Learning safety critics via a non-contractive binary Bellman operator

Agustin Castellano
Johns Hopkins University
Baltimore, MD, USA
acastell@jhu.edu

Hancheng Min
University of Pennsylvania
Philadelphia, PA, USA
hanchmin@seas.upenn.edu

Juan Andrés Bazerque
University of Pittsburgh
Pittsburgh, PA, USA
juanbazerque@pitt.edu

Enrique Mallada
Johns Hopkins University
Baltimore, MD, USA
mallada@jhu.edu

Abstract—**The inability to naturally enforce safety in Reinforcement Learning (RL), with limited failures, is a core challenge impeding its use in real-world applications. One notion of safety of vast practical relevance is the ability to avoid (unsafe) regions of the state space. Though such a safety goal can be captured by an action-value-like function, a.k.a. safety critics, the associated operator lacks the desired contraction and uniqueness properties that the classical Bellman operator enjoys. In this work, we overcome the non-contractiveness of safety critic operators by leveraging that safety is a binary property. To that end, we study the properties of the binary safety critic associated with a deterministic dynamical system that seeks to avoid reaching an unsafe region. We formulate the corresponding binary Bellman equation (B2E) for safety and study its properties. While the resulting operator is still non-contractive, we fully characterize its fixed points representing–except for a spurious solution–maximal persistently safe regions of the state space that can always avoid failure. We provide an algorithm that, by design, leverages axiomatic knowledge of safe data to avoid spurious fixed points.**

Index Terms—**safety-critical systems, reinforcement learning, safe reinforcement learning, reachability theory**

I. Introduction

The last decade has witnessed a resurgence of Reinforcement Learning (RL) as a core enabler of Artificial Intelligence (AI). Today, RL algorithms can provide astonishing demonstrations of super-human performance in multiple settings, such as Atari [1], Go [2], StarCraft II [3], and even poker [4]. However, this super-human success in RL is overwhelmingly limited to *virtual domains* (games in particular), where not only one has a vast amount of data and computational power, but also there is little consequence to failure in achieving a task. Unfortunately, physical domain applications (autonomous driving, robotics, personalized medicine) lack most of these qualities and are particularly sensitive to scenarios where the consequences of poor decision-making are catastrophic [5],[6].

Guaranteeing safety in an RL setting is challenging, as agents often lack a priori knowledge of the safety of states and actions [7]. Inspired by these challenges, numerous methods have been proposed to imbue RL methods with safety constraints, including expectation constraints [8, 9], probabilistic/conditional value at risk constraints [10, 11], and stability constraints [12, 13], among others. Such methods naturally lead to different safety guarantees, some of which can be theoretically characterized [14, 15]. However, most of these methods fail to capture the safety-critical nature of some events that must be avoided at all costs, i.e., with probability one.

One type of safety constraint of practical relevance in safety-critical applications is reachability constraints (e.g.[16]; [17, Ch. 3]; [18]), wherein one seeks to avoid regions of the state space that are associated with failure events by computing sets that are either, persistently safe (a.k.a. control invariant safe sets [19]), i.e., regions of the state that can avoid failure regions *for all times* by proper choice of actions, or unsafe regions (a.k.a. as backward reachable tubes [20]) where *failure is unavoidable* irrespectively of the actions taken. Recent research efforts incorporating such constraints in RL problems have shaped the notion of safety critics [21, 22, 23], which aim to compute action value-like functions that, based on information about either the (signed) distance to failure or a logical fail/not fail feedback, predict whether a state-action pair is safe to take or is doomed to catastrophic failure.

Unfortunately, the computation and learning of safety critics is a challenging task since their corresponding Bellman-like equations (and associated operators) lack typical uniqueness (resp. contraction) properties that guarantee the validity of the solution (and convergence of RL algorithms). As a result, most works seek to compute approximate safety critics by introducing an artificial discount factor [21, 24]. This approximation, however, can drastically affect the accuracy of the critic, as approximately safe sets are not, by design, safe.

a) Contributions of our work: In this work, we seek to overcome the difficulties in computing accurate safety critics by developing supporting theory and algorithms that allow us to learn accurate safety critics directly from the original non-contractive safety critic operator. Precisely, we consider a setting with deterministic, continuous state dynamics that are driven by stochastic policies on discrete action spaces, and model safety as a binary (`safe` / `unsafe`) quantity. Building on the literature of risk-based safety critics, we develop a deeper theoretical understanding of the properties of the corresponding *binary safety (action-)value function* and how to exploit them to learn accurate safety critics. In doing so, we make the following contributions.

- **Characterization of solutions to the binary Bellman equations for safety** We study the properties of the *action-*

This work was supported by NSF through grants CAREER 1752362, CPS 2136324, Global Centers 2330450.

value function associated with the binary safe/unsafe feedback and formulate a *binary Bellman equation* (B2E) that such function must satisfy. This B2E is undiscounted and has a non-contractive operator with multiple fixed points. Nevertheless, we show (Theorem 1) that all (but one) of the possibly infinite solutions to the B2E represent regions of the state space that are: *(i)* persistently safe regions that can avoid failure for all future times and *(ii)* maximal, in the sense that no state that is declared to be unsafe can reach the declared safe region.

- **Algorithm for learning fixed points of a non-contractive operator** Finally, we provide an algorithm that can find a fixed point of the non-contractive operator, despite the lack of contraction. Our algorithm has two distinctive features that make this possible. First, it uses *axiomatic data points*, i.e., points of the state space that are a priori known to be safe. Secondly, it uses a classification loss that enforces *self-consistency* of the Bellman equation across samples. Preliminary numerical evaluations indicate that our proposed methodology outperforms a well-known safety critic [21] in a simple setup.

