious Adaptation For Reinforcement Learning in Safety-Critical Settings. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 11055–11065. https://proceedings.mlr.press/v119/zhang20e.html

Enhancing Safety in Learning from Demonstration Algorithms via Control Barrier Function Shielding (Supplementary)

Yue Yang*
Letian Chen*
Zulfiqar Zaidi*
letian.chen@gatech.edu
Georgia Institute of Technology
Atlanta, GA, USA

Sanne van Waveren
Arjun Krishna
Matthew Gombolay
matthew.gombolay@cc.gatech.edu
Georgia Institute of Technology
Atlanta, GA, USA

ACM Reference Format:
Yue Yang, Letian Chen, Zulfiqar Zaidi, Sanne van Waveren, Arjun Krishna, and Matthew Gombolay. 2024. Enhancing Safety in Learning from Demonstration Algorithms via Control Barrier Function Shielding (Supplementary). In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24), March 11–14, 2024, Boulder, CO, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3610977.3635002

1 VISUALIZATION FOR SECURE'S CBF

The control barrier function $h(\cdot)$ learned in our method plays a crucial role in filtering safe actions. To validate that SECURE can accurately learn a CBF from demonstrations, in the main paper Section 5, we conduct a computational study in a 2D Double Integrator domain. Here, we visualize the learned CBF against the ground-truth CBF on the 2D space of x and y with a fixed $\dot{x}$ and $\dot{y}$ in Figures 1 and 2. It can be seen that SECURE learns a really close approximation of the ground-truth CBF using a small dataset of safe and unsafe states.

We further visualize the learned $h(\cdot)$ for both simulated domains (main paper Section 6). In the demolition derby domain, we visually represent the safety aspects through a heatmap, as illustrated in Figure 3, which exhibits darker regions where $h(s) < 0$, indicating close proximity between the agent and the obstacles. For Panda arm push task, we visualize $h(s)$ in a three-dimensional space. As depicted in Figure 4, the regions situated behind the block are designated as dangerous areas (indicated by the red color). Furthermore, the size of the hazardous region above the block is larger than that below it, as pushing the upper part can result in more unsafe scenarios. The qualitative evidence supports that SECURE is able to learn high-quality CBFs from data.

2 DOMAIN DETAILS

2.1 Demolition Derby Domain

The demolition derby domain is based on a simulation environment introduced in [3], which involves a multi-agent setting where agents navigate towards their respective goals from individual start points. In our adaptation, we designate one agent as our car and the remaining agents as obstacles. The state space comprises the car's position, velocity, and the positions and velocities of the nearest 12 obstacles. The car's actions determine its acceleration. The car starts at the bottom-left corner of the environment, with the goal located at the top-right corner. Demonstrations are collected using the codebase from Qin et al. [3], filtering out collisions to ensure collision-free states in the data. Additionally, we programmatically generate a set of near-dangerous states, denoted as $\mathcal{S}_{nd}$, by collecting 1024 states where the distance between the car and an obstacle falls below a predefined threshold. This dynamic navigation task allows us to effectively evaluate the safety and task completion capabilities of SECURE.

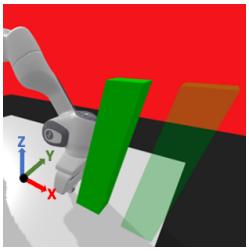
2.2 Panda Arm Pushing Domain

The panda arm pushing domain is simulated using the Panda-Gym environment [1]. The goal is to push a block with a high center of gravity to a target location without causing it to topple. The state space includes the position and velocity of the end-effector, as well as the position, velocity, and angular velocity of the block. Actions are defined as movements of the end-effector in Cartesian coordinates.

To collect demonstrations, we teleoperate the panda arm's end-effector using a keyboard. The program interprets user input of direction and movement size signals, which correspond to specific actions executed by the end-effector. Table 1 provides a summary of the input signals and their meanings. To gather near-dangerous states, we teleoperate the arm and record states where it interacts with the block in specific ways. This includes scenarios such as pushing the upper part of the block, pushing the block with excessive velocity, and pushing the upper part of the block with high velocity, as depicted in Figure 5.

