Safe Value Functions: Learned Critics as Hard Safety Constraints

Daniel Chee Hian Tan[1,2], Robert McCarthy[1], Fernando Acero[1], Andromachi Maria Delfaki[1], Zhibin Li[1], Dimitrios Kanoulas[1,3]

***Abstract*— In the domain of safety-critical applications, there is a pressing need for control methods that are not only scalable but also verifiable. Traditional control strategies, which rely on certification processes, often struggle to adapt to the complexity inherent in these systems. Conversely, while reinforcement learning (RL) techniques show promise in scaling effectively, their verifiability remains a significant challenge. Our research introduces a novel approach that bridges this gap by offering strong guarantees on constraint satisfaction for general dynamical systems, diverging from previous works that primarily focus on certification. Our study delves into the prerequisites for the verification of learned Value Functions (VFs) through the lens of Control Barrier Function (CBF) attributes. We leverage the foundational principles of safe VFs (SVFs) to design a reward mechanism that inherently guides the optimal VF to embody a CBF. Our approach allows the resulting VF to restrict subsequent policy actions to safe trajectories, in the context of complex control problems. Furthermore, we investigate the feasibility of conducting formal verification of VFs by exploiting CBF properties. This research marks a significant advancement towards achieving control methods that are both scalable to complex systems and amenable to rigorous verification processes. Through the integration of learning-based control with traditional safety guarantees, we pave the way for more reliable and efficient solutions in safety-critical applications. The code and supplementary video can be found under our webpage[1].**

I. Introduction

In the realm of robotics, applications deemed safety-critical, such as collaborative human-robot interactions, autonomous vehicular navigation, and versatile domestic robots, stand to gain significantly from control algorithms that not only scale with the complexity of robotic platforms but also come with verifiable safety assurances. The advent of deep reinforcement learning (DRL) has showcased its scalability and efficacy across a broad spectrum of complex tasks ranging from playing Atari games [1], to robotic manipulation [2], and even to protein structure prediction [3], as per the insights of Sutton and Barto [4]. Historically, the assurance of safe control in such systems has been the purview of safety certificates, with Control Barrier Functions (CBFs) being a prominent example. These functions assign scalar values to system states, delineating a safe operational zone. Traditional methodologies involve crafting these certificates for specific dynamical systems, thereby enabling the derivation of control laws that adhere to safety constraints [5], [6], [7]. Provided the existence of a safe action for every state, such strategies ensure the system's operations remain within the bounds of safety. However, the manual design of certificate functions for complex systems remains a challenge, limiting their practical deployment. This limitation has spurred interest in methodologies that learn these certificate functions [8], [9], [10], [11], [12], [13], though predominantly these methods have been applied to control-affine systems and often rely on known system dynamics for loss function formulation, presenting a significant barrier to their generalization.

On a parallel track, safe reinforcement learning (Safe RL) seeks to infuse RL policies with constraints to ensure safety, operating within the framework of constrained Markov Decision Processes (cMDPs) [14], and striving to optimize rewards while maintaining cost below a predetermined threshold [15]. Unlike certificate-based methods, Safe RL does not presuppose specific knowledge about the system, rendering it potentially more versatile. However, despite theoretical safety assurances, practical implementation often falls short, primarily due to the challenges in verifying policy convergence and the potential for policy and critics imperfections. This delineates a clear dichotomy within safe control research: the trade-off between robust safety guarantees and widespread applicability. Our work endeavors to bridge this divide by enhancing the safety guarantees of RL-based control strategies through a novel synthesis of VFs and CBFs.

Contributions

This work is an extension of our ICML/WFVML paper [16] and presents several key advancements:

- Leveraging the foundational principles of SVFs, we introduce a reward mechanism that naturally guides the optimal VF to function as a CBF with a precisely definable safety threshold.
- Through the aforementioned reward structure, we illustrate the efficacy of our developed VFs in serving as a safety layer for subsequent policy applications.

[1]All authors are with Department of Computer Science, University College London, Gower Street, WC1E 6BT, London, UK. daniel.tan.22@ucl.ac.uk

[2]Daniel C.H. Tan is also with Institute for Infocomm Research, A*STAR, Singapore.

[3]Dimitrios Kanoulas is also with Archimedes/Athena RC, Greece.

This work was supported by the UKRI Future Leaders Fellowship [MR/V025333/1] (RoboHike). Daniel C.H. Tan is supported by the Agency for Science, Technology, and Research, Singapore. Fernando Acero is supported by the UKRI CDT in Foundational Artificial Intelligence (EP/S021566/1).

[1]https://rl-cbf.github.io/

- We further investigate the feasibility of conducting formal verification of these VFs by utilizing the inherent properties of CBFs.

Collectively, our contributions mark significant progress towards the development of control methodologies that are both scalable and amenable to verification, addressing a critical need in safety-critical applications.

II. Related Work

Safe Model-Based Policy Optimization (Safe MBPO) detailed by Thomas et al. [17] introduced analytic safety penalties within a model-based framework, while Massiani et al. [18] established a theoretical foundation for safe value functions, offering a reward framework that ensures reward-optimized policies adhere to state constraints. Building on these foundational works, our study introduces a pragmatic approach to employ learned critics as a mechanism for enforcing safety constraints in downstream tasks, thereby broadening the applicability of learning-based safety analysis. As mentioned above, we extend the theoretical and practical insights presented in our earlier workshop-presented work on safe VFs and control methodologies [16]. It situates itself within the broader context of learning-based control for safety-critical systems, augmenting and refining existing approaches with novel contributions.

Earlier efforts in this domain have utilized nominal certificates for reward shaping and have explored modifying RL algorithms to incorporate barrier critics for learning safe policies [19], [20], [21], [22], [23], [24], [25], [26]. While these studies concentrated on the direct learning of safe policies, our work diverges by examining the conditions under which learned VFs can serve as CBFs, facilitating their application in a broader range of tasks. A comprehensive review of safe learning-based control methods is provided by Brunke et al. [27].

The literature on learning neural certificates is vast, with many studies focusing on self-supervised learning within known control-affine dynamics [28], [29], [30], [31], [8] or certificates for discrete-time systems [32], [33]. Our approach distinguishes itself by being applicable to a wider variety of systems, including those with black-box dynamics, thus addressing a gap in the current body of research on certificate learning [34].