II. Problem Formulation

a) Environment: We consider a Markov Decision Process $\langle \mathcal{S}, \mathcal{A}, F, \mathcal{G}, i, \rho \rangle$ where the state space $\mathcal{S}$ is *continuous* and compact, the action space $\mathcal{A}$ is *discrete* and finite, the map $F : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a *deterministic* transition function. The set $\mathcal{G}$ represents a set of "failure" states to be avoided. At each time step, the agent receives as feedback the *insecurity* of state s_t, that is $i(s_t) = \mathbb{1}\{s_t \in \mathcal{G}\} \in \{0, 1\}$. Episodes start at a state $s_0 \sim \rho$ and run indefinitely or end when the system enters $\mathcal{G}$.

b) Policies: We consider stochastic, stationary policies $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ in the simplex $\Delta_{\mathcal{A}}$, and denote $\pi(a|s)$ the probability of $a \in \mathcal{A}$ when at state $s \in \mathcal{S}$. With discrete and finite $\mathcal{A}$ and deterministic transition dynamics, the set of reachable states starting from any s is finite, as defined next

Definition 1 (t-step reachable sets): For any policy π and any state $s \in \mathcal{S}$, the t-step reachable set from s under π is $\mathcal{F}^{\pi}_t(s) \triangleq \big\{s' \in \mathcal{S} : \mathbb{P}^{\pi}\left(s_t = s' \mid s_0 = s\right) > 0\big\}$. Similarly, for any $a \in \mathcal{A}$ we define $\mathcal{F}^{\pi}_t(s, a) \triangleq \big\{s' \in \mathcal{S} : \mathbb{P}^{\pi}\left(s_t = s' \mid s_0 = s, a_0 = a\right) > 0\big\}$.

Given these notions of reachable sets, we can define the binary safety value functions for any policy.

Definition 2 (Binary safety value functions): The binary safety (action-)value function of policy π at state s (and action a) is:

$$v^{\pi}(s) \triangleq \sup_{t \geq 0} \max_{s_t \in \mathcal{F}^{\pi}_t(s)} i(s_t)\,, \tag{1}$$

$$b^{\pi}(s, a) \triangleq \sup_{t \geq 0} \max_{s_t \in \mathcal{F}^{\pi}_t(s,a)} i(s_t)\,. \tag{2}$$

We choose the notation $b(\cdot, \cdot)$ instead of the usual Q to emphasize that it is a binary action-value function. Note that $b^{\pi}(s, a) = 1$ if and only if starting from (s, a) and following π, there is a positive probability of entering $\mathcal{G}$. The optimal (action-)value functions are then defined.

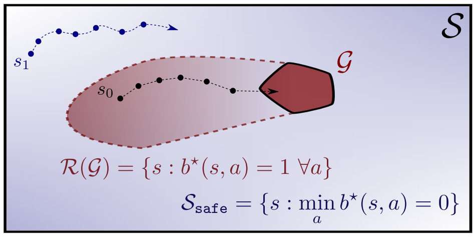
$\mathcal{S}$
s_1
$\mathcal{G}$
s_0
$\mathcal{R}(\mathcal{G}) = \{s : b^{\star}(s,a) = 1\ \forall a\}$
$\mathcal{S}_{\mathtt{safe}} = \{s : \min_a b^{\star}(s,a) = 0\}$

Fig. 1: The optimal $b^{\star}$ describes different regions of the state space. The set $\mathcal{G}$ (solid red) is to be avoided at all times. Due to system dynamics, there is a region of the state space $\mathcal{R}(\mathcal{G})$ (shaded red) such that any trajectory starting there (e.g., from s_0) will inevitably enter $\mathcal{G}$. For any point in its complement $\mathcal{S}_{\mathtt{safe}}$ (e.g. s_1), the optimal policy avoids $\mathcal{G}$ at all times.

Definition 3 (Optimal binary value functions): For all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, the optimal value and action-value functions are $v^{\star}(s) \triangleq \min_{\pi} v^{\pi}(s)$ and $b^{\star}(s, a) \triangleq \min_{\pi} b^{\pi}(s, a)$.

c) Relationship between safety and the optimal binary functions: These optimal value functions fully characterize the logical `safe/unsafe` nature of each state or state-action pair, and have nice interpretations in terms of how they partition the state-space, as illustrated in Fig. 1. Recall that the safety goal is to avoid $\mathcal{G}$. However, due to the MDP dynamics, this might not be possible for every state outside $\mathcal{G}$[1]. A state s is persistently safe if trajectories from s can avoid $\mathcal{G}$ at all times—in other words, if $\exists a \in \mathcal{A} : b^{\star}(s, a) = 0$. Conversely, a state s is doomed to fail if $b^{\star}(s, a) = 1\ \forall a \in \mathcal{A}$. We use $\mathcal{R}(\mathcal{G})$ to denote this set of "unsafe states" that are doomed to fail. The complement of this set is the set of persistently safe states, and the "safe" actions for each state are given by:

$$\mathcal{S}_{\mathtt{safe}} = \Big\{s \in \mathcal{S} : \min_{a \in \mathcal{A}} b^{\star}(s, a) = 0\Big\} \tag{3}$$

$$\mathcal{A}_{\mathtt{safe}}(s) = \big\{a \in \mathcal{A} : b^{\star}(s, a) = 0\big\}. \tag{4}$$

Just like in the standard RL setup, each (action-)value function has associated Bellman equations.

Proposition 1 (Binary Bellman Equations): For any policy π, the following set of Bellman equations hold for all $s \in \mathcal{S}$, for all $a \in \mathcal{A}$: $b^{\pi}(s, a) = i(s) + \big(1 - i(s)\big)v^{\pi}(s')$, where $s' = F(s, a)$. In particular, any optimal policy satisfies:

$$b^{\star}(s, a) = i(s) + \big(1 - i(s)\big) \min_{a' \in \mathcal{A}} b^{\star}(s', a'). \tag{5}$$

Proof: See Appendix B. □

d) Unsafety as a logical `OR`*:* The Bellman equation for the optimal $b^{\star}$ can be understood as: "an (s, a) pair is *unsafe* ($b^{\star}(s, a) = 1$) if either: the current state is unsafe ($i(s) = 1$), `OR` it leads to an unsafe state later in the future ($\min_{a'} b^{\star}(s', a') = 1$)."

e) Non-contractive Bellman operator: The optimal binary function of (5) has an associated operator, acting on the space of functions $\mathcal{B} = \{b : \mathcal{S} \times \mathcal{A} \to \{0, 1\}\}$, $\mathcal{T} : \mathcal{B} \to \mathcal{B}$ s.t.

$$(\mathcal{T}b)(s, a) = i(s) + (1 - i(s)) \min_{a' \in \mathcal{A}} b(s', a') \quad \forall (s, a) \tag{6}$$

[1] A car heading to a wall ($\mathcal{G}$) one meter away at 100mph will hit it, regardless of the actions taken.