Table 1: Panda Arm pushing domain keyboard teleoperation control signals

Input Signal	Effect
D	Positive X
A	Negative X
Q	Positive Y
E	Negative Y
W	Positive Z
S	Negative Z
1~10	Movement Size



*Authors contributed equally.

HRI '24, March 11–14, 2024, Boulder, CO, USA

ACM ISBN 979-8-4007-0322-5/24/03.
https://doi.org/10.1145/3610977.3635002

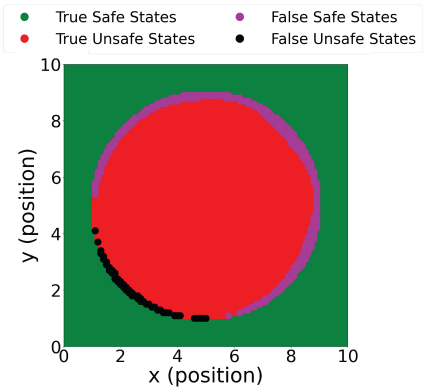


Figure 1: Visualization of the learned CBF in contrast with the ground-truth CBF at $\dot{x} = \dot{y} = 0$.

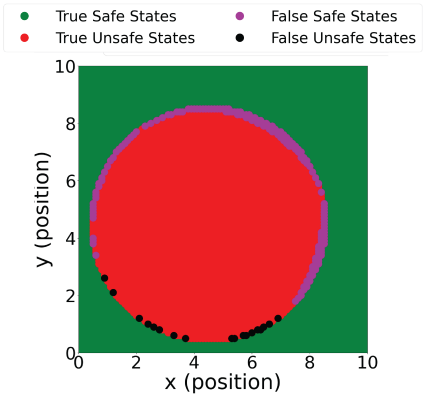


Figure 2: Visualization of the learned CBF in contrast with the ground-truth CBF at $\dot{x} = \dot{y} = 0.5$.

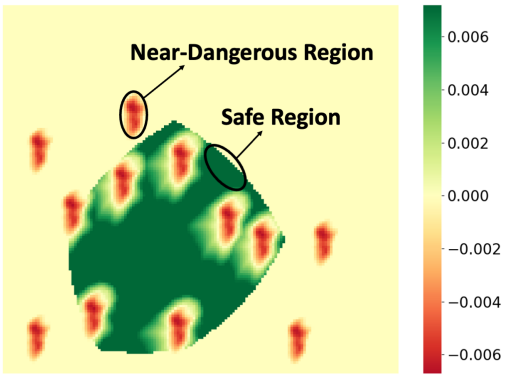


Figure 3: Heatmap visualization of $h(\cdot)$ for the demolition derby task, obtained by fixing the obstacle positions and allowing the agent to explore different states.

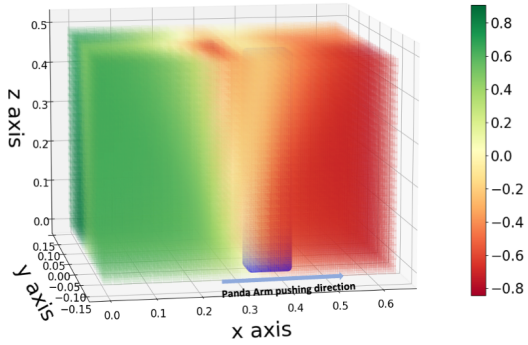


Figure 4: Visualization of $h(\cdot)$ for the Panda Arm Push Task. The learned CBF marks the regions in the upper half of the block as unsafe.

Figure 5: Illustration of near-dangerous scenarios in the Panda Arm Pushing domain.

2.3 Real-Robot Domain

2.3.1 System Setup. For both our real-robot tasks, we utilize a 7 degree-of-freedom Jaco Gen 2 robotic arm from Kinova robotics. The arm is equipped with a gripper holding a knife or a cup, and we incorporate a ZED 2 stereo camera into the setup. Communication between the vision system and the robotic arm, as well as control of the arm, are facilitated through the Robot Operating System (ROS). Control commands are sent to the arm at a frequency of 5 Hz. The state space for this domain encompasses the joint positions of the robot arm, the pose of the cutting knife, the pose of the human arm (if present in the scene), and the position of the tofu sitting on the table top or the target cup position. Actions are defined as changes in each joint's position, and each episode terminates after 200 timesteps.

