Relaxed Exploration Constrained Reinforcement Learning

Shahaf S. Shperberg
Ben-Gurion University
Beer Sheva, Israel
shperbsh@bgu.ac.il

Bo Liu
The University of Texas at Austin
Austin, Texas, United States
bliu@cs.utexas.edu

Peter Stone
The University of Texas at Austin and Sony AI
Austin, Texas, United States
pstone@cs.utexas.edu

ABSTRACT

This research introduces a novel setting for reinforcement learning with constraints, termed Relaxed Exploration Constrained Reinforcement Learning (RECRL). Similar to standard constrained reinforcement learning (CRL), the objective in RECRL is to discover a policy that maximizes the environmental return while adhering to a predefined set of constraints. However, in some real-world settings, it is possible to train the agent in a setting that does not require strict adherence to the constraints, as long as the agent adheres to them once deployed. To model such settings, we introduce RECRL, which explicitly incorporates an initial training phase where the constraints are relaxed, enabling the agent to explore the environment more freely. Subsequently, during deployment, the agent is obligated to fully satisfy all constraints. To address RECRL problems, we introduce a curriculum-based approach called CLiC, designed to enhance the exploration of existing CRL algorithms during the training phase and facilitate convergence towards a policy that satisfies the full set of constraints by the end of training. Empirical evaluations demonstrate that CLiC yields policies with significantly higher returns during deployment compared to training solely under the strict set of constraints. The code is available at https://github.com/Shperb/RECRL.

KEYWORDS

Constrained Reinforcement Learning, Curriculum Learning

ACM Reference Format:
Shahaf S. Shperberg, Bo Liu, and Peter Stone. 2024. Relaxed Exploration Constrained Reinforcement Learning. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 INTRODUCTION

Reinforcement learning algorithms aims to maximize (discounted) returns received from the environment. However, many real-world scenarios also require adhering to constraints not naturally characterized within the reward function. Examples include maximizing throughput without exceeding a specified average end-to-end delay in message routing [2], optimizing energy consumption for cooling data centers while staying below a certain temperature threshold [20], and maximizing distance driven under behavioral constraints (e.g., smoothness and lane centering) in autonomous driving problems. Other notable examples of constraints are fairness (e.g., balancing resources for workers in a factory [11]), safety [17], and energy consumption (fuel and/or power [39]). While it is possible to shape the reward function to introduce a high penalty upon constraint violation, this approach often hinders the learning process and leads to suboptimal policies (which often violate the constraints) [1]. To tackle this issue, constrained reinforcement learning (CRL) was introduced. In CRL, the objective is to optimize the return while simultaneously adhering to a predefined set of constraints. CRL is often modeled as a constrained Markov Decision Process (CMDP, [2]), where constraints are represented as cost functions with specified limits that cumulative costs must not exceed.

Many CRL approaches focus on quickly identifying and exploring policies that satisfy the constraints while improving return. Some even go further and attempt "safe exploration" [16] to avoid constraint violations during training. However, strict adherence to constraints can limit exploration and result in suboptimal policies. This strict approach is necessary for certain scenarios, especially those with critical safety constraints. However, in domains that admit for a safe training period, either in the real world or due to the availability of an accurate simulator, it becomes feasible to allow more relaxed exploration during training and enforce constraints only during deployment. For instance, consider the training of rescue workers tasked with saving individuals in cold water environments. In real operational scenarios, these workers might need to operate in frigid waters, enduring no more than three minutes before succumbing to hypothermia. However, during training, they could practice in relatively warmer waters with a 15-minute tolerance. To model such scenarios, we introduce the problem of relaxed exploration constrained reinforcement learning (RECRL). This framework, built upon the CMDP formulation, facilitates an initial relaxation of constraints throughout the training process, with the ultimate objective of developing policies that adhere to these constraints while maximizing environmental returns during actual deployment. Notably, while some existing CRL methods may experience constraint violations during training, their main objective is the swift convergence to a policy that satisfies constraints. In contrast, RECRL is purposefully crafted for domains where strict adherence to constraints during training is unnecessary. The clear differentiation between RECRL's training and deployment phases provides strategic flexibility regarding constraints, thereby enabling the attainment of superior policies.

In addition, we introduce the Cost-Limit Curriculum (CLiC) approach, which adapts existing CRL algorithms to RECRL. While CRL algorithms typically use cost limits that are constant throughout training, CLiC varies the cost-limit thresholds. Doing so enables the given CRL algorithm to learn a policy that adheres to deployment

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

constraints while achieving better returns compared to training with constant cost limits. In the example above, the rescue workers are trained with a gradually decreasing time limit in the water until they learn to accomplish the task within three minutes.

The main contributions of this work are as follows: i) Introduction and formulation of the problem of Relaxed Exploration CRL (RECRL). ii) Development of the CLiC approach with two types of curricula, predefined (static) curricula that decay the cost limits at different rates (linear, exponential, or cosine), and dynamic curricula that adjust cost limits based on agent-environment interactions. iii) Review of several existing CRL algorithms as inputs for CLiC, and an analysis of their safety guarantees with cost-limit curricula. iv) Demonstration of the approach on a toy gridworld domain. v) An empirical evaluation conducted on the Safety Gym benchmark [32], showcasing the performance improvements of multiple CRL algorithms achieved by CLiC.

2 RELATED WORK

This section briefly surveys the recent advances in constrained RL (CRL) and curriculum learning for RL.

Constrained RL. CRL is typically modeled as a constrained Markov decision process (CMDP, [2]), where the agent's goal is to optimize the environmental return under the condition that the expected discounted cost is below a pre-specified threshold [18, 23]. Various optimization methods, including augmented Lagrangian-based methods [21], trust-region methods [1], and Lyapunov-based methods [10], have been employed to address CMDP problems. Bayesian optimization has also been explored to address exploration in safe RL, where a backup safety policy corrects the agent's behavior in unsafe states, allowing exploration in the face of uncertainty [3, 7, 19, 38]. Learning from hallucination (LfH) is another method that balances exploration and learning under constraints by generating trajectories offline without constraints, then inferring restrictive constraints from this data [42]. In contrast to all this work, we do not assume that the constraints are constant throughout the training, thus granting the agent more freedom to explore and discover a safe policy that achieves a high return at convergence.

Throughout this paper, we will be focusing on two CRL algorithms, Constrained policy optimization (CPO, [1]), which enforces constraints throughout training by solving trust region optimization problems at each policy update, and PPO-Lagrangian (denoted by PPO-L, [32]), a variant of the well-known Proximal policy optimization (PPO, [36]) that enforces constraints by using adaptive penalty coefficients. While some of the CRL algorithms mentioned above are more sophisticated than CPO and PPO-L, these two algorithms are representative baseline algorithms that have been commonly used in CRL empirical evaluations (e.g., [4]).