One of the key features in the standard (discounted) Bellman equations for infinite-horizon problems is that it has an associated operator that is contractive [25, p.45]. As such, it admits a unique fixed point (the optimal value function). This is crucial for applying value iteration procedures or for methods reliant on temporal differences [26]. Surprisingly, the operator defined in (6) is non-contractive, and as such, it admits more fixed points than the optimal $b^{\star}$. In the next section we will see that all but one of these fixed points possess the desired safety properties.

A. Closely Related Work

a) Control-theoretic approaches for computing $\mathcal{S}_{\texttt{safe}}$: Standard tools from Control Theory exist to approximate the safe regions corresponding to $b^{\star}(\cdot,\cdot)$, both for linear [27] and non-linear dynamics [28]. The latter requires knowledge of the transition map $F(\cdot,\cdot)$ along with the signed distance to the unsafe region [29]. This amounts to solving partial differential equations (PDEs) of the Hamilton-Jacobi-Isaacs (HJI) type [18], and yields value functions whose zero super-level sets correspond to $\mathcal{S}_{\texttt{safe}}$.

b) Risk-based vs Reachability-based safety critics: The binary action-value function $b^{\star}$ defined in (5) is closely related to recent work on *Risk-based* safety critics [22, 23], which use binary information to indicate the risk of unsafe events. However, unlike risk-based critics, which seek to measure a cumulative expected risk $b^{\star}_{\text{risk}}(s,a) = \max_{\pi} E_{\pi}[\sum_{k=t}^{\infty}\gamma^{k-t}i(s_t)|s_t = s, a_t = a] \in [0,1]$, our binary critic only takes values $b^{*}(s,a) \in \{0,1\}$, and outputting 1 whenever unavoidable failure has positive probability. *Reachability based* safety critics, build on the literature of HJI equations and seek to measure the largest (signed) distance $h(s_t)$ that one can sustain from the failure set $\mathcal{G}$, i.e., $b^{\star}_{\text{reach}}(s,a) = \sup_{\pi}\inf_{t\geq 0} h(s_t)$ almost surely [21]. Our binary critic $b^{\star}$ is indeed related to $b^{\star}_{\text{reach}}$ when the signed distance $h(s)$ is replaced with the binary signal $-i(s)$. We will soon show that this particular choice of safety measure allows for a precise characterization of the fixed points of (6).

c) To contract or not to contract: Despite the diversity of safety critics present in the literature, a common practice in both risk-based critics [22, 23] and reachability-based critics [21, 30] is the introduction of a discount factor $\gamma < 1$. While this leads to desired uniqueness and contraction properties for the operator, it comes with trade-offs: it degrades the accuracy, requiring the introduction of conservative thresholds [22, 30], which further limits exploration. Notably, such an approach is particularly problematic when seeking to guarantee persistent safety (the ability to avoid failure for all future times), as such property is not preserved for finite accuracy approximation, even for thresholded ones. In this work, we overcome this limitation by seeking to learn directly using the non-contractive operator, thus guaranteeing, by design, the correctness of the solution.

III. Binary characterization of safety

The fixed points $b^{\star}$ of the binary Bellman operator have a meaningful interpretation in terms of the topology of the state-space and can be used to derive persistently safe policies. This connection will be better understood once we define the notion of control invariant safe sets.

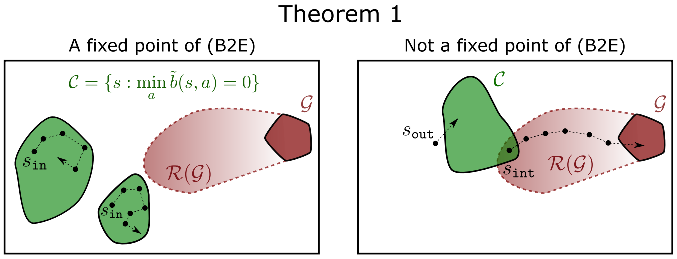

Fig. 2: An illustration of Theorem 1. Left: a valid fixed point $\tilde{b}$ of $\mathcal{T}$ and its corresponding safe control invariant set. Trajectories starting in $\mathcal{C}$ can be driven to remain in $\mathcal{C}$. Right: a function $\tilde{b}$ that is not a fixed point. A state $s_{\texttt{int}}$ in the intersection will inevitably lead to the unsafe region $\mathcal{G}$, so $\tilde{b}(s,a)$ should be 1 for all states in the trajectory (which would mean $s_{\texttt{int}} \notin \mathcal{C}$). Similarly, a state $s_{\texttt{out}}$ outside $\mathcal{C}$ cannot reach inside. If it could, $\tilde{b}(s_{\texttt{out}},a) = 1$ for some $a \in \mathcal{A}$, but it would transition to a state where $\min_{a'} \tilde{b}(s',a') = 0$, violating (5).

Definition 4 (Control invariant safe (CIS) set): A set $\mathcal{C} \subset \mathcal{S}$ is a control invariant safe (CIS) set if there exists a policy π such that:

i) (Control invariance): $\forall s_0 \in \mathcal{C}, \forall t \geq 0, \quad \mathcal{F}^{\pi}_{t}(s_0) \subset \mathcal{C}$
ii) (Safety): $\forall s_0 \in \mathcal{C}, \forall t \geq 0, \quad \mathcal{F}^{\pi}_{t}(s_0) \cap \mathcal{G} = \emptyset$.

In essence, *(i)* means that there exists a controller that guarantees that trajectories starting in $\mathcal{C}$ can be made to remain in $\mathcal{C}$ forever, which is a standard notion in control theory [16, 31]. Property *(ii)* means this can be done while also avoiding the unsafe region $\mathcal{G}$!

With these definitions in place, we are ready for our main result.