2.3.2 Vision System. To detect the presence of the human arm in the scene, we mount a ZED 2 stereo camera above the robot's workspace. Calibration of the camera's position with respect to the robot is accomplished using April Tags. Once calibrated, we utilize the built-in body tracking module provided by ZED to track the location of the human arm. The module detects key-points on the human body, and we specifically utilize the wrist and elbow key-points to determine the location of the human forearm. A cylinder is then drawn around the detected position of the arm, as illustrated in Figure 6.

2.3.3 Roboticists Demonstration Collection. For the kitchen cutting task, we employ kinesthetic teaching to obtain the demonstration data. This involved recording five trajectories of the robot arm cutting tofu. To simulate different scenarios, each of these trajectories

Figure 6: Vision system for detecting and estimating the pose of the human arm in the scene.

was replayed twice: once with the human arm present in the scene but not obstructing the robot's path, and once with the human arm in the robot's way. In the latter case, the trajectory playback was paused until the human arm moves out of the way, ensuring safe operation. This data collection process results in ten trajectories in total: five with pauses to accommodate the human arm, and five without pauses as the human arm was not obstructing the robot's movement. To capture a set of potentially dangerous states $\mathcal{S}_{pd}$, we expand upon the recorded trajectories from kinesthetic teaching. We randomly positioned the robot arm within these trajectories, and then manipulate the position of the human arm relative to the knife held by the robot.

2.3.4 Additional Experiments in Tofu-cutting. In addition to the scenarios mentioned in the main text, we conduct two additional experiments to further evaluate the effectiveness of SECURE. Firstly, we evaluate a scenario where the human arm enters the scene but did not obstruct the path of the robot. In such cases, SECURE correctly ignores the presence of the human arm and seamlessly continues the execution of the task, showcasing its ability in personalized safety. Secondly, we evaluate a scenario where the human arm repeatedly obstructs the path of the knife held by the robot. Even in this challenging scenario, SECURE remains effective in ensuring safety and maintaining task execution. The time-lapse of these additional experiments is depicted in Figure 7.

3 FURTHER ABLATION STUDIES

We conduct ablation studies to evaluate the effectiveness of the *CBF Shield* and *Adaptive Resampling* components of SECURE in the two simulated environments.

3.1 Effect of Action Averaging in *CBF Shield*

We examine the impact of not using action averaging inside the *CBF Shield* by randomly selecting one safe action from the batch. Results in the Ablation Study in the main text show that this substitution leads to an increased number of dangerous cases in both simulated domains.

To aggregate safe actions within the action batch, we employ a simple averaging method. In order to showcase the effectiveness of this approach, we compare it against two Q-function-based methods for selecting the ultimate safe action from the available pool.

M-0: Select the action with the highest Q-value among all safe actions, utilizing the Q-function provided by AIRL.

M-1: Sort all safe actions based on their corresponding Q-values and calculate the average of the top r% actions. In our experiments, we set r to 70.

Table 2: Comparison between our method and the two Q-function-based action selection methods. Each method is evaluated on 100 episodes and mean (standard deviation) is reported.

	Success Rate (Stdev)	Dangerous Rate (Stdev)
Ours	**96.6% (5.3%)**	**0.0% (0.0%)**
M-1	29.5% (24.0%)	0.6% (0.9%)
M-2	92.4% (9.9%)	0.1% (0.3%)

We evaluate the performance of our method by comparing it to the two Q-function-based approaches on the panda arm pushing domain. The test results, summarized in Table 2, clearly demonstrate that our method achieves a higher number of successful episodes and a lower number of dangerous cases compared to the Q-function-based methods.

To gain deeper insights into the inferior performance of the Q-function-based methods, we visualize the trajectories of our method (SECURE), M-1, and M-2. For each timestep along the trajectory of our method, we compare the actions selected by M-1 and M-2. We then plot the action values of each method in three dimensions (x, y, and z) over time. Figure 8 illustrates the results, demonstrating that the curves for all three methods exhibit similar patterns in the x and z dimensions. However, in the y dimension, the actions generated by M-1 and M-2 consistently deviate from zero, while our method's actions remain closer to zero. This observation indicates that the learned Q-function introduces a bias on the y dimension in M-1 and M-2. Over time, this bias accumulates and increases the chance of task failure.