Our work also engages with the constrained RL literature, which aims at embedding safety constraints directly into the learning process to ensure policy safety [35], [36], [37]. Unlike these existing methods, which predominantly focus on augmenting Markov Decision Processes (MDPs) with safety constraints, our findings demonstrate that learned Q-functions can act as direct constraints to ensure safety, offering a novel perspective on enforcing safety within the framework of constrained RL.

In summary, our development of a reward framework that guides VFs to act as CBFs, our application of these functions as safety filters, and our exploration of their formal verification, represent significant strides towards creating scalable and verifiable control methods for safety-critical applications. We aim to address the theory pressing need for reliable control strategies in complex systems.

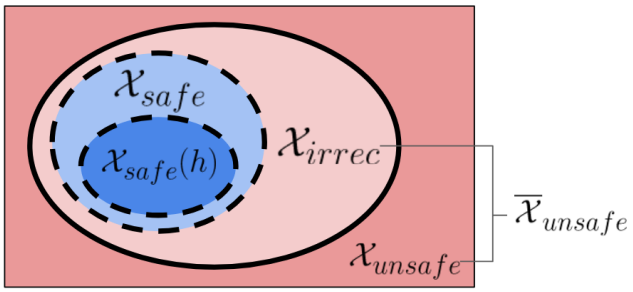

Fig. 1. The state space $\mathcal{X}$ is partitioned into unsafe states $\mathcal{X}_{\text{unsafe}}$, irrecoverable states $\mathcal{X}_{irrec}$, and safe states $\mathcal{X}_{\text{safe}}$. When synthesizing a control barrier function h, the certified safe set $\mathcal{X}_{\text{safe}}(h)$ should be distinct from $\mathcal{X}_{irrec}, \mathcal{X}_{\text{unsafe}}$.

III. Preliminaries

A. Markov Decision Processes

In this work, we consider deterministic Markov Decision Processes (MDPs) with states $x \in \mathcal{X}$ and actions $u \in \mathcal{U}$. Formally, an MDP can be written as a tuple $M = (\mathcal{X}, \mathcal{U}, f, r, \gamma)$, where $f(x,u)$ is the dynamics function, $r(x,u)$ is the reward function, and $\gamma \in [0,1)$ is the discount factor. The objective is to find the control that maximizes expected discounted reward $J = \sum_{t=0}^{\infty} \gamma^t r(x_t, u_t)$.

B. Reinforcement Learning

Reinforcement learning (RL) is a general framework for optimal control in MDPs. Given a policy π, we define its value as $V_\pi = \sum_{t=0}^{\infty} \gamma^t r(x_t, \pi(x_t))$. The optimal VF satisfies the Bellman condition: $V^*(x) = r(x,u^*) + \gamma V^*(x')$. In practice, it is common to define the Q-value function $Q(x,u) = r(x,u,x') + \gamma V(x')$, where $V(x) = \sup_u Q(x,u)$. RL consists of repeatedly looping two steps: (i) policy evaluation, (ii) policy iteration. In policy evaluation, given the current policy iterate $\pi^{(k)}$, we repeatedly apply the Bellman update $Q(x,u) \leftarrow (1-\alpha) * Q(x,u) + \alpha * (r(x,u) + \gamma V(x'))$ to determine $Q_{\pi^{(k)}}$. In policy iteration, we optimize the next policy iterate $\pi^{(k+1)}$ to maximize $V_{\pi^{(k)}}$. It is well-known that $\pi^{(k)}, Q^{(k)}$ converge to π^*, Q^* respectively [4].

C. Safe Control

We consider augmenting an MDP with a set of *safety violations* $\mathcal{X}_{\text{unsafe}}$, unsafe states specified by the practitioner. We may subdivide $\mathcal{X} - \mathcal{X}_{\text{unsafe}}$ into irrecoverable states $\mathcal{X}_{irrec}$ and safe states $\mathcal{X}_{\text{safe}}$ (see Fig. 1). Irrecoverable states arise from the system dynamics; for example, due to input limits, or due to under-actuation. By definition, states $x \in \mathcal{X}_{irrec}$ necessarily transition into $\mathcal{X}_{\text{unsafe}}$, regardless of the control applied, after some number of timesteps k_x. It is common to assume a uniform bound $k_x \leq H, \forall x \in \mathcal{X}_{irrec}$ [18], [17]. The value of H mainly depends on the actuation capacity of the system; all else being equal, systems with stricter actuation limits will have a longer horizon of irrecoverability.

We say that a trajectory $\tau = \{x_t : t \in \mathbb{N}\}$ is safe if $x_t \notin \mathcal{X}_{\text{unsafe}}$ for all $x_t \in \tau$. A control policy π is safe if, for

all $x \in \mathcal{X}_{\text{safe}}$, the trajectory defined by $x_{+1} = f(x_t, \pi(x_t))$ is safe.

D. Control Barrier Functions

A Control Barrier Function (CBF) $h : \mathcal{X} \to \mathbb{R}$ assigns a pseudo-energy to each state in the state space, such that the set $\{x : h(x) \geq 0\}$ defines a safe set $\mathcal{X}_{\text{safe}}(h) \subseteq \mathcal{X}_{\text{safe}}$. Formally, given $(M, \mathcal{X}_{\text{unsafe}})$, and $\alpha \in (0, 1]$, we say that $h : \mathcal{X} \to \mathbb{R}$ is a (discrete-time) CBF *against* $\mathcal{X}_{\text{unsafe}}$ if it satisfies:

$$\begin{aligned} &(i) \quad \forall x \in \mathcal{X}_{\text{unsafe}}, \quad h(x) < 0 \\ &(ii) \quad \forall x : h(x) \geq 0, \quad \sup_u \{h(f(x,u))\} \geq (1-\alpha)h(x) \end{aligned} \tag{1}$$

Here, $\alpha \in [0, 1]$ is some constant. We will derive suitable values of α in Section IV-A. We note the following properties:

Lemma III.1. *By condition (1)(i), $\mathcal{X}_{\text{safe}}(h) \cap \mathcal{X}_{\text{unsafe}} = \emptyset$. By condition (1)(ii), there exists a safe policy $\pi = \sup_u \{h(f(x,u)\}$. Proof: Note that if $x \in \mathcal{X}_{\text{safe}}(h)$, then $h(x') \geq (1-\alpha)h(x) \geq 0$. Hence, $x' \in \mathcal{X}_{\text{safe}}(h)$. Since $\mathcal{X}_{\text{safe}}(h) \subseteq \mathcal{X}_{\text{safe}}$, we have that π is safe.*

A CBF h eliminates the need to reason about safety over long horizons. Instead, we only need to check a one-step bound in condition (1)(ii) to guarantee safety. Hence, a CBF is capable of acting as a *safety filter* for downstream policies. One edge case occurs when $\mathcal{X}_{\text{safe}}(h) = \emptyset$; we call such CBFs *trivial*. Subsequently we assume that there exist nontrivial CBFs against $\mathcal{X}_{\text{unsafe}}$ (if not, this indicates that $\mathcal{X}_{\text{unsafe}}$ is 'too large' and we should reconsider the choice of $\mathcal{X}_{\text{unsafe}}$).

E. Safe Value Functions

We say that a value function $V : \mathcal{X} \to \mathbb{R}$ is a safe value function if the optimal policy for V is also guaranteed to remain indefinitely safe. Previous work [18] has established the sufficient condition (2) for V to be a SVF, and noted the connection to control barrier functions.

$$\sup_{x \in \mathcal{X}_{irrec} \cup \mathcal{X}_{\text{unsafe}}} V(x) < \inf_{x \in \mathcal{X}_{\text{safe}}} V(x) \tag{2}$$

IV. Reinforcement Learning of CBFs

A. SVFs from the Zero-One Safety Reward

In previous works [18], [17], the authors propose to apply a reward penalty on unsafe states, and show that a sufficiently large penalty results in a SVF. However, in practice, it is difficult to know what value the penalty should take. For one, the penalty depends on the magnitude of the task reward. Furthermore, in tasks with dense rewards, high safety penalties may be required [18]. For reward-penalty methods, it is also highly nontrivial to recover the resulting safety threshold. To avoid these problems, we propose to study the zero-one safety reward as a simple task structure that induces a safe value function V_{safe}.

Theorem IV.1. *Define $r_{safe}(x) = 0$ if $x \in \mathcal{X}_{\text{unsafe}}$ and $r_{safe}(x) = 1$ otherwise, and assume early termination. Then, V^*_{safe} is a safe value function.*

To see why, we consider the cases of $\mathcal{X}_{\text{safe}}, \mathcal{X}_{irrec}, \mathcal{X}_{\text{unsafe}}$ respectively. leftmargin=*

- $x \in \mathcal{X}_{\text{unsafe}}$. Since the episode terminates immediately, we trivially have $V^*_{safe}(x) = 0$.
- $x \in \mathcal{X}_{\text{safe}}$. In this case, we know there exists a policy which preserves safety indefinitely, hence we have $V^*_{safe}(x) = \sum_{j=0}^{\infty} \gamma^j(1) = \frac{1}{1-\gamma}$.
- $x \in \mathcal{X}_{\text{irrec}}$. Let x be k-irrecoverable. Then $V^*_{safe}(x) = \sum_{j=0}^{k-1} \gamma^j = \frac{1-\gamma^k}{1-\gamma}$.

Then $\sup_{x \in \mathcal{X}_{irrec} \cup \mathcal{X}_{\text{unsafe}}} V^*_{safe}(x) < \inf_{x \in \mathcal{X}_{\text{safe}}} V^*_{safe}(x) = \frac{1-\gamma^k}{1-\gamma} < \frac{1}{1-\gamma} = \inf_{x \in \mathcal{X}_{\text{safe}}} V^*_{safe}(x)$; hence $V^*_{safe}(x)$ is satisfies the condition 2.

Here, we have written V^*_{safe} with the subscript to emphasize that it is derived from the zero-one safety reward r_{safe}, as opposed to a task reward r_{task}. In subsequent sections, we omit this notation for brevity. Unless otherwise stated, the reward function considered is r_{safe}.

B. Analytic Safety Threshold for SVFs

In the previous section, we showed that V^*_{safe} is a safe value function. We now derive our main theoretical result: an analytic safety threshold $R \in [0, \frac{1}{1-\gamma}]$ such that $V^*_{safe} > R$ guarantees safety. We do this by relating $V^*_{safe} - R$ to a control barrier function h:

$$h^* = V^*_{safe} - R \tag{3}$$

We find that a more general result holds: For learned approximations $V_{safe} \approx V^*_{safe}$, we can find R, α such that $h = V_{safe} - R$ is a valid CBF, if we assume global error bounds $|V_{safe} - V^*_{safe}| < \epsilon$. We formalize this in the next theorem.

Theorem IV.2. *Assume that V satisfies $\sup_{x \in X} |V(x) - V^*(x)| < \epsilon$ for $\epsilon < \frac{\gamma^H}{2(1-\gamma)}$. Then for $\alpha \in [\frac{2\epsilon}{\frac{1}{1-\gamma} + \epsilon - R}, 1]$ and any $R \in (\frac{1-\gamma^H}{1-\gamma} + \epsilon, \frac{1}{1-\gamma} - \epsilon]$, we have that $h_{safe} = V_{safe} - R$ is a control barrier function against $\mathcal{X}_{\text{unsafe}}$.*

Proof. We prove that $h = V - R$ satisfies both conditions discussed in (1) to be a valid CBF. First, let $x \in \mathcal{X}_{\text{unsafe}}$; then $V(x) \leq V^*(x) + \epsilon = \frac{1-\gamma^H}{1-\gamma} + \epsilon < R$. Hence $h(x) < 0$ and condition (1)(i) is satisfied.