Theorem 1 (Fixed points and control invariant safe sets): Let $\tilde{b} : \mathcal{S} \times \mathcal{A} \to \{0,1\}$ be a fixed point of (6). Then either $\tilde{b}(s,a) = 1 \ \forall (s,a)$ (spurious fixed point), or:

i) $\mathcal{C} \triangleq \left\{s \in \mathcal{S} : \min_a \tilde{b}(s,a) = 0\right\}$ is control invariant safe (CIS).
ii) $\mathcal{C}$ is unreachable from outside, i.e., $\mathcal{F}^{\pi}_{t}(s_0) \cap \mathcal{C} = \emptyset \quad \forall s_0 \in \mathcal{S} \setminus \mathcal{C}, \forall \pi, \forall t \geq 0$.
iii) Any policy π that satisfies (7) renders $\mathcal{C}$ CIS.

$$\tilde{b}(s,a) = 1 \ \Rightarrow \pi(a|s) = 0, \ \forall s \in \mathcal{C}. \tag{7}$$

Proof: The proof is in Appendix C. □

The first statement proclaims that starting in $\mathcal{C}$, the system can be made to remain in $\mathcal{C}$ forever (thus ensuring safety). The contrapositive of property *(ii)* sheds light on a notion of *maximality* of $\mathcal{C}$:

Corollary 1 (Maximality of the CIS set): Let $\mathcal{X}$ be a strict subset of $\mathcal{C}$. If $\mathcal{X}$ is reachable[2] from $\mathcal{C} \setminus \mathcal{X}$, then $\mathcal{X}$ cannot be associated[3] with any fixed point of (6).

[2] i.e. if $\exists \pi, \exists t \geq 0, \exists s_0 \in \mathcal{C} \setminus \mathcal{X} : \mathcal{F}^{\pi}_{t}(s_0) \cap \mathcal{X} \neq \emptyset$

[3] that is to say: $\forall \tilde{b} : \tilde{b} = \mathcal{T}\tilde{b}, \mathcal{X} \neq \left\{s \in \mathcal{S} : \min_a \tilde{b}(s,a) = 0\right\}$

We refer the reader to Fig. 2 for an illustration of valid and nonvalid fixed points. By means of Theorem 1 and Corollary 1, we achieve our goal of identifying the fixed points of the binary Bellman operator to maximal persistently safe states. In the following section, we will present an algorithm that finds fixed points of $\mathcal{T}$.

Algorithm 1: Pseudocode for learning the binary value function

```
   Input: Safe dataset $\mathcal{D}_{\texttt{safe}}$;
   Output: Binary value function $b^\theta(\cdot,\cdot)$;
 1 Initialize $b^\theta(\cdot,\cdot)$ using $\mathcal{D}_{\texttt{safe}}$ and $\mathcal{M} = [\ ]$ ;
     ▷ Transition buffer.
 2 repeat
 3     for i=0,... NUM_EPISODES-1 do
 4         Run episodes, store $\left(s_k, a_k, i(s_k), s'_k\right)_{k=1}^{K}$
             transitions in $\mathcal{M}$;
 5     end
 6     $\mathcal{D}_{\texttt{unsafe}} \leftarrow$ build_unsafe_dataset($b^\theta, \mathcal{M}$) ;
         ▷ Use $b^\theta$ to compute labels.
 7     Build $\mathcal{D} = \mathcal{D}_{\texttt{safe}} \cup \mathcal{D}_{\texttt{unsafe}}$ ;        ▷ Complete
         dataset.
 8     repeat
 9         Run gradient steps on $\mathcal{L}_{\texttt{train}}$ ;  ▷ Update $b^\theta$
10     until Accuracy($b^\theta, \mathcal{D}$) $= 1$;
11     $\mathcal{D}_{\texttt{unsafe}} \leftarrow$ build_unsafe_dataset($b_i, \mathcal{M}$) ;
         ▷ $b^\theta$ has changed w.r.t.6
12     Build $\mathcal{D} = \mathcal{D}_{\texttt{safe}} \cup \mathcal{D}_{\texttt{unsafe}}$ ;  ▷ New dataset
13     if Accuracy($b^\theta, \mathcal{D}$) $\neq 1$;                ▷ Check
         consistency of B2E
14      then
15         go to Line 8 ;  ▷ Not self-consistent
             ⇒ Re-train the network
16     end
17 until termination;
```

IV. Algorithm

We propose training a neural network classifier to learn fixed points of $\mathcal{T}$ in (6). We will denote the learned function by $b^\theta(\cdot,\cdot)$ where θ collects the network parameters. The network takes each state as input and outputs the value $b^\theta(s,a)$ for each possible action. The last layer is a point-wise sigmoid activation function ensuring $b^\theta(s,a)$ lies in the unit interval. We use $\hat{b}^\theta(s,a) \triangleq \texttt{Round}\left(b^\theta(s,a)\right)$ to denote the predicted label. Note that our threshold (at 1/2) will be fixed during training and testing. The pseudocode for the main algorithm can be found in Alg. 1. We provide a comprehensive breakdown of its main components next.

a) Dataset: The dataset $\mathcal{D}$ consists of (s,a,y) tuples, where y is a $\{0,1\}$ label, and has two components. A prescribed [4] safe set $\mathcal{D}_{\texttt{safe}}$ (for which $y=0$) and a dynamically updated $\mathcal{D}_{\texttt{unsafe}}$ of unsafe transitions detected during data collection. We have observed empirically that the addition of $\mathcal{D}_{\texttt{safe}}$ helps prevent the resulting binary value function from collapsing into the trivial fixed point described in Theorem 1. The algorithm iterates over the following three loops:

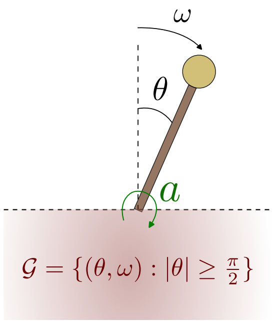
ω
θ
a
$\mathcal{G} = \{(\theta,\omega) : |\theta| \geq \frac{\pi}{2}\}$

Fig. 3: The custom inverted pendulum environment, with state $s = [\theta, \omega]^\top$. The region past the horizontal $\mathcal{G}$ is to be avoided at all times.