3.2 Effect of not Increasing Standard Deviation during Resampling

We study the impact of not increasing the standard deviation (σ) of actions during resampling. In this approach, if the resampling number reaches a maximum threshold and a safety criterion is not met, we randomly select one action. The results in the Ablation Study in the main text show that this substitution leads to an increased number of dangerous cases in both simulated domains, indicating the importance of increasing σ for finding task-aware safe actions.

3.3 Effect of Increasing Standard Deviation along Unsafe Direction

Additionally, we explore the impact of not specifically increasing σ along the unsafe direction according to *Adaptive Resampling*. Instead, we uniformly increase σ across all dimensions, denoted as $\sigma' \leftarrow \sigma + \alpha[\mathbb{1}]^n$. The results in Table 3 show this ablation's comparable number of success cases and dangerous cases to SECURE in the demolition derby domain. This suggests that in a relatively easier learning environment, the interference with task completion

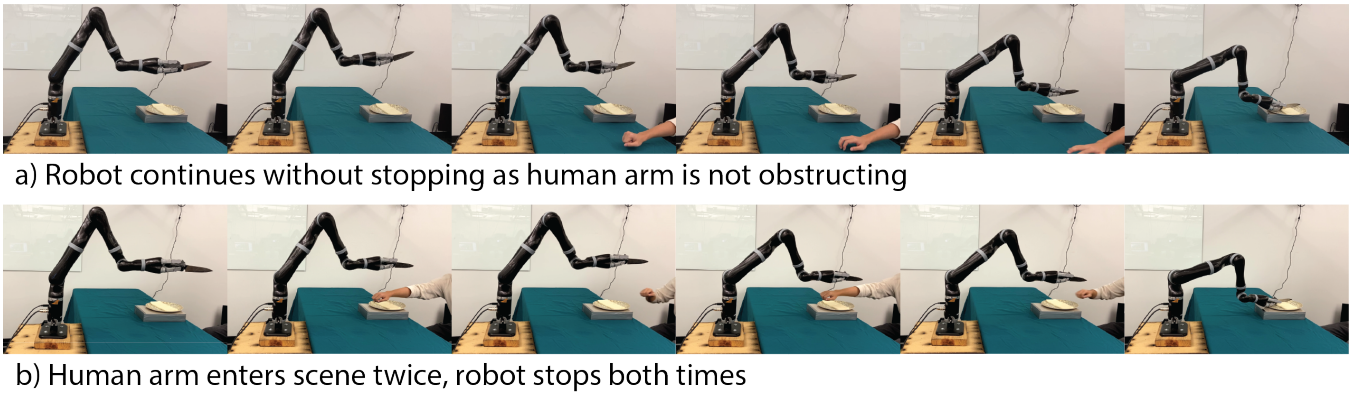


Figure 7: Time-lapse of two scenarios: a) Human arm enters scene without obstruction b) Human arm obstructs robot twice.