Now, let $h(x) \geq 0$. Then $x \in \mathcal{X}_{\text{safe}}$, thus $\sup_u h(f(x,u)) \geq V^*(x) - \epsilon - R = \frac{1}{1-\gamma} - \epsilon - R$. Similarly, we have $h(x) \leq V^*(x) + \epsilon - R = \frac{1}{1-\gamma} + \epsilon - R$. Then, to satisfy condition (1)(ii), it suffices that:

$$\begin{aligned} &\frac{1}{1-\gamma} - \epsilon - R \geq (1-\alpha)(\frac{1}{1-\gamma} + \epsilon - R) \\ \implies &\alpha \geq \frac{2\epsilon}{\frac{1}{1-\gamma} + \epsilon - R} \end{aligned}$$

Note that this can be satisfied because $R < \frac{1}{1-\gamma} - \epsilon$; hence the R.H.S is strictly smaller than 1. Hence condition (1)(ii) is satisfied, which completes the proof. □

Remark IV.3. The bounds on R are rather permissive. To illustrate, let $H = 10$ as in [17] and $\gamma = 0.99$. Then $\epsilon \leq$

$\frac{1-\gamma^H}{2(1-\gamma)} \approx 47$ suffices, inducing a corresponding $R = \frac{1}{1-\gamma} - \epsilon \approx 53$ and $\alpha = 0.96$. For smaller ϵ, a wider range of values of R will be valid.

C. Reinforcement Learning of CBFs.

Our proposed method is a simple modification of any baseline RL algorithm. To avoid committing safety violations during online exploration, we focus on the offline RL setting. Concretely, we assume access to an offline dataset of experience $\mathcal{D}$, and an RL algorithm $\mathcal{A}$. We obtain a modified dataset $\mathcal{D}'$ by relabelling all $(x, u, x') \in \mathcal{D}$ with r_{safe}. Lastly, we apply the RL algorithm $\mathcal{A}$ on $\mathcal{D}'$; the resulting value function is a CBF.

For our experiments, in line with standard actor-critic RL methods for high-dimensional continuous-control environments, we propose to parametrize $V = \sup_u Q(x, u)$. Based on the relation $h_{safe} = V_{safe} - R$, we can then interpret learned value functions as a CBF for any R satisfying the bounds derived in Theorem IV.2. In practice, we find that $R = \frac{1}{2(1-\gamma)}$ works well empirically.

V. Validation

As mentioned in the introduction, this paper provides a theoretical approach to verify safe policies using control theory, mainly by proving that the Value Functions (VFs) and Control Barrier Functions (CBFs). To validate this claim, in this section, we use a standard robotics control benchmark for Reinforcement Learning (RL) to apply the method on offline learning tasks (locomotion in Sec. V-A) and on online settings (CartPole in Sec. V-B)

A. Offline Learning: Locomotion

To validate the offline learning part, we use a standard control benchmark for RL is the locomotion environments from the Gymnasium [38] (Hopper and Walker2D) implemented in MuJoCo [39] and the D4RL datasets [40]. We visualize the environments in Fig. 2. We take the safety condition to be the same as the default termination condition (for Hopper: $0.8 \leq z \leq 2.0$ and for Walker2D: $0.7 \leq z$, with z being the height of the head of the Hopper/Walker2D from the ground), which checks whether the robot has fallen.

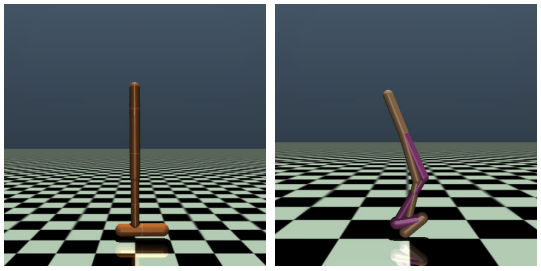

Fig. 2. Visualization of the MuJoCo locomotion environments considered (left: Hopper, right: Walker2d.

In these experiments, we learn the CBF using conservative Q-learning [41], which implements *conservative policy evaluation*. Compared to standard policy evaluation, CQL "pushes up" the Q-values of state-action pairs in the dataset distribution $(x, u) \sim \mathcal{D}$ and "pushes down" the Q-values of unseen actions from known states $x \sim \mathcal{D}, u \sim \pi(x)$. The effect is to reduce over-estimation of Q-values of unseen actions, which is ideal for safety.

Environment	Dataset	Safety-filtered episode length	Filter passthrough
Walker2d	random	89.3 ± 22.3	0.000 ± 0.000
Walker2d	medium	951.9 ± 46.8	$\mathbf{0.061 \pm 0.003}$
Walker2d	expert	$\mathbf{1000 \pm 0}$	0.030 ± 0.003
Walker2d	mixed	993.6 ± 11.0	0.029 ± 0.002
Hopper	random	451.5 ± 389.3	0.000 ± 0.000
Hopper	medium	340.5 ± 10.0	0.234 ± 0.037
Hopper	expert	$\mathbf{550.0 \pm 59.5}$	0.287 ± 0.013
Hopper	mixed	337.5 ± 25.4	$\mathbf{0.344 \pm 0.013}$

TABLE I

Safety-filtered episode length and exploration fraction for learned CBFs on the Walker2d and Hopper environments across 10 episodes and 3 seeded runs

We find that conservatism is an important inductive bias, which we discuss further in Sec. V-A.4. Although the reward contains no direct information about the locomotion task, we find that the resulting policies still learn effective locomotion due to the inductive bias provided by the demonstrations. Our implementation of CQL is adapted from Clean Offline RL [42].

1) Safety Filter with Learned CBF: We first demonstrate that the critics learned with our method are CBFs that can act as a safety filter for downstream policies. Concretely, we consider a nominal policy π_{nom}. At every timestep, if $Q(x, \pi_{nom}(x)) < R$, we deem the nominal action to be unsafe and instead take the safe action $\pi_{safe}(x)$. Otherwise, we do not modify π_{nom}. Here, we choose π_{nom} to be the policy that takes an action uniformly sampled from the action space, which consists of the hypercube $[-1, 1]^n$.

We evaluate the *safety-filtered episode length* of the filtered policy, defined as the number of timesteps the safety-filtered nominal policy remains safe. A valid CBF h will preserve safety for the full $T = 1000$ timesteps for each trajectory. To check for cases where the CBF is trivial ($\mathcal{X}_{\text{safe}}(h) = \emptyset$), we additionally evaluate *filter passthrough*, defined as the fraction of timesteps where an unfiltered action was taken. The filter ratio is a measure of the size of $\mathcal{X}_{\text{safe}}(h)$; a trivial CBF will have filter passthrough of 0.