Curriculum Learning in RL. Curriculum learning (CL, [8, 37]) involves designing curricula for specific tasks or task distributions to improve the agent's eventual performance compared to direct learning on the target task. CL has been applied in RL by modifying learning tasks' order and complexity [25]. Examples include decomposing hard tasks into easier missions [5], prioritizing transitions for policy updates [35], controlling initial or goal states [13, 31], and modifying reward functions or transition dynamics [41]. In related work, Turchetta et al. [40] considered a teacher-agent framework where the teacher generates a curriculum of subtasks to teach the agent safe behavior. The teacher intervenes by temporarily modifying transition dynamics to steer the agent back to safe states when it is in danger. Our approach, to the best of our knowledge, is the first to propose a curriculum based on cost limits for CRL problems. By relaxing constraints during training, we enable better exploration and a higher likelihood of finding improved solutions for CMDPs.

3 BACKGROUND ON CONSTRAINED MDPS

A constrained Markov decision process (CMDP) is an extension of the standard Markov decision process (MDP) that allows for constraints on the set of valid policies. Formally, a CMDP is a tuple $M = (\mathcal{S}, \mathcal{A}, T, \gamma, R, C, d)$. As in ordinary MDPs, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, and T is the transition function (where $T(s' \mid s, a)$ is the probability of reaching state s' as a result of taking action a from state s). $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function and γ is the discount factor, which determines the planning horizon. The expected discounted return of a policy π is defined as:

$$J(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_{j=0}^{\infty} \gamma^t \cdot R(s_t, a_t, s_{t+1})]$$

The last two elements in a CMDP, C and d, are used for restricting the set of feasible policies. $C = \{C_1, \ldots, C_k\}$ is a set of k cost functions, $C_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ for all $1 \le i \le k$. $d = \{d_1, \ldots, d_k\}$ is a set of cost limits that correspond to the cost functions in C. The set of feasible policies, which satisfy the constraints, is defined with respect to C and d:

$$\Pi(C, d) = \{\pi | J_{C_i}(\pi) \le d_i, \forall 1 \le i \le k\}$$

where $J_{C_i}(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{\infty} \gamma^t \cdot C_i(s_t, a_t, s_{t+1})]$

The CRL problem, defined for CMDPs, is to find a feasible policy that maximizes the expected discounted return,

$$\pi^* = \underset{\pi\in\Pi(C,d)}{\operatorname{argmax}} J(\pi)$$

While this work primarily focuses on discounted costs as constraints, the introduced methods can be easily extended to handle other constraint formulations, such as bounding the worst-case violation, value-at-risk (VaR), or conditional value-at-risk (CVaR) [33].

4 RELAXED EXPLORATION CRL

In CRL problems, the aim is to find a policy that maximizes environmental return while adhering to given constraints. To address scenarios where constraints can be relaxed during policy training, we introduce the *Relaxed Exploration Constrained Reinforcement Learning* (RECRL) problem.

In RECRL, agents undergo two distinct phases: a *training phase* and a *deployment phase*. During deployment, agents are restricted to policies that do not violate constraints, similar to standard CRL. However, the training phase allows for relaxed constraints (cost limits). Formally, a RECRL problem consists of a training budget B and two CMDPs, $M_t = (\mathcal{S}, \mathcal{A}, T, \gamma, R, C, d_t)$ and $M_d = (\mathcal{S}, \mathcal{A}, T, \gamma, R, C, d_d)$, corresponding to training and deployment phases, respectively.

Both CMDPs are identical, except for their cost limits. To accommodate relaxed constraints during training, we assume that $d_{t_i} \geq d_{d_i}$ for all $1 \leq i \leq k$.[1] If the agent is trained using a simulator, we can set $d_{t_i} = \infty$ for all $1 \leq i \leq k$, effectively reducing M_t to an MDP. The objective in RECRL is to find a policy for M_d that optimizes the return subject to the constraint:

$$\pi^* = \underset{\pi \in \Pi(C, d_d)}{\text{argmax}} \; J(\pi)$$

However, during the initial B episodes, the agent operates on M_t and can explore policies within $\Pi(C, d_t)$, which may be more permissive. Existing CRL algorithms can be applied to RECRL problems by enforcing d_d throughout training, potentially applying tighter constraints than required by d_t. In this work, we propose methods that gradually transition the constraints from d_t to d_d, facilitating better exploration compared to direct training on d_d and reducing the risk of getting stuck in local optima.

Some existing approaches for addressing standard CRL problems, formulated as CMDP, often encounter some level of constraint violation in the pursuit of finding a policy that adhere to the specified constraints (e.g., [1, 12, 32, 34, 44]). Nevertheless, the overarching objective of these methods is to rapidly converge toward a policy that satisfies the constraints. In contrast, RECRL is explicitly designed for domains in which it is not necessary to adhere to the constraints during training. In RECRL, both the deployment and training phases are governed by distinct sets of constraints, marking a fundamental departure from the conventional CRL setting. Moreover, the flexibility allowed between constraints during the training and deployment phases can be strategically leveraged to attain superior policies, as demonstrated below.

5 CURRICULUM-BASED RECRL APPROACH

CRL algorithms can learn a valid policy when applied directly on M_d. However, there's an opportunity to adapt them for RECRL, capitalizing on the benefits of training with relaxed constraints. We present a toy domain called `RiverGrid` (based on the popular `MiniGrid` environment [9]), inspired by the scenario of training rescue workers. In this grid-based environment (shown in Figure 1), the agent (depicted as a red triangle) must reach a goal tile (green tile) while adhering to a constraint limiting the number of time steps it can spend in cold river tiles (blue tiles). Figure 1a illustrates an instance where the agent must cross at least five river tiles to reach the goal. When the cost limit is 5, the optimal policy of moving forward only allows the agent to reach the goal while respecting the constraint. However, it is challenging for CRL algorithms to discover this optimal policy. We have tested CPO and PPO-L on this simple example; both fail to find the optimal policy and converge to a policy where the agent remains at its initial position.