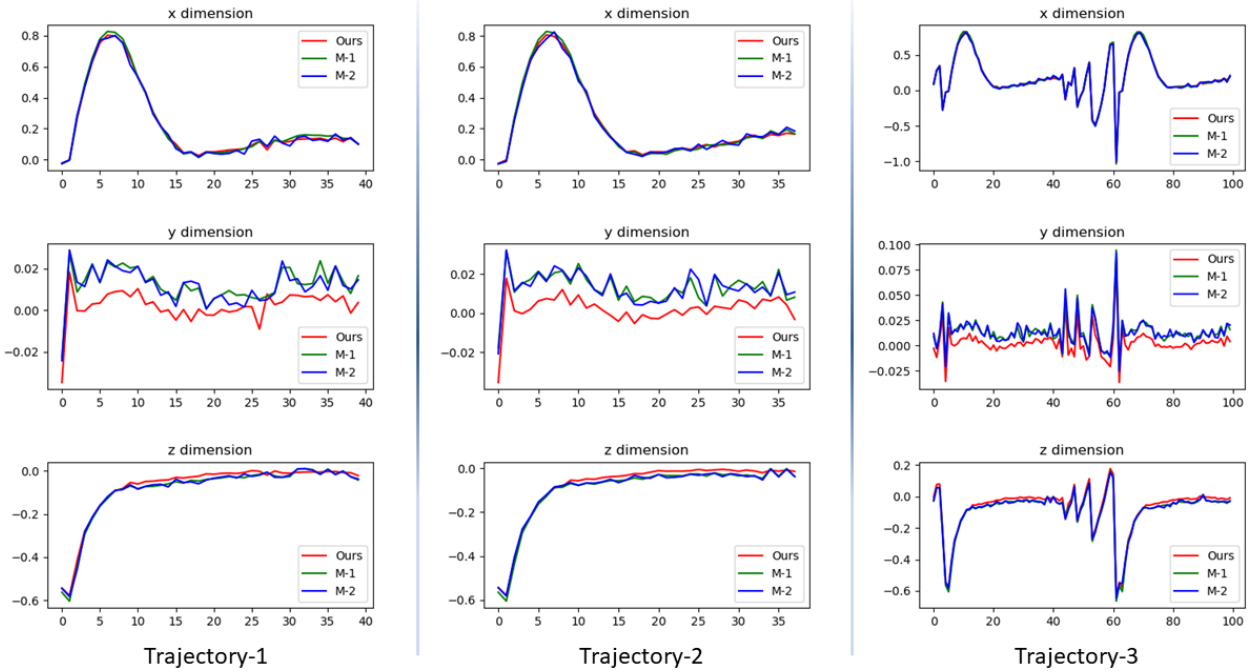


Figure 8: Visualization of action profile in Panda Arm Push domain. Each column represents a trajectory, and each row corresponds to one of the three dimensions (x, y, and z). The horizontal axis denotes the timestep within a trajectory, while the vertical axis represents the action value in the respective dimension.

can be mitigated. However, in the Panda Arm Push domain, this substitution leads to an increased number of dangerous cases. This emphasizes the importance of adaptively increasing σ along the unsafe direction for ensuring safe task completion. For instance, an undesired but safe action, such as moving the end-effector backward instead of forward to push the block, can result in unobserved states that both the learned CBF and AIRL policy are unfamiliar with, leading to unexpected behaviors.

4 HYPERPARAMETERS

Table 4 summarizes the hyperparameters used for training the AIRL policy. Table 5 lists the hyperparameters for training the CBF. The hyperparameters for the *CBF Shield* are provided in Table 6.

5 SENSITIVITY ANALYSIS DETAILS

In this section, we conduct a comprehensive series of experiments to analyze the sensitivity of SECURE to various aspects of demonstration quality and quantity. Our investigation encompasses multiple

Table 3: Results for Increasing σ Uniformly. Each method is evaluated on 100 episodes and mean (standard deviation) is reported.

		SECURE (ours)	Without σ Along Unsafe Direction
Demolition Derby Domain	Success Rate (Stdev)	**49.2%** (5.6%)	45.4% (3.4%)
	Dangerous Rate (Stdev)	**4.2%** (1.2%)	6.4% (3.9%)
Panda Arm Push Domain	Success Rate (Stdev)	**96.6%** (5.3%)	45.8% (28.1%)
	Dangerous Rate (Stdev)	**0.0%** (0.0%)	54.2% (28.1%)

Table 4: Hyperparameters for training AIRL Policy

	Demolition Derby	Panda Arm Push	Kitchen Cutting
Learning Rate	1e-3	1e-3	1e-3
Fusion Number	2000	2000	2000
Max. KL Divergence	0.01	0.0001	0.0001
Discriminator Train Iterations	10	10	10
Generator Train Iterations	10	10	10

Table 5: Hyperparameters for training CBF

	Demolition Derby	Panda Arm Push	Kitchen Cutting
Learning Rate	1e-4	1e-4	1e-4
Batch Size	32	64	64

Table 6: Hyperparameters for *CBF shield*

	Demolition Derby	Panda Arm Push	Kitchen Cutting
ρ_0	0.1	0.1	0.1
Probing Extent	0.01	0.2	0.4

factors, including dataset size, the balance between safe and unsafe states, and the presence of label noises. Through these experiments, we aim to gain insights into the robustness and effectiveness of SECURE's CBF learning. Furthermore, we extend our analysis to explore the data requirements for training SECURE in comparison to the baseline AIRL policy. By examining the number of demonstrations needed for successful training, we provide valuable insights into the efficiency and advantages offered by SECURE's safety-driven learning approach.