We train CBFs on the Walker2d and Hopper environments, across 4 choices of dataset: random, medium, expert, and a *mixed* dataset consisting of a mixture of the former 3 datasets for $1M$ timesteps[2]. We present final results in Table I. Overall, we find that 'medium', 'expert', and 'mixed' datasets all result in broadly similar safety performance. Across the two environments, we find that the 'mixed' dataset has the best trade-off between a reasonably high safety-filtered episode length and filter passthrough. For this set of experiments, a safety threshold of $R = \frac{1}{2(1-\gamma)}$ was used. For the Walker environment, this results in rather low filter passthrough. We discuss the tradeoff induced by different choices of R in Sec. V-A.4.

2) Safe Control with Learned Policy: We now evaluate the policies learned with our approach on r_{task}. Our primary

[2]random, medium, and expert instances are provided by the D4RL [40]

aim is to demonstrate that that the CBF does not simply learn the trivial safety condition of standing still. As an auxiliary result, we also find that the learned policies are capable of safe control. Because r_{safe} contains no direct information about the task, we do not expect our approach to achieve expert-level task performance. Nonetheless, the learned policies will prefer safe trajectories over unsafe ones. Because performance on r_{safe} and r_{task} are correlated, the learned policies are also effective at r_{task}.

We compare our method to two standard RL baselines: PPO [43] and SAC [44]. These baselines simply optimize for J_{task} without considering safety. We list the final results reported by the implementation from CleanRL [45]. To formulate locomotion as a cMDP, we use the safe cost function $c(x, u) = 1$ if $x' \in \mathcal{X}_{\text{unsafe}}$ and 0 otherwise. The safe RL algorithm then optimizes π to maximize J_{task} subject to $J_c \leq 0$. Note that here, J_{task} is the expected infinite-horizon discounted sum of the locomotion task reward r_{task}. We consider two safe RL baselines: (i) PPO-Lagrangian [46], a Lagrangian-based method implemented on top of the base PPO algorithm, and (ii) constrained policy optimization (CPO) [15], a method for safe policy optimization based on trust regions. We use the implementations from OmniSafe [47].

We train all methods on $N = 10^6$ total environment transitions and measure final episode return (with respect to the original reward function). For our introduced method, we use the *mixed* dataset. Results are summarized in Table II.

Discussion. We find that, despite the caveats addressed above, policies learned with our method still learn effective task performance, achieving similar performance with the unconstrained RL baseline methods on Walker2d and outperforming the safe RL baselines on Hopper. We also note that our approach exhibits better sample efficiency compared to the safe RL baselines, which failed to learn effective policies within the training duration of 10^6 timesteps. This could be due to the fact that the used mixed dataset contains expert demonstrations, alleviating the need for exploration. With further training on $10\times$ more environment transitions, we note that both PPOLag and CPO achieved a similar final performance as the unconstrained baselines.

3) Formal Verification with CBF Metrics: Unlike other works, our theoretical framework allows us to formally verify learned value functions using control barrier function properties. If the learned CBF is valid, it can be used as a safety certificate; otherwise, it cannot provide a guarantee of the safety performance. Concretely, for each state x, we can define $\rho(h, x)$ to be a predicate that is true if the conditions in (1) hold and false otherwise. Given a state distribution $\mathcal{P}_{\mathcal{X}}$, we can then evaluate the *validity* metric $m_{valid}(h) = \mathbb{E}_{x\sim\mathcal{P}_{\mathcal{X}}}[\rho(h, x)]$. Here, we select $\mathcal{P}_{\mathcal{X}}$ to be the state distribution obtained by rolling out the safety-filtered policy.

For 10 checkpoints linearly spaced throughout the training duration, we plot safe episode length versus validity. Across the two environments, we find that validity is a useful predictor of determining safe episode length. In the case of Walker, final checkpoints achieved both high validity and high safety-constrained episode length. In the case of Hopper, the low validity throughout training indicates that the CBF has not learned an effective safety filter (See Fig. 3).

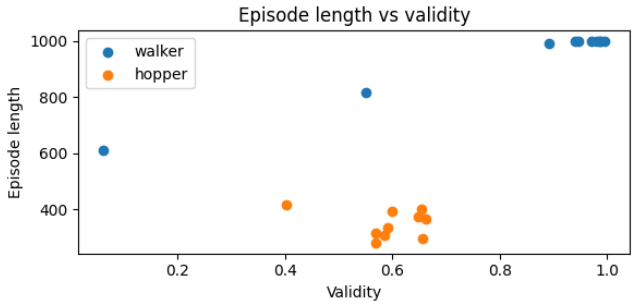

Fig. 3. A scatter plot of the validity vs safety-constrained episode length for many data points.

4) Analysis: In our main results, we used CQL to learn the CBFs. Instead of standard policy evaluation, CQL implements a conservative policy evaluation step to find a lower-bound estimate of V^*_{safe}. **Conservatism** is an ideal inductive bias as it is much more important to not overestimate the Q-value (and falsely predict safety in unsafe states) than it is to underestimate the Q-value. To study the importance of conservatism, we vary the base offline RL algorithm to TD3-BC [48]. We find that CQL performs much better on safe episode length at the cost of sacrificing some of the task performance.

In the above experiments, we use a fixed safety threshold of $\frac{1}{2(1-\gamma)}$. We now investigate the effect of varying the **safety threshold**. We plot safety-constrained episode length and filter passthrough in Fig. 4. We find that varying the safety threshold results in a smooth trade-off between safety-constrained episode length and exploration fraction.