b) Environment interaction: Episodes start from a state s_0 sampled from the initial distribution ρ. To collect $(s,a,s',i(s))$ transitions and store them in a memory buffer $\mathcal{M}$ we run episodes by following a policy that satisfies (7), for example the *uniform safe policy*, which takes actions uniformly over the presumed-safe ones:

$$\pi^\theta(a|s) = \begin{cases} 0 & \text{if } \hat{b}^\theta(s,a) = 1 \\ 1/\sum_{a' \in \mathcal{A}} \mathbb{1}\{\hat{b}^\theta(s,a') = 0\} & \text{if } \hat{b}^\theta(s,a) = 0 \end{cases} \tag{8}$$

c) Building the dataset: After collecting transitions, the binary value function is used to compute labels via the right-hand side of (5), that is, $y_k^\theta = i(s) + (1 - i(s)) \min_{a'} b^\theta(s'_k, a')$ for all $(s_k, a_k, i(s_k), s'_k) \in \mathcal{M}$. Note that these are "soft" labels $y_k^\theta \in [0,1]$. Those that satisfy $y_k^\theta \geq \frac{1}{2}$ are added to $\mathcal{D}_{\texttt{unsafe}}$. This procedure is dubbed build_unsafe_dataset($b, \mathcal{M}$) in Algorithm 1.

d) Training the network: The network is trained by running mini-batch gradient descent on the binary cross-entropy loss until it can correctly predict all the labels in $\mathcal{D} := \mathcal{D}_{\texttt{safe}} \cup \mathcal{D}_{\texttt{unsafe}}$. Once that is achieved, the labels in $\mathcal{D}_{\texttt{unsafe}}$ are re-computed (some might have changed since b^θ was updated during this process), and the program checks whether it can correctly predict the labels again. It repeats this process until all labels are predicted correctly, yielding a binary function that is self-consistent across the whole dataset.

V. Numerical Experiments

We present numerical validations of our algorithm in an inverted pendulum environment [32], contrasting our method against SBE [21], a well-known safety-critic. This environment allows easy visualization of the learned control invariant safe sets and can be compared against numerically obtained "grounds truth" references.

[4] e.g., (s,a) pairs close to the system's equilibrium or sampled trajectories from a known, safe policy.

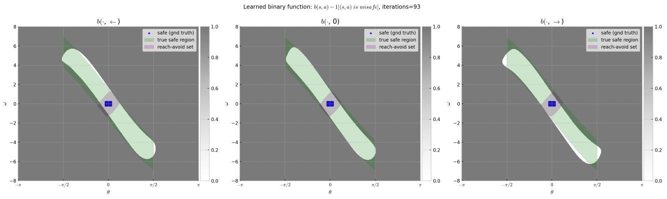

Fig. 4: Learned safe regions for the inverted pendulum problem during training. Each panel depicts the learned barrier for a fixed action (maximum clockwise torque, maximum counter-clockwise torque, no torque). The white area corresponds to the states classified as safe (for each of those actions). The solid maroon lines show the boundary of the unsafe region $\mathcal{G}$ (falling past the horizontal). The green region shows the set of states that can avoid $\mathcal{G}$ at all times, and the purple region shows the set of safe states reachable from $\mathcal{D}_{\texttt{safe}}$. These sets were computed using an optimal control toolbox [28].

a) Environment: The state of the system $s = [\theta,\ \omega]^\top$ is the angular position and angular velocity of the pendulum with respect to the vertical. The action $a \in [-a_{\texttt{max}}, a_{\texttt{max}}]$ is the torque applied on the axis, which we discretize in 5 equally spaced values. The *goal* in this task is to avoid falling past the horizontal, i.e., $\mathcal{G} = \{(\theta, \omega) : |\theta| \geq \frac{\pi}{2}\}$, as show in Fig. 3.

b) Training protocol: We take $\mathcal{D}_{\texttt{safe}}$ to be a small grid of (s, a)-pairs near the unstable equilibrium. Episodes are started from safe states (depicted in green in Fig. 4) and end whenever the pendulum reaches the unsafe region, or after 200 steps. The behavioral policy is the "uniform safe", as defined in (8). We alternate between collecting data for 10 episodes, building the dataset, and training the network as explained in Sec. IV. Details on network architecture and hyperparameters are relegated to the Appendix D.

c) Ground truth: We compare the safe region learned by our algorithm against ground truths computed numerically with optimal control tools [28]. Fig. 4 shows in green (resp. light gray) the maximum CIS set in the entire state (resp. the maximum CIS for trajectories that start inside the support of ρ). The learned safe region (in white) at different stages of training is also shown. At the beginning, the network is only fit to $\mathcal{D}_{\texttt{safe}}$. As episodes run and it collects unsafe transitions, it effectively learns a CIS set included in the true safe region for the problem.

d) Training performance: We benchmark our proposed methodology against the Safety Bellman Equation (SBE) of [21]. This algorithm learns a safety-critic $q(s, a)$ and considers actions to be "safe" if $q(s, a) \geq \eta$ for a prescribed threshold η. Hyperparameters for that algorithm are taken from [24] and can be seen in Appendix D. Fig. 5 (left) shows the cumulative failures during training; (a *failure* is an episode that touched the unsafe region $\mathcal{G}$). Our algorithm is significantly safer during training.

e) Post-training evaluation: We evaluate the performance of each model after training and show it in Fig. 5 (right). We test the uniform safe policy of our model against the safety critic for SBE. In the latter, we consider—for varying threshold η—the safe policy that maximizes exploration, i.e., the uniform policy taking actions a such that $q(s, a) \geq \eta$. We illustrate the *safety rate*, defined as the proportion of safe episodes, and the *average entropy* of each policy $\tilde{\mathcal{H}}_\pi \triangleq \mathbb{E}_{s \sim \mathcal{RA}} [\mathcal{H}(\pi(\cdot \mid s))]$, where $\mathcal{RA}$ is the set of safe states reachable from the origin (see 'reach-avoid' set in Fig. 4). Our algorithm obtains a perfect safety rate, while SBE only achieves it for safer policies (large enough η). These latter policies, though safe, are less exploratory—i.e., smaller entropy—than ours. In summary, our achievements are twofold: we learn a *persistently safe* family of policies that is *more exploratory* than the SBE counterpart. As argued in Section II-A, for traditional safety critics, there is no straightforward connection between the threshold η and discount factor $\gamma < 1$ needed to achieve safe policies, and safety comes at the expense of less exploration, which is undesired and difficult to balance. The solution found with our algorithm strikes a good balance between safety and the richness of the class of policies guaranteed to be safe.