In this section, we present a method called Cost-Limit Curriculum (CLiC) to adapt any CRL algorithm and leverage the additional exploration opportunities provided by RECRL. Instead of training the agent on M_t throughout the entire training phase, the agent is presented with a curriculum, i.e., a sequence of models, $\mathcal{M} = M_{t_1} \dots M_{t_n}$, that differ in their cost limits. In this work, we consider CRL agents represented by a set of parameters (e.g., weights of a neural network), i.e., π_θ is an agent that corresponds to a parameter configuration θ. Let $\theta_N^{\mathcal{M}}$ be the parameters of π_θ after training on curriculum $\mathcal{M}$ for N episodes. Given a CRL agent, the objective of a curriculum learning problem for RECRL is to find a curriculum $\mathcal{M}$, to be used by the agent during the training phase, that maximizes the performance of the agent when it is deployed:

$$\mathcal{M}^* = \underset{\mathcal{M}}{\text{argmax}} \; J(\pi_{\theta_B^{\mathcal{M}}}) \quad (1)$$
$$\text{s.t.} \quad \pi_{\theta_B^{\mathcal{M}}} \in \Pi(C, d_d),$$
$$M \in \mathbb{M} \; \forall M \in \mathcal{M}$$

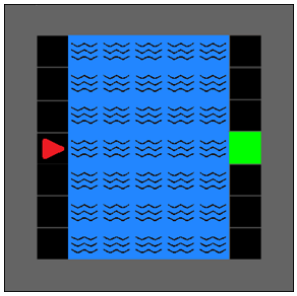
where $\mathbb{M} = \{(\mathcal{S}, \mathcal{A}, T, \gamma, R, C, d) | d_i \leq d_{t_i} \text{ for all } 1 \leq i \leq k\}$ is the set of all possible models with cost limits that are at least as strict as the constraint of M_t. In curriculum learning terms, M_d is the *target task*, and $\mathbb{M}$ is the set of all potential *source tasks*. In addition, the algorithm that generates the curriculum is known as the *teacher*, while the agent that solves each model in the curriculum is called the *student*.

Directly optimizing Eq. 1 is challenging, as the effect of any curriculum can only be accurately measured once the agent has finished training, where the effect of a curriculum is the difference between the initial performance of the agent and its performance after training with the curriculum; this effect is known as *global learning progress* [29]. Consequently, most curriculum learning approaches aim to optimize alternative objectives such as local learning progress [6, 15, 22, 24, 28], intermediate difficulty [13, 14], diversity [27], and surprise [26]. However, these methods are defined for unconstrained RL, making them unsuitable for generating a curriculum by actively managing and controlling the cost limit.

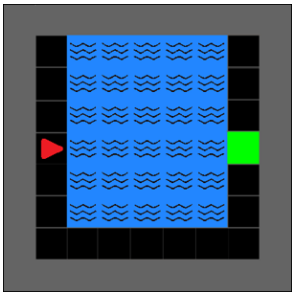
To overcome this limitation, we explore an alternative way to obtain a curriculum by considering sequences with non-ascending cost limits, where the final source task is the target task (M_d), i.e., curricula of the form:

$$\left\{ \mathcal{M} = M_{t_1}, \dots, M_{t_n} \;\middle|\; \begin{array}{l} d_{t_i} \geq d_{t_{1_i}} \geq \dots \geq d_{t_{n_i}} = d_{d_i} \\ \forall 1 \leq i \leq k \end{array} \right\}$$

This formulation enables CRL algorithms to train with progressively tighter constraints, ensuring the final policy adheres to the deployment constraints while promoting better exploration during



(a) An instance in which algorithms can benefit from relaxed exploration; agents can step onto at most 5 river tiles.



(b) An instance in which relaxed exploration might be harmful; agents can step onto at most 1 river tile.

Figure 1: Two instances of the `RiverGrid` domain

[1] We do not consider cases where some constraints could benefit from being tighter during training in this paper.

Algorithm 1 Static Curriculum Teacher

Input: d_t, d_d, B, π, $M = (\mathcal{S}, \mathcal{A}, T, \gamma, R, C, d)$
for episode b from 1 to B **do**
  $p \leftarrow \frac{b}{B}$ // Progress of the training
  for $1 \le i \le k$ **do**
    if exponential curriculum **then**
      Set $d_i \leftarrow \frac{d_{t_i} - d_{d_i}}{1-e^{-1}} \exp(-p) + \frac{d_{d_i} - e^{-1} d_{t_i}}{1-e^{-1}}$.
    else if linear curriculum **then**
      Set $d_i \leftarrow p \cdot d_{d_i} + (1-p) d_{t_i}$.
    else if cosine curriculum **then**
      Set $d_i \leftarrow \cos\left(\frac{\pi \cdot p}{2}\right) d_{t_i} + \left(1 - \cos\left(\frac{\pi \cdot p}{2}\right)\right) d_{d_i}$.
    end if
  end for
  Execute π on M
end for

training (compared to training only with d_d). To generate the curriculum, the teacher must decide which cost limits (source tasks) to present to the agent, and the duration of training on each task.

5.1 Static Curricula

We initially consider *static curricula* (static CLiC), which are predetermined and do not require any runtime information. Three types of static curricula are studied, based on common scheduling strategies: Linear decay (CLiC$_L$), Cosine decay (CLiC$_C$), and Exponential decay (CLiC$_E$). These curricula change the cost limit at each episode based on the fraction of completed training episodes, relative to the training budget B, and the training and deployment costs d_t and d_d, respectively. The pseudocode for the three types of static curricula is shown in Algorithm 1. Note that the decay functions are not well-defined when $d_{t_i} = \infty \neq d_{d_i}$ for some $1 \le i \le k$. Hence, a finite initial cost limit should be used. For instance, the agent can be executed on the training model M_t for a few episodes, and the incurred costs can be used as the initial cost limits. We denote CLiC$_X$-A as the policy obtained by training a CRL algorithm A on the curriculum produced by CLiC$_X$ (where X represents one of the CLiC methods described above). While the static CLiC methods are relatively straightforward, they effectively improve the agent's exploration during training. For example, CLiC$_X$-CPO and CLiC$_X$-PPO-L successfully find the optimal solution in the example depicted in Figure 1a, for all types of static curricula ($X \in \{L, C, E\}$).

5.2 Bounding Constraint Violations

The static CLiC approach has the useful property of bounding the cumulative constraint violation induced by the relaxed exploration when students have bounds on their constraint violation. This property can be important when training is done in a real environment (rather than a simulator). For instance, in the example of training rescue workers, having a bound on the maximal time actually spent in the water throughout training can be important.