5.1 Impact of Dataset Size on the Learned CBF

To assess the impact of dataset size on the learned CBF, we conduct an experiment on the two simulation domains. In the experiment, we reduce the dataset size used for training the CBF, varying from 100% to 1%, and the results are detailed in Table 7. In the demolition derby domain, a reduction in dataset size does not substantially decrease the performance of the learned CBF. Notably, even with a dataset as small as 1% of the original size, only a minor performance drop is observed, indicating the potential effectiveness of CBF learning with limited data in this specific domain.

Table 7: Impact of reduced dataset size on the effectiveness of SECURE

	Demolition Derby		Panda Arm Push	
	Success Rate	**Collision Rate**	**Success Rate**	**Fall Rate**
Reduction Ratio 0.01	46.0% (2.2%)	14.0% (9.1%)	64.3% (14.4%)	15.3% (18.3%)
Reduction Ratio 0.1	42.7% (3.7%)	8.3% (2.9%)	91.0% (11.3%)	8.7% (11.6%)
Reduction Ratio 0.25	44.3% (1.7%)	5.7% (3.1%)	98.7% (0.5%)	1.3% (0.5%)
Reduction Ratio 0.5	45.7% (3.1%)	11.7% (9.5%)	96.3% (1.7%)	1.3% (0.5%)
SECURE	**53.2% (4.1%)**	**3.3% (1.2%)**	**99.3% (0.9%)**	**0.0% (0.0%)**

In contrast, for the panda arm push domain, we observe a performance drop when the dataset size is reduced to 10% of the original. A larger performance decrease happens when the dataset size is further reduced to 1%.

5.2 Impact of Imbalanced Dataset on the Learned CBF

In most practical scenarios, it is often easier to acquire safe states compared to dangerous or near-dangerous states. Consequently, we investigate the influence of dataset imbalance on the learned CBF. By progressively reducing the ratio of unsafe to safe states in the dataset, from 1:1 to 1:10, we evaluate the performance of the learned CBF in two simulated domains. To gauge the learned CBF's effectiveness, we test its predictions on a test set comprising of safe and unsafe states, and we report the over-conservative rate (labeling safe states as unsafe) and under-conservative rate (labeling unsafe states as safe) metrics. Additionally, we report the success and failure rates achieved by SECURE when using the respective CBF. The results are summarized in Table 8.

In the demolition derby domain, the analysis reveals that the learned CBF remains effective up to an imbalance ratio of 1:2. However, beyond this point, the learned CBF's performance drops rapidly (under-conservative rate is 100%, meaning $h > 0$ for all states) due to the imbalanced data, resulting in a jump in collision rate. In the panda arm push domain, the learned CBF's ability to accurately predict safe and unsafe states on the test set diminishes notably as the imbalance ratio reaches 1:10. This decline in learned CBF performance aligns with a marked decrease in the success rate achieved by the policy at an imbalance ratio of 1:10. Consequently, the learned CBF's resilience to dataset imbalance appears to be environment-dependent. Nevertheless, the collective observations from these two domains suggest that the learned CBF is capable of accommodating a notable level of dataset imbalance.

5.3 Impact of Noisy Data on the Learned CBF

We aim to assess the impact of noisy data labels on the efficacy of the learned CBF. To examine this effect, we introduce label noise by flipping the safe/unsafe labels in the dataset by 10%, 25%, and 50%

Table 8: Impact of an imbalanced dataset on the learned CBF and SECURE

	Demolition Derby				Panda Arm Push			
	Success Rate	Collision Rate	Under Conservative	Over Conservative	Success Rate	Fall Rate	Under Conservative	Over Conservative
Imbalance Ratio 1:10	49.6% (4.5%)	73.7% (2.1%)	100%	**37.7%**	35.7% (27.8%)	37.0% (33.7%)	**0%**	37%
Imbalance Ratio 1:4	49.6% (4.5%)	73.7% (2.1%)	100%	**37.7%**	98.7% (0.5%)	1.3% (0.5%)	1%	13%
Imbalance Ratio 1:2	44.3% (1.2%)	5.0% (3.6%)	0%	39.2%	77.0% (19.8%)	1.3% (0.5%)	**0%**	22%
SECURE (1:1)	**53.2% (2.5%)**	**3.3% (1.2%)**	**0%**	**39.0%**	**99.3% (0.9%)**	**0.0% (0.0%)**	0.6%	**11%**