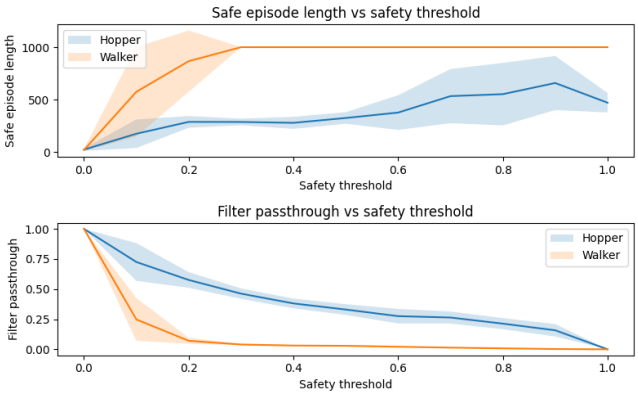

Fig. 4. Trade-off between filter passthrough and safe episode length as safety threshold is varied. Mean and standard deviation recorded over $n = 10$ episodes.

B. Online Setting: CartPole

We also explore how our method can be applied in the online setting, where the conservative regularization

Environment	Ours	PPOLag	CPO	PPO	SAC
Walker	$\mathbf{5000 \pm 8.3}$	752 ± 95	826 ± 209	3223 ± 885	4318 ± 404
Hopper	1480 ± 127	668 ± 27	994 ± 10	2264 ± 574	$\mathbf{2651 \pm 689}$

TABLE II

Episode return of learned policies of our approach versus baselines.

Environment	Dataset	CQL	TD3-BC
Walker2d	random	89.3 ± 22.3	87.9 ± 30.7
Walker2d	medium	951.9 ± 46.8	31.0 ± 13.1
Walker2d	expert	1000 ± 0	128.1 ± 16.4
Walker2d	mixed	993.6 ± 11.0	144.3 ± 71.7
Hopper	random	451.5 ± 389.3	333.5 ± 119.2
Hopper	medium	340.5 ± 10.0	204.2 ± 43.8
Hopper	expert	550.0 ± 59.5	56.0 ± 36.3
Hopper	mixed	337.5 ± 25.4	210.4 ± 32.7

TABLE III

Safety-filtered episode length of CQL vs TD3-BC. In all cases, CQL outperforms TD3-BC

penalty cannot be applied directly (since there is no dataset). To replace this, we instead exploit properties of V_{safe}. Recall from the analysis of Theorem IV.1 that $V^*_{safe} = 0$ on $\mathcal{X}_{\text{unsafe}}$. Therefore, we implement the *supervised* loss $\mathcal{L}_{\text{unsafe}} = \mathbb{E}_{x \sim \mathcal{X}_{\text{unsafe}}} \|V(x)\|$, which similarly "pushes down" the Q-values of state-action pairs leading to unsafe states. Since we can specify $\mathcal{X}_{\text{unsafe}}$, this loss can be approximated by sampling $\mathcal{X}_{\text{unsafe}}$ (e.g. by rejection sampling). Another benefit of conservatism is to prevent Q-value overestimation; we recreate this effect this using *bounding*. Recall that $V^*_{safe} \in [0, \frac{1}{1-\gamma}]$; therefore, we can bound the Q-values of the deep Q-network to $[0, \frac{1}{1-\gamma}]$ using a sigmoid activation.

In these experiments, we learn the CBF using Deep Q-Networks, modified with the two implementation details of *bounding* and *supervision*. We find that these implementation details are critical to achieving good CBF performance. We provide quantitative evaluations and qualitative analysis, showing that our method effectively learns CBFs that act as an effective safety filter. While rolling out the optimal policy results in the agent remaining within a relatively small portion of the state space, the safety-filtered policy traverses a large fraction of the state space while remaining safe, as depicted in Fig. 5.

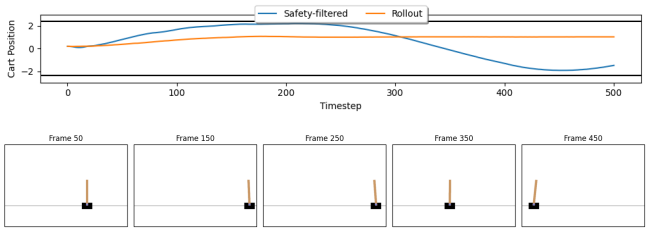

Fig. 5. While both policies remain safe (solid lines), safety-filtered policy (blue) traverses a large portion of the state space compared to the optimal policy (orange).

1) Analysis: In this section, we ablate the implementation details of *bounding* and *supervision*. Quantitative results are plotted in Fig. 6 We find that both bounding and supervision are essential to achieve good safety-filtered performance. As a sanity check, we plot the values of the learned CBFs across the 4-dimensional state space in Fig. 7. Consistent with the quantitative results, we find that bounding and supervision result in a large and correct $\mathcal{X}_{\text{safe}}(h)$.

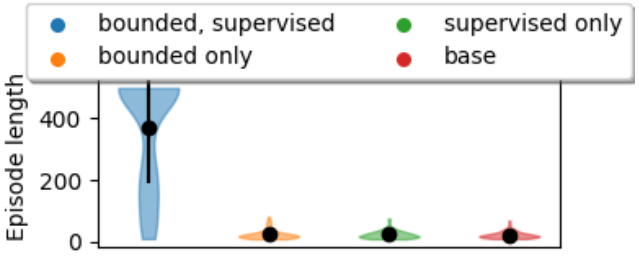

Fig. 6. Safety-filtered episode length across four experimental settings ablating bounding and supervision. The setting with both bounding and supervision, greatly outperforms other settings.

VI. Limitations and Future Work

Imperfect CBFs. Despite high empirical validity metrics measured by sampling, the fact that the validity is not at 100%, indicates that the learned CBF is imperfect. In general, formal verification of learned VFs is fundamentally difficult, so imperfections must be expected. Nonetheless, it is possible to deploy imperfect CBFs with online safety monitors [8] that detect when the CBF has become invalid, allowing fallback to an auxiliary safe policy. Alternatively, the theory of almost-Lyapunov functions [49] has been used effectively to mitigate similar issues in learning control Lyapunov functions [50].

Safety Filter for Downstream Tasks. In this work, we have shown that it is possible to learn CBFs which encode a safety criterion for general systems. In future work, pre-trained CBFs could be used for safe exploration while learning downstream tasks. Concretely, the CBF could be used to filter nominal actions to avoid exploring into unsafe states. Because the safety-filtered trajectories are off-policy for the unfiltered nominal policy, an off-policy method (such as SAC) would need to be used.