To demonstrate how the student's bounds on constraint violation can be utilized to bound the cumulative constraint violation during training with relaxed constraints, we analyze CPO's worst-case constraint violation (WCCV) with respect to a static curriculum D. Let $D(b, i)$ denote the cost limit d_{b_i} assigned to the i-th cost function at episode b. CPO updates the policy at each iteration according to the following objective:

$$\pi_{b+1} = \underset{\pi}{\operatorname{argmax}} \underset{a\sim\pi}{\mathbb{E}} [A^{\pi_b}]$$
$$\text{s.t. } J_{C_i}(\pi_b) + \frac{1}{1-\gamma} \underset{a\sim\pi}{\mathbb{E}} [A^{\pi_b}_{C_i}(s,a)] \le d_{d_i}, \ \forall i$$
$$\bar{D}_{KL}(\pi || \pi_b) \le \delta. \tag{2}$$

where $\bar{D}_{KL}(\pi||\pi_b) = \mathbb{E}_{s\sim\pi_b} D_{KL}(\pi||\pi_b)$, D_{KL} is the KL-divergence of the two policies, $\delta > 0$ is the step size, and $A^{\pi_b}(s,a)$, $A^{\pi_b}_{C_i}(s,a)$ are the advantage functions of the reward and costs, respectively. Achiam et al. [1] analyzed the above policy-update rule and proved the following bound on the worst-case constraint violation of CPO.

LEMMA 1 (CPO WCCV). *Suppose π_b, π_{b+1} are related by (2), and that Π_θ in (2) is any set of policies with $\pi_b \in \Pi_\theta$. An upper bound on the C_i-return of π_{b+1} is*

$$J_{C_i}(\pi_{b+1}) \le d_i + \frac{\sqrt{2\delta}\gamma\epsilon^{\pi_{b+1}}_{C_i}}{(1-\gamma)^2},$$
$$\textit{where } \epsilon^{\pi_{b+1}}_{C_i} = \max_s |\mathbb{E}_{a\sim\pi_{b+1}}[A^{\pi_b}_{C_i}(s,a)]|.$$

Denote by SUM-$J_{C_i} = \sum_{b=1}^{B} J_{C_i}(b)$ the cumulative constraint violations during training in dimension i. SUM-J_{C_i} can be bounded directly by using Lemma 1:

$$\text{SUM-}J_{C_i} = \sum_{b=1}^{B} \left(d_i + \frac{\sqrt{2\delta}\gamma\epsilon^{\pi_{b+1}}_{C_i}}{(1-\gamma)^2} \right) \tag{3}$$

The policy update procedure of CPO when using the curriculum D can be defined by replacing d_i with $D(b,i)$:

$$\pi_{b+1} = \underset{\pi\in\Pi_\theta}{\operatorname{argmax}} \underset{a\sim\pi}{\mathbb{E}} [A^{\pi_b}]$$
$$\text{s.t. } J_{C_i}(\pi_b) + \frac{1}{1-\gamma} \underset{a\sim\pi}{\mathbb{E}} [A^{\pi_b}_{C}(s,a)] \le D(b,i), \ \forall i$$
$$\bar{D}_{KL}(\pi||\pi_b) \le \delta. \tag{4}$$

By considering this adapted update rule, SUM-J_{C_i} can be bounded when using a static curriculum D.

PROPOSITION 1 (CPO CUMULATIVE WCCV WITH CURRICULUM). *Suppose π_b, π_{b+1} are related by (4), that Π_θ in (4) is any set of policies with $\pi_b \in \Pi_\theta$, and that D is a linear curriculum used for training π. An upper bound on SUM-J_{C_i} is*

$$\textit{SUM-}J_{C_i} \le \frac{B(d_{d_i} + d_{t_i})}{2} + \sum_{b=1}^{B} \frac{\sqrt{2\delta}\gamma\epsilon^{\pi_{b+1}}_{C_i}}{(1-\gamma)^2}$$

PROOF. By replacing d_i with the cost limit obtained by the curriculum in Prop. 1, we get:

$$J_{C_i}(\pi_{b+1}) \le D(b,i) + \frac{\sqrt{2\delta}\gamma\epsilon^{\pi_{b+1}}_{C_i}}{(1-\gamma)^2}$$

Consequently, SUM-J_{C_i} can be bounded with respect to any curriculum D as:

$$\text{SUM-}J_{C_i} \le \sum_{b=1}^{B} \left(D(b,i) + \frac{\sqrt{2\delta}\gamma\epsilon^{\pi_{b+1}}_{C_i}}{(1-\gamma)^2} \right)$$

Algorithm 2 Dynamic Curricula Teacher

1: **Input:** d_t, d_d, B, π, $M = (\mathcal{S}, \mathcal{A}, T, \gamma, R, C, d)$, W, ϵ_r, ϵ_c, , ϵ_π.
2: $d \leftarrow d_t$, $last\_change \leftarrow 0$
3: **for** episode b from 1 to B **do**
4: **if** $b \geq 2W$ and $last\_change \geq W$ **then**
5: **if** $|\frac{\bar{r}_{(b-W:b)}}{\bar{r}_{(b-2W:b-W)}} - 1| \leq \epsilon_r$ and $\forall i$, $|\frac{\bar{C}_{i_{(b-W:b)}}}{\bar{C}_{i_{(b-2W:b-W)}}} - 1| \leq \epsilon_c$ **then** // π has converged
6: **if** $\forall i, \bar{C}_{i_{(b-W:b)}} \leq d_i$ **then** // Switch cost limit
7: $\forall i,\ d_i \leftarrow \max(d_{d_i}, \bar{C}_{i_{(b-W:b)}} - \frac{\bar{C}_{i_{(b-W:b)}} - d_{d_i}}{B-b})$
8: **else** // Stuck at a higher cost
9: Increase exploration by adding ϵ_π noise to π
10: **end if**
11: $last\_change \leftarrow 0$
12: **end if**
13: **end if**
14: Execute π on M
15: $last\_change \leftarrow last\_change + 1$
16: **end for**

In a linear curriculum, $\sum_{b=1}^{B} D(b, i) = \frac{B(d_{d_i} + d_{t_i})}{2}$, thus

$$\text{SUM-}J_{C_i} \leq \frac{B(d_{d_i} + d_{t_i})}{2} + \sum_{b=1}^{B} \frac{\sqrt{2\delta}\gamma\epsilon_{C_i}^{\pi_{b+1}}}{(1-\gamma)^2}$$

□

With these bounds, it becomes possible to compute appropriate training cost limits d_t given a budget for overall constraint violations during training. This ensures that the violations remain within the specified bound when using static curricula. While proposition 1 is defined for linear curriculum and cumulative WCCV, similar bounds can be defined for the other static curriculum types and maximal WCCV.

5.3 Dynamic Curricula

Using static curricula offers certain advantages, such as improved exploration and the ability to bound worst-case cost violations during training. However, this approach is not without its limitations, manifesting in three critical aspects.