Table 9: Impact of noisy data on learned CBF and SECURE

	Demolition Derby		Panda Arm Push	
	Success Rate	Collision Rate	Success Rate	Fall Rate
Flip Ratio 0.5	42.7% (5.8%)	5.3% (2.5%)	64.7% (42.4%)	3.0% (4.2%)
Flip Ratio 0.25	48.3% (2.1%)	4.0% (1.6%)	32.0% (34.0%)	1.7% (1.7%)
Flip Ratio 0.1	44.0% (4.3%)	8.3% (2.9%)	77.6% (13.4%)	17.0% (17.5%)
SECURE	**52.3% (2.5%)**	**3.3% (1.2%)**	**99.3% (0.9%)**	**0.0% (0.0%)**

Table 10: AIRL's performance in the demolition derby domain across different numbers of demonstrations

	Success Rate
Using 1000 demonstrations	**57%**
Using 500 demonstrations	34%
Using 100 demonstrations	41%
Using 10 demonstrations	29%

for the two simulated domains. The results of this investigation are presented in Table 9. Flipping labels does not have a strong effect on the performance of the learned CBF, particularly evident in the panda arm push domain and thus SECURE demonstrates resilience to noisy labels.

5.4 Number of Demonstrations Required for AIRL

In this experiment, we explore the data requirements for training SECURE in comparison to the baseline AIRL policy. We focus on demolition derby domain and assess the success rate achieved by AIRL using different numbers of demonstrations episodes: 10, 100, 500, and 1000. The results are summarized in Table 10. The results show that SECURE does not require additional demonstrations than what is required for training an AIRL policy, which underscores SECURE's ability to maintain high policy performance without imposing the need for a larger dataset.

6 OTHER DESIGN CHOICES IN OPTIMIZING CBF REQUIREMENTS

We conduct a set of supplementary experiments to affirm the validity of our design decision to independently learn the CBF h and policy π and present supplementary metrics that highlight SECURE's enhanced ability to produce a safer policy in comparison to the baseline AIRL method.

6.1 Joint Training of CBF and Policy

In this section, we substantiate our decision to independently train the CBF h and the policy π within the SECURE framework. We compare the performance of SECURE with an alternative approach that jointly optimizes both the CBF h and the policy π in the demolition derby domain. The result of this comparison is presented in Table 11. We observe a significant contrast in collision rates: when jointly optimizing the barrier function and policy, the collision rate dramatically increases to 77.7%, in contrast to the 3.3% collision rate achieved by SECURE. We posit that this discrepancy arises due to the inherent challenge of tuning the relative weights between the LfD objective and the safety objective for the policy. Introducing a dynamic CBF function h into this learning process introduces additional complexity, further contributing to the instability of the learning dynamics and hindering the learning of a robust policy.

Table 11: Effect of jointly optimizing CBF h and policy π in demolition derby domain

	Optimize jointly	AIRL	SECURE
Success Rate	32.3% (11.0%)	49.3% (6.1%)	**52.3% (2.5%)**
Collision Rate	77.7% (3.4%)	72.3% (0.5%)	**3.3% (1.2%)**

6.2 Gradient Ascent to Find Safe Action

In this section, we perform an additional experiment to assess the efficacy of our proposed resampling approach. Rather than employing our adaptive resampling technique, we substitute it with an alternate method: optimizing for the closest safe action when the policy output is deemed unsafe, achieved through gradient ascent on the learned CBF. For the demolition derby domain, we present the outcomes of this ablation experiment in Table 12. We observe that this alternative approach exhibits inferior performance compared to SECURE across both evaluation metrics. We hypothesize this is due to local optimum existing in the learned CBF and the gradient ascent approach is stuck. This finding reaffirms the significance of SECURE's safe-action-batch averaging operation, underscoring its role in achieving superior performance.