VII. Conclusion

In this work, we have primarily considered our approach within the safe reinforcement learning context. However, it is important to note that the same method could be used for certificate learning in known systems. Building on the theoretical connection between barrier functions and value functions, this paper demonstrates the feasibility of learning barrier functions through RL. We explore and ablate

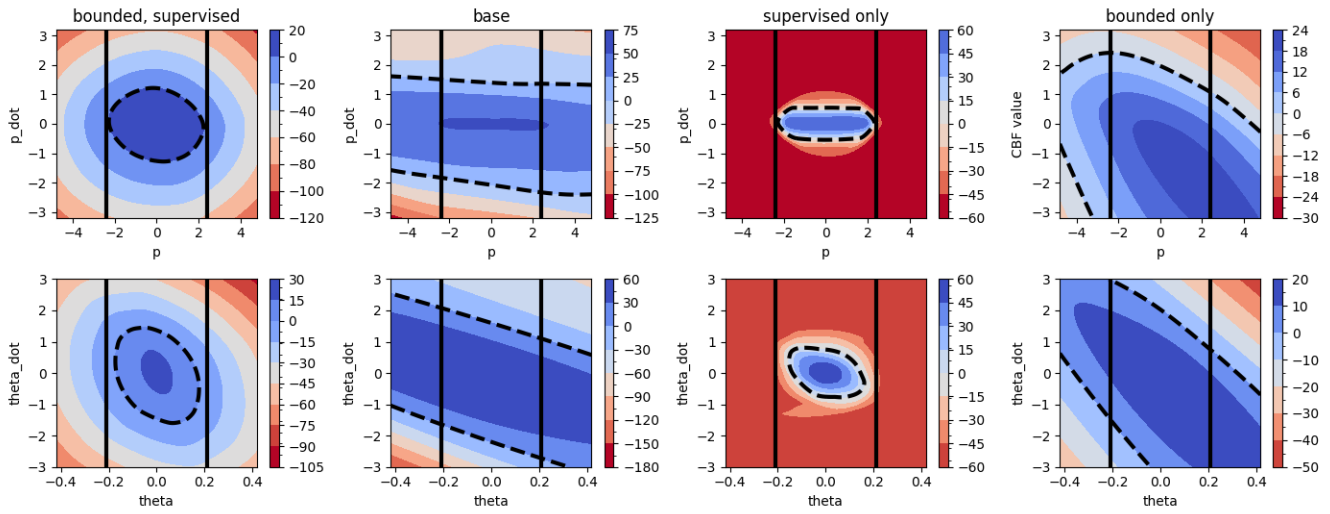

Fig. 7. Visualization of learned CBFs $h(x)$ over the state space $\mathcal{X}$ of CartPole. Top: Cart position, velocity $p, \dot{p}$. Bottom: Pole angle, angular velocity $\theta, \dot{\theta}$. $\mathcal{X}_{\text{safe}}(h)$ is marked in dotted lines and $\mathcal{X}_{\text{unsafe}}$ is marked in solid lines. Bounding and supervision results in a valid and large $\mathcal{X}_{\text{safe}}(h)$.

critical implementation details for learning high-quality barrier functions using our method. We also highlight a path towards formal verification based on CBF properties.

The proposed approach is especially suitable for learning *perceptual CBFs*, where safety can be defined as a direct function of sensor inputs. In one case study, perceptual CBFs on LiDAR scans enabled safe obstacle avoidance in cluttered environments [51]. In contrast to self-supervised learning, which requires careful handling of sensor dynamics, reinforcement learning naturally scales to end-to-end robot control [52], making it a promising alternative.

The method used in this work has broad applicability and can extend to any MDP M. This suggests that our method can be employed to learn barrier functions for safe control in diverse tasks. Overall, our work marks a step towards scalable and verifiable control methods.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013.

[2] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 9 2015. [Online]. Available: https://arxiv.org/abs/1509.02971v6

[3] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with alphafold," *Nature 2021 596:7873*, vol. 596, pp. 583–589, 7 2021. [Online]. Available: https://www.nature.com/articles/s41586-021-03819-2

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. A Bradford Book, 2018.

[5] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*, Dec. 2014, pp. 6271–6278, iSSN: 0191-2216.

[6] P. Wieland and F. Allgöwer, "CONSTRUCTIVE SAFETY USING CONTROL BARRIER FUNCTIONS," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, Jan. 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016355690

[7] C. Liu and M. Tomizuka, "Control in a Safe Set: Addressing Safety in Human-Robot Interactions," in *ASME 2014 Dynamic Systems and Control Conference*, vol. 3, Nov 2014.

[8] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods," 2022.

[9] Y.-C. Chang, N. Roohi, and S. Gao, "Neural Lyapunov Control," Sep. 2022. [Online]. Available: http://arxiv.org/abs/2005.00611

[10] M. Saveriano and D. Lee, "Learning Barrier Functions for Constrained Motion Planning with Dynamical Systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 112–119, arXiv:2003.11500 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2003.11500

[11] M. Srinivasan, A. Dabholkar, S. Coogan, and P. Vela, "Synthesis of Control Barrier Functions Using a Supervised Machine Learning Approach," Mar. 2020, arXiv:2003.04950 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2003.04950

[12] H. Ma, J. Chen, S. E. Li, Z. Lin, Y. Guan, Y. Ren, and S. Zheng, "Model-based Constrained Reinforcement Learning using Generalized Control Barrier Function," Mar. 2021, arXiv:2103.01556 [cs]. [Online]. Available: http://arxiv.org/abs/2103.01556

[13] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning Safe Multi-Agent Control with Decentralized Neural Barrier Certificates," Apr. 2021, arXiv:2101.05436 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2101.05436

[14] E. Altman, *Constrained Markov Decision Processes*. Routledge, 1999.

[15] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," 2017.

[16] D. C. Tan, F. Acero, R. McCarthy, D. Kanoulas, and Z. Li, "Your Value Function is a Control Barrier Function (Outstanding Paper Award)," in *International Conference on Machine Learning, Workshop on Formal Verification and Machine Learning*, 2023. [Online]. Available: https://www.ml-verification.com/2023/accepted-papers

[17] G. Thomas, Y. Luo, and T. Ma, "Safe reinforcement learning by imagining the near future," 2022.