First (Limitation 1), it neglects the consideration of the student's return, potentially resulting in constant changes to the cost limits without stable policy learning.

Second (Limitation 2), there is a persistent risk of converging to a policy that violates d_d. Take, for instance, the RiverGrid problem illustrated in Figure 1b, where agents can navigate at most one river tile. In this scenario, the policy of navigating around the water allows the agent to reach the goal without breaching the constraint. Nonetheless, some agents that use a cost-limit curriculum, particularly those relying on trust-region methods, may fail to discover this policy even as the cost limits converge to d_d. This limitation became evident in the application of CLiC$_X$-CPO and CLiC$_X$-PPO-L with three static curricula ($X \in L, C, E$) on the specified problem instance. In all cases, the models converged to a policy that violates the deployment constraint. In contrast, CPO and PPO-L, directly trained on the deployment model M_d (without curriculum), successfully identified the optimal constraint-obeying policy.

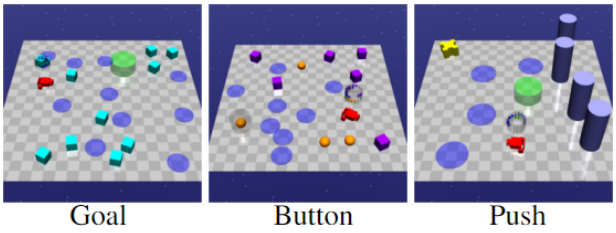


Figure 2: Safety Gym benchmark environments with safety level 2 and a car robot.

Third (Limitation 3), the use of predetermined cost limits in the curriculum may not accurately reflect the student's actual cost experiences. This inaccuracy can result in wasted iterations and a lack of progress toward satisfying d_d.

To mitigate these weaknesses of static CLiC, we introduce a novel teacher capable of generating a dynamic curriculum (CLiC$_D$, Algorithm 2) based on the recent history of the student's experience. Initially, the teacher assigns the student the task corresponding to the training model M_t and observes its experience costs and rewards across two consecutive time windows, each of W episodes (where W is a hyperparameter). Then, at every episode, the teacher assigns the student a new model (corresponding to a set of thresholds d) based on its performance in the previous $2W$ episodes. The teacher first determines whether the student has converged to a policy with respect to the current task by evaluating the ratio between the average reward (and cost) in the last time window and the previous time window (line 5). If these ratios significantly deviate from 1, with respect to two thresholds ϵ_r and ϵ_c, it indicates that the student has not converged to a policy for the current task and will continue facing the same task in the next episode. Therefore, the teacher only assigns a new task to the student once a stable policy is reached, effectively addressing limitation 1.

Once the student has converged to a policy, the teacher checks if the policy adheres to the current task's constraints. If the learned policy does not satisfy the current costs, the teacher introduces noise to the student's policy to encourage exploration and escape possible local optima (lines 8-9), thereby mitigating limitation 2. Finally, if the student has converged to a policy that adheres to the constraints, the teacher introduces the student to a new task. In this new task, the cost limits are gradually reduced in each dimension toward d_d, taking into account the average experienced cost in the last window and the remaining training budget (lines 6-7).

In the context of the problem instances depicted in Figure 1b, both CLiC$_D$-CPO and CLiC$_D$-PPO-L outperform their static CLiC counterparts by successfully learning the optimal policy. This notable improvement underscores the dynamic curriculum's potential to effectively mitigate the limitations of static curricula, all the while encouraging additional exploration.

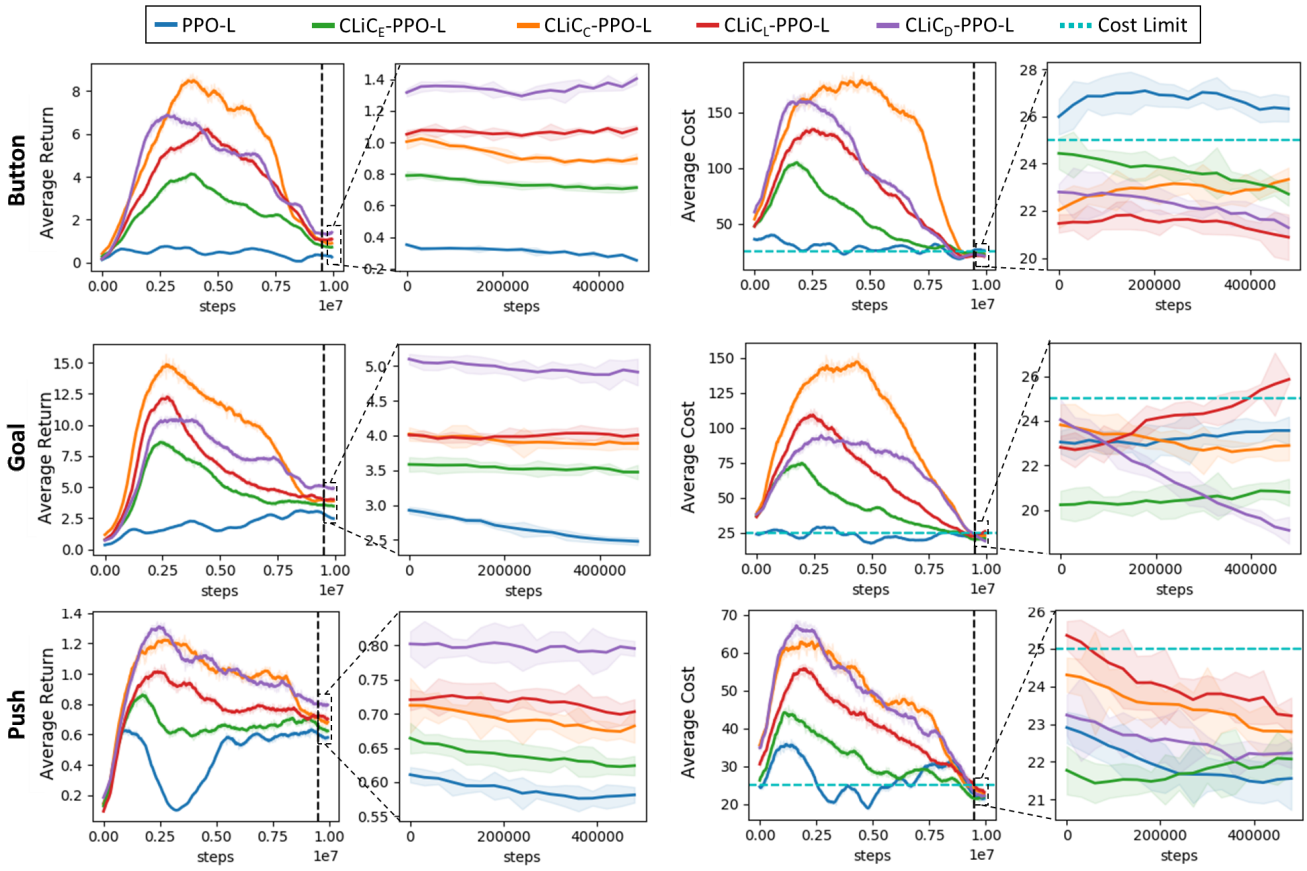


Figure 3: Evaluation of the Static and Dynamic Curricula applied to PPO-L on the Safe-RL benchmark.