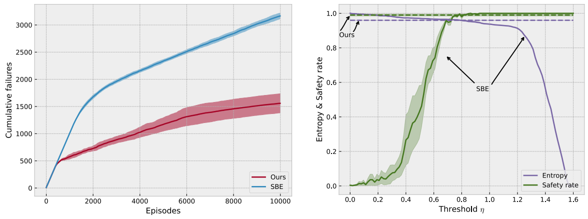

Fig. 5: Left: cumulative failures during training of our algorithm (red) and SBE (blue) for the inverted pendulum. Solid lines represent the means across 5 seeds; shaded areas are 95% confidence intervals. Our algorithm learns safe policies with less failures. Right: safety rate (fraction of safe episodes) and *entropy* of each learned model. Our algorithm (shaded lines) always uses the uniform safe policy. SBE is tested for different threshold values η. Our policy achieves almost perfect safety rate and is exploratory (high entropy). Only the most conservative SBE policies (large η) are 100% safe, but have low entropy (limited exploration).

VI. Conclusion

We proposed a framework for obtaining correct-by-design safety critics in RL, to persistently avoid a region of the state space. Our framework exploits the logical safe/unsafe nature of the problem and yields binary Bellman equations with multiple fixed points. We argue that all these fixed points are meaningful by characterizing their structure in terms of safety and maximality. Numerical experiments validate our theory and show that we can safely learn safer, more exploratory policies.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control

through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[3] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser, *et al.*, "Starcraft ii: A new challenge for reinforcement learning," *arXiv preprint arXiv:1708.04782*, 2017.

[4] J. A. Nichols, H. W. H. Chan, and M. A. Baker, "Machine learning: applications of artificial intelligence to imaging and diagnosis," *Biophysical reviews*, vol. 11, no. 1, pp. 111–118, 2019.

[5] C. Yu, J. Liu, S. Nemati, and G. Yin, "Reinforcement learning in healthcare: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–36, 2021.

[6] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[7] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, Y. Yang, and A. Knoll, "A review of safe reinforcement learning: Methods, theory and applications," *arXiv preprint arXiv:2205.10330*, 2022.

[8] S. Paternain, M. Calvo-Fullana, L. F. Chamon, and A. Ribeiro, "Safe policies for reinforcement learning via primal-dual methods," *IEEE Transactions on Automatic Control*, vol. 68, no. 3, pp. 1321–1336, 2022.

[9] A. Castellano, H. Min, J. A. Bazerque, and E. Mallada, "Learning to act safely with limited exposure and almost sure certainty," *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 2979–2994, 2023.

[10] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6070–6120, 2017.

[11] W. Chen, D. Subramanian, and S. Paternain, "Probabilistic constraint for safety-critical reinforcement learning," *arXiv preprint arXiv:2306.17279*, 2023.

[12] S. Li and O. Bastani, "Robust model predictive shielding for safe reinforcement learning with stochastic dynamics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7166–7172, IEEE, 2020.

[13] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Learning for Dynamics and Control*, pp. 708–717, PMLR, 2020.

[14] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3717–3724, IEEE, 2020.

[15] A. Castellano, H. Min, E. Mallada, and J. A. Bazerque, "Reinforcement learning with almost sure constraints," in *Learning for Dynamics and Control Conference*, pp. 559–570, PMLR, 2022.

[16] D. Bertsekas, "Infinite time reachability of state-space regions by using feedback control," *IEEE Transactions on Automatic Control*, vol. 17, no. 5, pp. 604–613, 1972.

[17] E. D. Sontag, *Mathematical control theory: deterministic finite dimensional systems*, vol. 6. Springer Science & Business Media, 2013.

[18] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2242–2253, IEEE, 2017.

[19] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 98–106, IEEE, 2018.

[20] I. M. Mitchell, "Comparing forward and backward reachability as tools for safety analysis," in *International Workshop on Hybrid Systems: Computation and Control*, pp. 428–443, Springer, 2007.

[21] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, S. Ghosh, and C. J. Tomlin, "Bridging hamilton-jacobi safety analysis and reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8550–8556, 2019.

[22] K. Srinivasan, B. Eysenbach, S. Ha, J. Tan, and C. Finn, "Learning to be safe: Deep rl with a safety critic," *arXiv preprint arXiv:2010.14603*, 2020.

[23] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg, "Recovery rl: Safe reinforcement learning with learned recovery zones," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4915–4922, 2021.

[24] K.-C. Hsu, V. Rubies-Royo, C. J. Tomlin, and J. F. Fisac, "Safety and liveness guarantees through reach-avoid reinforcement learning," *arXiv preprint arXiv:2112.12288*, 2021.

[25] D. P. Bertsekas, "Dynamic programming and optimal control 4th edition, volume ii," *Athena Scientific*, 2015.

[26] A. Schwartz, "A reinforcement learning method for maximizing undiscounted rewards," in *Proceedings of the tenth international conference on machine learning*, vol. 298, pp. 298–305, 1993.

[27] A. Girard, C. Le Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs," in *Hybrid Systems: Computation and Control: 9th International Workshop, HSCC 2006, Santa Barbara, CA, USA, March 29-31, 2006. Proceedings 9*, pp. 257–271, Springer, 2006.

[28] I. M. Mitchell and J. A. Templeton, "A toolbox of hamilton-jacobi solvers for analysis of nondeterminis-

tic continuous and hybrid systems," in *International workshop on hybrid systems: computation and control*, pp. 480–494, Springer, 2005.

[29] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[30] B. Chen, J. Francis, J. Oh, E. Nyberg, and S. L. Herbert, "Safe autonomous racing via approximate reachability on ego-vision," *arXiv preprint arXiv:2110.07699*, 2021.

[31] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[32] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis, "Gymnasium," Mar. 2023.