[18] P.-F. Massiani, S. Heim, F. Solowjow, and S. Trimpe, "Safe value functions," *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 2743–2757, may 2023. [Online]. Available: https://doi.org/10.1109%2Ftac.2022.3200948

[19] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence,*

EAAI 2019, pp. 3387–3395, 2019. [Online]. Available: https://dl.acm.org/doi/10.1609/aaai.v33i01.33013387

[20] T. Westenbroek, A. Agrawal, F. Castañeda, S. S. Sastry, and K. Sreenath, "Combining model-based design and model-free policy optimization to learn safe, stabilizing controllers," *IFAC-PapersOnLine*, vol. 54, pp. 19–24, 2021, 7th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240589632101243X

[21] T. Westenbroek, F. Castaneda, A. Agrawal, S. Sastry, and K. Sreenath, "Lyapunov design for robust and efficient robotic reinforcement learning," 2022.

[22] L. Zhao, K. Gatsis, and A. Papachristodoulou, "A barrier-lyapunov actor-critic reinforcement learning approach for safe and stable control," 2023.

[23] Y. Yang, Z. Zheng, and S. E. Li, "Feasible policy iteration," 2023.

[24] P. Liu, D. Tateo, H. B. Ammar, and J. Peters, "Robot reinforcement learning on the constraint manifold," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 1357–1366. [Online]. Available: https://proceedings.mlr.press/v164/liu22c.html

[25] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe exploration in continuous action spaces," 2018.

[26] S. Liu, C. Liu, and J. Dolan, "Safe control under input limits with neural control barrier functions," 2022.

[27] 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, no. 1, pp. 411–444, 2022. [Online]. Available: https://doi.org/10.1146/annurev-control-042920-020211

[28] A. Abate, D. Ahmed, M. Giacobbe, and A. Peruffo, "Formal synthesis of lyapunov neural networks," *IEEE Control Systems Letters*, vol. 5, pp. 773–778, 7 2021. [Online]. Available: https://doi.org/10.1109%2Flcsys.2020.3005328

[29] S. M. Richards, F. Berkenkamp, and A. Krause, "The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems," 2018.

[30] G. Manek and J. Z. Kolter, "Learning stable deep dynamics models," *Advances in Neural Information Processing Systems*, vol. 32, 1 2020. [Online]. Available: https://arxiv.org/abs/2001.06116v1

[31] N. Gaby, F. Zhang, and X. Ye, "Lyapunov-net: A deep neural network architecture for lyapunov function approximation," *Proceedings of the IEEE Conference on Decision and Control*, vol. 2022-December, pp. 2091–2096, 9 2021. [Online]. Available: https://arxiv.org/abs/2109.13359v2

[32] J. W. Grizzle and J. M. Kang, "Discrete-time control design with positive semi-definite lyapunov functions," *Systems & Control Letters*, vol. 43, pp. 287–292, 7 2001.

[33] H. Dai, B. Landry, M. Pavone, and R. Tedrake, "Counter-example guided synthesis of neural network lyapunov functions for piecewise linear systems," *Proceedings of the IEEE Conference on Decision and Control*, vol. 2020-December, pp. 1274–1281, 12 2020.

[34] Z. Qin, D. Sun, and C. Fan, "Sablas: Learning safe control for black-box dynamical systems," *IEEE Robotics and Automation Letters*, vol. 7, pp. 1928–1935, 1 2022. [Online]. Available: https://arxiv.org/abs/2201.01918v2

[35] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward constrained policy optimization," 2018.

[36] A. Stooke, J. Achiam, and P. Abbeel, "Responsive safety in reinforcement learning by pid lagrangian methods," *37th International Conference on Machine Learning*, 2020.

[37] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A lyapunov-based approach to safe reinforcement learning," 2018.

[38] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[39] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[40] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4RL: Datasets for Deep Data-Driven Reinforcement Learning," Feb. 2021, arXiv:2004.07219 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2004.07219

[41] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-Learning for Offline Reinforcement Learning," Aug. 2020, arXiv:2006.04779 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2006.04779

[42] D. Tarasov, A. Nikulin, D. Akimov, V. Kurenkov, and S. Kolesnikov, "CORL: Research-oriented Deep Offline Reinforcement Learning Library," Jun. 2023, arXiv:2210.07105 [cs]. [Online]. Available: http://arxiv.org/abs/2210.07105

[43] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Aug. 2017, arXiv:1707.06347 [cs]. [Online]. Available: http://arxiv.org/abs/1707.06347

[44] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," Aug. 2018, arXiv:1801.01290 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1801.01290

[45] S. Huang, R. F. J. Dossa, C. Ye, J. Braga, D. Chakraborty, K. Mehta, and J. G. M. Araújo, "Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms," *Journal of Machine Learning Research*, vol. 23, pp. 1–18, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1342.html

[46] J. Achiam and D. Amodei, "Benchmarking Safe Exploration in Deep Reinforcement Learning," 2019.

[47] J. Ji *et al.*, "Omnisafe: An infrastructure for accelerating safe reinforcement learning research," *arXiv preprint arXiv:2305.09304*, 2023.

[48] S. Fujimoto and S. S. Gu, "A Minimalist Approach to Offline Reinforcement Learning," Dec. 2021, arXiv:2106.06860 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2106.06860

[49] S. Liu, D. Liberzon, and V. Zharnitsky, "Almost Lyapunov Functions for Nonlinear Systems," Dec. 2018, arXiv:1812.04474 [math]. [Online]. Available: http://arxiv.org/abs/1812.04474

[50] Y.-C. Chang and S. Gao, "Stabilizing neural control using self-learned almost lyapunov critics," 2021.

[51] C. Dawson, B. Lowenkamp, D. Goff, and C. Fan, "Learning safe, generalizable perception-based hybrid control with certificates," *IEEE Robotics and Automation Letters*, vol. 7, pp. 1904–1911, 4 2022. [Online]. Available: https://arxiv.org/abs/2201.00932v1

[52] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, pp. 1–40, 2016.