6 EMPIRICAL EVALUATION

Experimental Setting. We conducted experiments to assess the impact of different CLiC methods on the learned policies in RECRL instances using the Safety Gym benchmark [32]. The benchmark involves a robot operating in environments with unsafe elements, and the goal is to achieve specific objectives while avoiding unsafe outcomes. We focused on the *car* robot, a wheeled robot with differential drive control, and considered three types of environments and objectives: pushing a yellow box into a green area (Push), moving the robot to the green area (Goal), and pressing a highlighted button (Button). Unsafe outcomes include entering dangerous areas, touching dangerous objects (movable or immovable, stationary or moving), and pressing the wrong button (Button environment). At each step, a binary cost function indicates whether any unsafe outcome occurred. We evaluated the scenarios with level 2 safety, where environments are densely populated with unsafe elements and safety constraints can hinder exploration.

To test our hypothesis that combining CRL algorithms with a cost-limit curriculum enhances policy learning, we integrated CLiC methods with CPO and PPO-L. We compared the performance of policies learned using the curriculum to those trained solely on deployment constraints. As explained previously, we use CPO and PPO-L as representative CRL algorithms. In principle, CLiC can be combined with any CRL algorithm.

Each experiment involved the agent interacting with the environment for $1e7$ steps. The first 95% of the steps constituted the training phase, where the agent was trained with relaxed constraints. The last 5% of the steps (500k) represented the deployment phase, where the agent had to adhere to the full set of constraints.

To obtain statistically significant results, each combination of CLiC method, environment, and CRL algorithm was executed five times with different seeds. The training was parallelized on five servers, each equipped with four A40 GPUs, Intel(R) Xeon(R) Gold 6342 CPU, 500 GB of RAM, and 3.6 TB of disk space. The implementations of all environments and CRL algorithms were directly taken from the publicly available Safety Gym repository [32], along with the corresponding hyperparameters and environment configurations.

For CLiC$_D$, the hyperparameters determining when the policy has converged (ϵ_r and ϵ_c) were set to 0.1 after brief tuning. The performance differences observed during tuning had a maximal difference of 11%. Therefore, the reported trends are relatively robust to the chosen parameter values.

Results. The results for PPO-Lagrangian are presented in Figure 3, while those for CPO are available in Figure 4. Each row in the

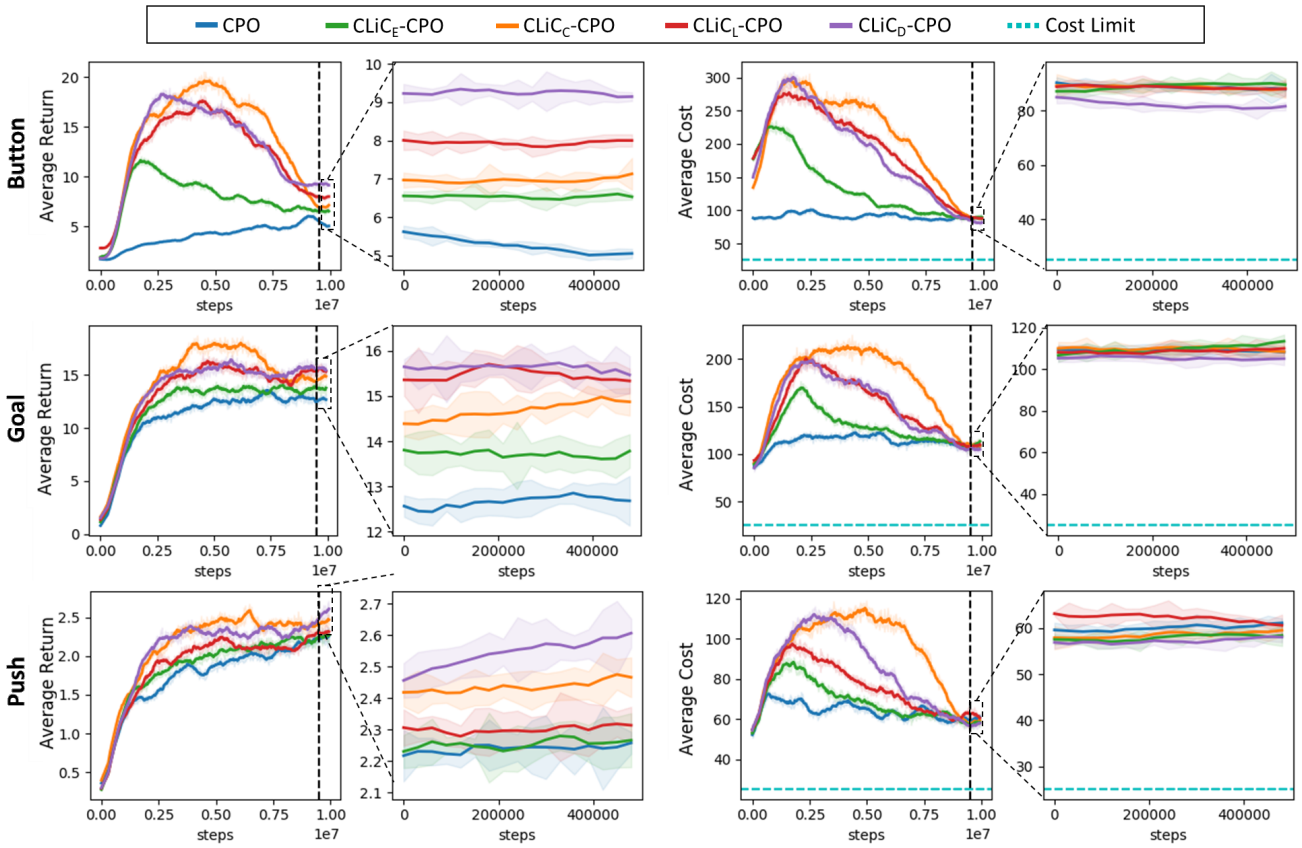


Figure 4: Evaluation of the Static and Dynamic Curricula applied to CPO on the Safe-RL benchmark.

figures corresponds to a different task (Button, Goal, Push), and the columns display the returns (first column) and costs (third column) during training and deployment. The second and fourth columns provide a magnification of the deployment phase (500k steps). The solid lines represent the mean values of returns or costs, the dashed horizontal line (cyan) indicates the deployment cost limit (d_d), and the dashed vertical line (black) marks the transition between the training phase and the deployment phase. The shaded areas in the plots indicate the standard error.