Appendix

A. Some identities

The t-step reachable set from s is the union of the $(t-1)$-step reachable set from all the successor states of s:

$$\mathcal{F}^\pi_t(s) = \bigcup_{s'\in\mathcal{F}^\pi(s)} \mathcal{F}^\pi_{t-1}(s') \tag{9}$$

Furthermore, the t-step reachable set from (s,a) is the $t-1$ set from the successor $s' = F(s,a)$:

$$\mathcal{F}^\pi_t(s,a) = \mathcal{F}^\pi_{t-1}(s') \qquad \forall t \geq 1 \tag{10}$$

B. Proof of Proposition 1

We will show:

$$b^\pi(s,a) = i(s) + \big(1 - i(s)\big) v^\pi(s') \qquad \text{where } s' = F(s,a).$$

The following identities hold, as explained below.

$$\begin{aligned}
b^\pi(s,a) &= \sup_{t\geq 0} \max_{s_t\in\mathcal{F}^\pi_t(s,a)} i(s_t) & (11)\\
&= \max\left\{ i(s),\ \sup_{t\geq 1} \max_{s_t\in\mathcal{F}^\pi_t(s,a)} i(s_t) \right\} & (12)\\
&= \max\left\{ i(s),\ \sup_{t\geq 1} \max_{s_t\in\mathcal{F}^\pi_{t-1}(s')} i(s_t) \right\} & (13)\\
&= \max\left\{ i(s),\ \sup_{t\geq 0} \max_{s_t\in\mathcal{F}^\pi_t(s')} i(s_t) \right\} & (14)\\
&= \max\left\{ i(s), v^\pi(s') \right\} & (15)
\end{aligned}$$

The first identity is the definition of b^π. In (12) unroll the first step in $\sup\{\cdot\}$. Next use the identity of (10). Finally introduce the change of variables $t \leftarrow t-1$ and recognize $v^\pi(s')$.

Recall $b^\pi(s,a), i(s)$ and $v^\pi(s')$ are binary. We consider two cases.

If $i(s) = 1$:

$$i(s) = 1 \geq b^\pi(s,a) \geq i(s) \implies b^\pi(s,a) = i(s) \tag{16}$$

If $i(s) = 0$:

$$b^\pi(s,a) = \max\{0, v^\pi(s')\} = 1 \cdot v^\pi(s') = (1 - i(s)) v^\pi(s') \tag{17}$$

Hence:

$$b^\pi(s,a) = i(s) + \big(1 - i(s)\big) v^\pi(s'),$$

which completes the first part of the proof.

Before proceeding to the last part of the proof, we note that a similar Binary Bellman equation holds for the value function (which we omitted in the manuscript for brevity):

Proposition 2 (Binary Bellman equation for v^π): For all π, for all $s \in \mathcal{S}, a \in \mathcal{A}$:

$$v^\pi(s) = i(s) + \big(1 - i(s)\big) \max_{s'\in\mathcal{F}^\pi(s)} v^\pi(s')\,; \quad s' = F(s,a)\,.$$

Proof: We omit the proof since it is virtually identical to equations 11–15. □

Now, going back to the proof of Proposition 1, remains to be shown:

$$b^\star(s,a) = i(s) + \big(1 - i(s)\big) \min_{a'\in\mathcal{A}} b^\star(s',a')\ .$$

In light of what we have just proved, it suffices to show the following Bellman optimality condition:

$$\min_{a\in\mathcal{A}} b^\star(s,a) = v^\star(s)\ . \tag{18}$$

We again consider two cases.

If $i(s) = 1$, then $\forall\pi, \forall a \in \mathcal{A}$:

$$1 = v^\pi(s) = b^\pi(s,a) \implies v^\star(s) = \min_a b^\star(s,a),$$

so the result holds trivially.

If $i(s) = 0$:

Let $\pi^\star$ be an optimal policy. Then by Proposition 2:

$$\begin{aligned}
v^\star(s) &= \overbrace{i(s)}^{=0} + \overbrace{\big(1 - i(s)\big)}^{=1} \max_{s'\in\mathcal{F}^{\pi^\star}(s)} v^\star(s') & (19)\\
&= \max_{s'\in\mathcal{F}^{\pi^\star}(s)} v^\star(s') & (20)\\
&= \max_{a\in\mathrm{Supp}[\pi(\cdot|s)]} v^\star(F(s,a)) & (21)\\
&\geq \min_{a\in\mathrm{Supp}[\pi(\cdot|s)]} v^\star(F(s,a)) & (22)\\
&\geq \min_{a\in\mathcal{A}} v^\star(F(s,a)) & (23)\\
&= \min_{a\in\mathcal{A}} \big[i(s) + \big(1 - i(s)\big) v^\star(F(s,a))\big] & (24)\\
&= \min_{a\in\mathcal{A}} b^\star(s,a) & (25)
\end{aligned}$$

where the first inequality follows from $\max \geq \min$, and the second one for optimizing over a larger set.

Thus:

$$v^\star(s) \geq \min_{a\in\mathcal{A}} b^\star(s,a) \quad \forall s \in \mathcal{S}, \tag{26}$$

and we want to show that the result holds with equality. By contradiction, assume the inequality is strict for some $s \in \mathcal{S}$, that is to say:

$$\exists a^\dagger \in \mathcal{A} : v^\star(s) > b^\star(s, a^\dagger). \tag{27}$$

Since the inequality is strict, it must be that $v^\star(s) = 1$ and $b^\star(s, a^\dagger) = 0$.

Now consider a policy $\pi^\dagger$ similar to $\pi^\star$, but that only takes action $a^\dagger$ at state s:

$$\pi^\dagger : \begin{cases} \pi^\dagger(a_1|s) = 1 \\ \pi^\dagger(\cdot|s') = \pi^\star(\cdot|s') \quad \forall s' \neq s \end{cases} \tag{28}$$

Let $v^\dagger \triangleq v^{\pi^\dagger}$. We then have:

$$v^\dagger(s) = \max_{s' \in \mathcal{F}^{\pi^\dagger}(s)} v^\dagger(s') = v^\dagger(F(s, a^\dagger)) = b^\pi(s, a^\dagger) < v^\star(s),$$

hence $v^{\pi^\dagger}(s) < v^\star(s)$ which means $\pi^\star$ is not optimal. This is a contradiction. It must then be that (26) holds with equality, as was claimed.

C. Proof of Thm. 1

Proof: Throughout this proof, we will make use of the following alternative representation of fixed points of the binary Bellman operator $\mathcal{T}$.