6.3 Additional Safety Metrics for SECURE

We extend our evaluation to incorporate additional safety metrics, assessed across 100 trajectories in both the demolition derby and panda arm push domains. In the demolition derby domain, we compute the minimum distance of the agent to an obstacle along its trajectory, serving as an indicator of potential safety concerns. For the panda arm push domain, we measure the maximum angle of the block, with a higher angle indicating an elevated risk of block instability.

The outcomes of this expanded assessment are presented in Table 13. Notably, SECURE consistently demonstrates a greater "minimum distance to obstacles" and a reduced "maximum fall down angle" in comparison to AIRL. These findings reaffirm our

Table 12: Effect of using gradient ascent to find closest safe action in demolition derby domain

	Demolition Derby	
	Success Rate	**Collision Rate**
Gradient Ascent	7.7% (8.7%)	81.0% (2.9%)
AIRL	49.3% (6.1%)	72.3% (0.5%)
SECURE	**52.3% (2.5%)**	**3.3% (1.2%)**

Table 13: Comparative safety metrics of AIRL and SECURE policies in the demolition derby and panda arm push domains

	AIRL	**SECURE**
Demolition Derby (minimum distance to obstacles)	0.004 (0.001)	**0.013 (0.001)**
Panda Arm Push (maximum fall down angle)	0.34 (0.03)	**0.06 (0.03)**

claim that SECURE consistently produces a safer policy than the baseline AIRL approach.

7 CBF NEURAL NETWORK

We empirically choose different Neural Networks to represent the CBF across various domains. In Demolition Derby, we employ a 4-layer 1D CNN, akin to [3], configured with respective numbers of output filters set as [64, 128, 64, 1]. In Panda Arm Push and Kitchen Cutting, we utilize a GaussianMLP [2], a model represented by a Gaussian distribution that is parameterized by a multilayer perceptron, with hidden layers of size [32, 128, 128, 256, 256, 256, 256, 128, 128, 32]. In Coffee Serving, we also use the GaussianMLP, but with hidden layers of size [64, 64].

8 SCALABILITY OF CBF LEARNING FROM DEMONSTRATION

As shown in Table 3-4 of the main paper, SECURE's CBF learning generally requires 1000 safe and unsafe states across domains of varying complexity. In our user study, this takes 45-60 minutes for each user. Importantly, the IRL itself requires more demonstrations than SECURE's CBF learning. We hypothesize that a more complex domain or safety specification may require more data for CBF learning but is likely to have a minor effect compared to the number of demonstrations required by IRL algorithms.

9 COMPUTATIONAL COST AND TIME TO FIND SAFE ACTIONS

SECURE finds safe action batches that exceed ρ_0 safe action rate with a 100% success rate for the Demolition Derby and Panda Arm domains, requiring only an average time of 0.061s (standard deviation: 0.162s) and 0.077s (standard deviation: 0.026s) of computation per action for the Demolition Derby and Panda Arm domains, respectively, on an AMD Ryzen 9 5900.

The worst-case computational cost of one action is $O(K(t_1 + t_2 + M))$ where K is the number of retries in finding safe actions, M is the CBF Shield sampling batch size, and t_1 and t_2 are the time complexity for a forward pass of the policy network and the CBF network, respectively. We note that the forward pass of CBF for a batch in the Adaptive Resampling procedure is parallelizable and, thus, has a constant cost for computation.

REFERENCES

[1] Quentin Gallouédec, Nicolas Cazin, Emmanuel Dellandréa, and Liming Chen. 2021. panda-gym: Open-Source Goal-Conditioned Environments for Robotic Learning. *4th Robot Learning Workshop: Self-Supervised and Lifelong Learning at NeurIPS* (2021).

[2] The garage contributors. 2019. Garage: A toolkit for reproducible reinforcement learning research. https://github.com/rlworkgroup/garage.

[3] Zengyi Qin, Kaiqing Zhang, Yuxiao Chen, Jingkai Chen, and Chuchu Fan. 2021. Learning safe multi-agent control with decentralized neural barrier certificates. *arXiv preprint arXiv:2101.05436* (2021).