The results demonstrate that the CLiC methods effectively improve student performance, yielding policies with better returns while incurring similar costs. In fact, all CLiC approaches consistently outperformed the students by the end of the training phase across all tasks. Among the static CLiC methods, CLiC$_E$ showed the smallest return improvement, ranging from 10% to 300% for PPO-L and 3% to 30% for CPO. It also induced the least cost overhead during training. CLiC$_L$ achieved the best performance among the static curricula in two out of the three tasks and had the second-smallest overhead, resulting in a return improvement ranging from 20% to 600% for PPO-L (CLiC$_L$-PPO-L) and 5% to 60% for CPO (CLiC$_L$-CPO). However, CLiC$_C$ had the highest overhead among the three static curricula and showed superiority in just one task. Nonetheless, it improved PPO-L by 20% to 400% and CPO by 10% to 40%.

Notably, CLiC$_D$ outperformed all static curricula in terms of return across all tasks. It learned policies that were significantly better than the base algorithms, improving by 40% to 900% over PPO-L and 15% to 80% over CPO. Additionally, the CLiC$_D$ incurred a cost overhead during training comparable to the Linear CLiC.

Finally, it is important to note that CLiC's ability to learn a policy that satisfies the final constraints depends on the student. CPO is not guaranteed to adhere to the constraints and often does not in practice. Consequently, the CLiC variants of CPO also learned policies that violated the constraint similarly. In contrast, PPO-L learned policies that satisfied the constraints, as did its CLiC variants.

Generated Curricula. The curricula generated by each of the approaches for the different tasks can be found in Figure 5. Naturally, the static CLiC methods result in the same curriculum for all tasks and algorithms, as they are solely dependent on the training and deployment cost limits (d_t and d_d, respectively). In contrast, the dynamic CLiC method results in different curricula for each underlying algorithm and task, as well as for each individual run.

7 CONCLUSION AND FUTURE WORK

In this work, we introduced RECRL, a constrained reinforcement learning (CRL) setting that is designed for agents that have the ability to train with more lenient constraints than during deployment.

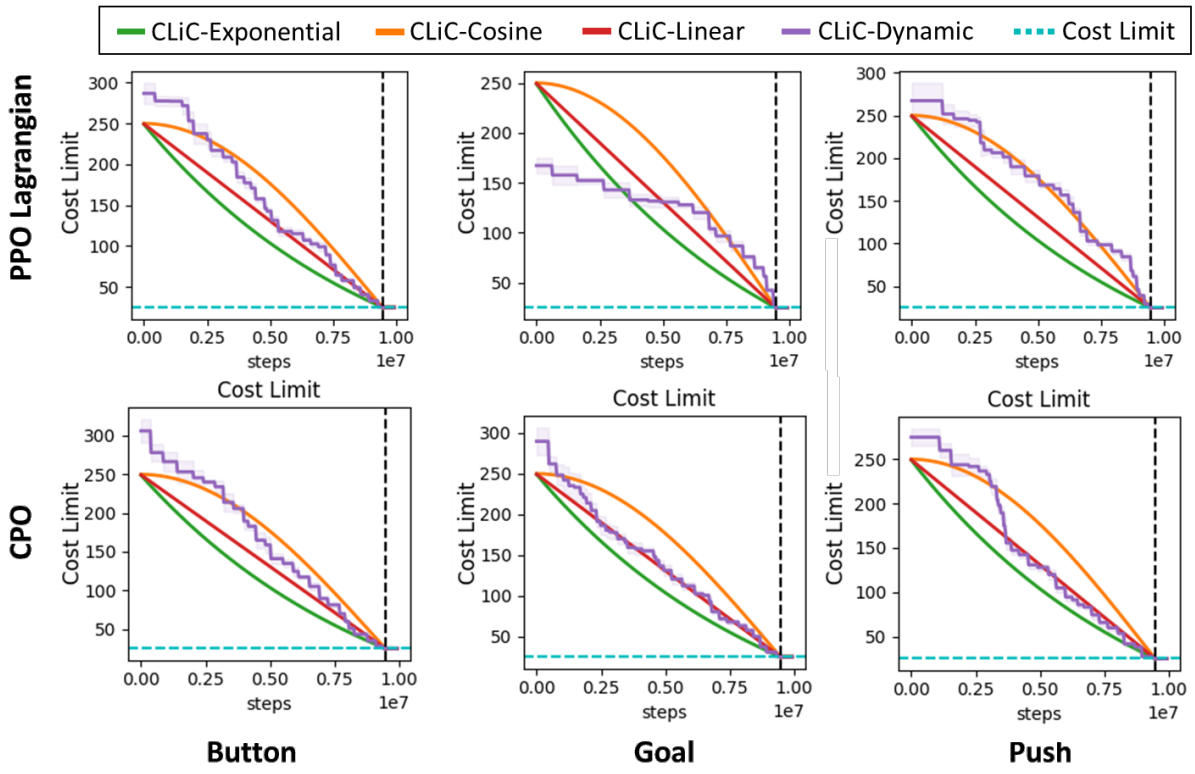


Figure 5: Cost-limit curricula generated by the different approaches.

In such settings, the distinctions between training constraints and deployment constraints foster superior exploration compared to standard CL, enabling the acquisition of high-return policies while maintaining adherence to deployment constraints. In addition, we presented CLiC, a curriculum-based approach, which enhances existing CRL algorithms by leveraging the RECRL framework. We explored static and dynamic curricula, enhancing the performance of CPO and PPO-L on Safery Gym benchmarks. All CLiC methods showcased improvements over their base algorithms, with dynamic CLiC achieving significant performance gains. However, CLiC's effectiveness relies on the student's ability to find constraint-respecting policies. In addition, the proposed CLiC methods only consider curricula in which the cost limits are in non-increasing order, a limitation that could be addressed in future work. In particular, incorporating meta-curriculum-learning models [30, 43] may efficiently learn cost-limit curricula based on experience from similar tasks. Moreover, enhancing results may be achieved by combining CLiC-based methods with other curriculum learning mechanisms. For instance, in the framework introduced by Turchetta et al. [40] for safety-oriented learning, CLiC could serve as a meta-teacher to determine the current cost limit (and noise), complementing the methodology outlined in Turchetta et al.'s work to more effectively acquire a safe policy within the given cost constraints. Finally, combining CLiC with Sim2Real approaches could be beneficial for transferring policies trained on simulators to physical robots.