Lemma 1: $\tilde{b}$ is a fixed point of $\mathcal{T}$ if and only if it satisfies, for all $s \in \mathcal{S}$, for all $a \in \mathcal{A}$:

$$\tilde{b}(s, a) = \max\left\{ i(s), \min_{a' \in \mathcal{A}} \tilde{b}(s', a') \right\} \quad \text{where } s' = F(s, a)\ . \tag{29}$$

Proof: The proof follows from equations 11–15 in Proposition 1 applied to $\pi^\star$. □

Now, to the main proof.

a) Spourious fixed point: Firstly, note that $\tilde{b} \equiv 1$ is indeed one possible fixed point of (5): $\forall (s, a) \in \mathcal{S} \times \mathcal{A}$,

$$1 = \tilde{b}(s, a) = i(s) + \big(1 - i(s)\big) \min_{a'} \tilde{b}(s', a') \geq \min_{a'} \tilde{b}(s', a') = 1$$

b) $\mathcal{C}$ is CIS: Now suppose $\tilde{b}$ is non-trivial. We begin by showing *(i)*. By contradiction, assume $\mathcal{C}$ is not control invariant, i.e.:

$$\forall \pi, \exists s_0 \in \mathcal{C}, \exists t \geq 0 : \mathcal{F}_t^\pi(s_0) \not\subset \mathcal{C}.$$

We consider the "safest" policy that stems from $\tilde{b}$, only taking actions such that $\tilde{b}(s, a) = 0$. More generally, we consider any policy $\tilde{\pi}$ that satisfies:

$$\tilde{\pi}(s) : \forall s \in \mathcal{S}, \quad \operatorname{Supp}\left[\tilde{\pi}(\cdot|s)\right] \subseteq \arg\min_{a \in \mathcal{A}} \tilde{b}(s, a). \tag{30}$$

We have $\mathcal{F}_t^{\tilde{\pi}}(s_0) \not\subset \mathcal{C}$ for some $t \geq 0$. Hence there is a transition (s, a, s') such that $s \in \mathcal{C}, a \in \arg\min_{a' \in \mathcal{A}} \tilde{b}(s, a')$ and $s' = F(s, a) \notin \mathcal{C}$. Therefore:

$$0 \overset{(s \in \mathcal{C})}{=} \tilde{b}(s, a) = \max\left\{ i(s), \min_{a'} \tilde{b}(s', a') \right\} \geq \min_{a'} \tilde{b}(s', a') \overset{(s' \notin \mathcal{C})}{=} 1,$$

which is a contradiction. Hence $\mathcal{C}$ is control invariant—and moreover, the policy defined above renders it invariant (this shows *(iii)*).

Now to show that $\mathcal{C}$ is safe, again assume by contradiction $\mathcal{C}$ is not safe, i.e.:

$$\forall \pi, \exists s_0 \in \mathcal{C}, \exists t \geq 0 : \mathcal{F}_t^\pi(s_0) \cap \mathcal{G} \neq \emptyset.$$

Consider once again a "safest" policy as defined in (30) (that renders $\mathcal{C}$ invariant). This policy along with the non-empty intersection in the previous equation implies that:

$$\exists s \in \mathcal{C},\ a \in \operatorname{Supp}\left[\pi(\cdot|s)\right],\ t \geq 0 : s' = F(s, a) \in \mathcal{F}_t^\pi(s_0) \cap \mathcal{G} \implies$$

$$\begin{aligned} 0 \overset{(s \in \mathcal{C})}{=} & \tilde{b}(s, a) = \max\left\{ i(s), \min_{a' \in \mathcal{A}} \tilde{b}(s', a') \right\} \\ & \geq \min_{a' \in \mathcal{A}} \tilde{b}(s', a') = \min_{a' \in \mathcal{A}} \max\left\{ i(s'), \min_{a'' \in \mathcal{A}} \tilde{b}\left(F(s', a'), a''\right) \right\} \\ & = \max\left\{ i(s'), \min_{a', a'' \in \mathcal{A}} \tilde{b}\left(F(s', a'), a''\right) \right\} \geq i(s') \overset{(s' \in \mathcal{G})}{=} 1, \end{aligned}$$

which is a contradiction. Hence $\mathcal{C}$ is safe.

c) Maximality of the CIS: We finish by showing *(ii)*. Assume (by contradiction) $\mathcal{C}$ is unreachable from outside. Assume furthermore $i(s) = 0$ (if $i(s) = 1$, this would mean $s \in \mathcal{G}$).

$\mathcal{C}$ reachable from outside means:

$$\exists s \notin \mathcal{C}, \exists a \in \mathcal{A} : s' \triangleq F(s, a) \in \mathcal{C} \implies$$

$$\begin{aligned} 1 \overset{(s \notin \mathcal{C})}{=} & \min_{a} \tilde{b}(s, a) \leq \tilde{b}(s, a) = \max\left\{ i(s), \min_{a'} \tilde{b}(s', a') \right\} \\ \overset{i(s)=0}{=} & \min_{a'} \tilde{b}(s', a') \overset{(s' \in \mathcal{C})}{=} 0 \end{aligned}$$

□

D. Numerical experiments

TABLE I: Hyperparameters for inverted pendulum experiment

	B2E (Ours)	SBE
$\dim(\mathcal{S})$	3	
$\lvert\mathcal{A}\rvert$	5	
NN hidden layers	[256, 256]	[256,256]
NN activation	Tanh	Tanh
Learning rate[†]	$(1 - p) \times 10^{-4} + p \times 10^{-6}$	
Optimizer	Adam	
Discount γ	N/A	0.9999
Exploration factor	1	$\max\{0.95 \times 0.6^p, 0.05\}$
DDQN update	N/A	Hard every 10 episodes
Buffer size	50000	

[†] $p \triangleq$ progress, the fraction between the current episode and the total number of episodes.

a) SBE safety critic: For Safety Bellman equation (SBE) [21], the MDP at each step returns the signed distance to the unsafe set $h(s) = \frac{\pi}{2} - |\theta|$. The algorithm learns $q(s, a)$, and in principle any (s, a) such that $q(s, a) \geq 0$ is safe.

A more conservative safety-critic is one such that $q(s, a) \geq \eta$ for some $\eta > 0$. When evaluating the learned models (Fig. 5, right) we consider different policies π_η, defined as the uniform-safe over the presumed safe actions (similar to our case):

$$\pi_\eta(a|s) = \begin{cases} 0 & \text{if } q_\eta(s, a) < \eta \\ 1/\sum_{a' \in \mathcal{A}} \mathbb{1}\{q^\theta(s, a') \geq \eta\} & \text{if } q^\theta(s, a) \geq \eta \end{cases}$$