ACKNOWLEDGMENTS

This collaboration involves Ben-Gurion University (BGU) and the Learning Agents Research Group (LARG) at UT Austin. The work at BGU was supported by the Israel Science Foundation (ISF) grant #909/23 and by Israel's Ministry of Innovation, Science and Technology (MOST) grant #1001706842, awarded to Shahaf Shperberg. LARG research is supported in part by NSF (FAIN-2019844, NRT-2125858), ONR (N00014-18-2243), ARO (E2061621), Bosch, Lockheed Martin, and UT Austin's Good Systems grand challenge. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

REFERENCES

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained Policy Optimization. In *ICML (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 22–31.
[2] Eitan Altman. 1999. *Constrained Markov decision processes*. Vol. 7. CRC Press.
[3] Yarden As, Ilnura Usmanova, Sebastian Curi, and Andreas Krause. 2022. Constrained policy optimization via bayesian world models. *arXiv preprint arXiv:2201.09802* (2022).
[4] Yarden As, Ilnura Usmanova, Sebastian Curi, and Andreas Krause. 2022. Constrained Policy Optimization via Bayesian World Models. In *ICLR*. OpenReview.net.
[5] Minoru Asada, Shoichi Noda, Sukoya Tawaratsumida, and Koh Hosoda. 1996. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine learning* 23, 2 (1996), 279–303.
[6] Adrien Baranes and Pierre-Yves Oudeyer. 2013. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics Auton. Syst.* 61, 1 (2013), 49–73.

[7] Dominik Baumann, Alonso Marco, Matteo Turchetta, and Sebastian Trimpe. 2021. GoSafe: Globally Optimal Safe Robot Learning. In *ICRA*. IEEE, 4452–4458.

[8] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. 41–48.

[9] Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. 2018. Minimalistic Gridworld Environment for OpenAI Gym. https://github.com/maximecb/gym-minigrid.

[10] Yinlam Chow, Ofir Nachum, Edgar A. Duéñez-Guzmán, and Mohammad Ghavamzadeh. 2018. A Lyapunov-based Approach to Safe Reinforcement Learning. In *NeurIPS*. 8103–8112.

[11] Houston Claure, Yifang Chen, Jignesh Modi, Malte F. Jung, and Stefanos Nikolaidis. 2019. Reinforcement Learning with Fairness Constraints for Resource Distribution in Human-Robot Teams. *ArXiv* abs/1907.00313 (2019).

[12] Davide Corsi, Raz Yerushalmi, Guy Amir, Alessandro Farinelli, David Harel, and Guy Katz. 2022. Constrained Reinforcement Learning for Robotics via Scenario-Based Programming. *CoRR* abs/2206.09603 (2022).

[13] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. 2018. Automatic Goal Generation for Reinforcement Learning Agents. In *ICML*.

[14] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. 2017. Reverse curriculum generation for reinforcement learning. In *CoRL*. PMLR, 482–495.

[15] Sébastien Forestier, Yoan Mollard, and Pierre-Yves Oudeyer. 2017. Intrinsically Motivated Goal Exploration Processes with Automatic Curriculum Learning. *CoRR* abs/1708.02190 (2017).

[16] Javier García and Fernando Fernández. 2012. Safe Exploration of State and Action Spaces in Reinforcement Learning. *J. Artif. Intell. Res.* 45 (2012), 515–564.

[17] Javier García and Fernando Fernández. 2015. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.* 16 (2015), 1437–1480.

[18] Yoshinobu Kadota, Masami Kurano, and Masami Yasuda. 2006. Discounted Markov decision processes with utility constraints. *Comput. Math. Appl.* 51, 2 (2006), 279–284.

[19] Johannes Kirschner, Mojmir Mutnỳ, Andreas Krause, Jaime Coello de Portugal, Nicole Hiller, and Jochem Snuverink. 2022. Tuning Particle Accelerators with Safety Constraints using Bayesian Optimization. *arXiv preprint arXiv:2203.13968* (2022).

[20] Nevena Lazic, Craig Boutilier, Tyler Lu, Eehern Wong, Binz Roy, M. K. Ryu, and Greg Imwalle. 2018. Data center cooling using model-predictive control. In *NeurIPS*. 3818–3827.

[21] Yongshuai Liu, Jiaxin Ding, and Xin Liu. 2020. IPO: Interior-Point Policy Optimization under Constraints. In *AAAI*. AAAI Press, 4940–4947.

[22] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. 2020. Teacher-Student Curriculum Learning. *IEEE Trans. Neural Networks Learn. Syst.* 31, 9 (2020), 3732–3740.

[23] Teodor Mihai Moldovan and Pieter Abbeel. 2012. Safe Exploration in Markov Decision Processes. In *ICML*. icml.cc / Omnipress.

[24] Siddharth Mysore, Robert Platt, and Kate Saenko. 2019. Reward-guided curriculum for robust reinforcement learning. *preprint* (2019).

[25] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. 2020. Curriculum learning for reinforcement learning domains: A framework and survey. *arXiv preprint arXiv:2003.04960* (2020).

[26] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. 2017. Curiosity-driven Exploration by Self-supervised Prediction. In *ICML*.

[27] Vitchyr H. Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. 2019. Skew-Fit: State-Covering Self-Supervised Reinforcement Learning. *CoRR* abs/1903.03698 (2019).

[28] Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. 2020. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In *Conference on Robot Learning*. PMLR, 835–853.

[29] Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. 2020. Automatic curriculum learning for deep rl: A short survey. *arXiv preprint arXiv:2003.04664* (2020).

[30] Rémy Portelas, Clément Romac, Katja Hofmann, and Pierre-Yves Oudeyer. 2020. Meta automatic curriculum learning. *arXiv preprint arXiv:2011.08463* (2020).

[31] Sébastien Racanière, Andrew K. Lampinen, Adam Santoro, David P. Reichert, Vlad Firoiu, and Timothy P. Lillicrap. 2020. Automated curriculum generation through setter-solver interactions. In *ICLR*.

[32] Alex Ray, Joshua Achiam, and Dario Amodei. 2019. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708* 7 (2019), 1.

[33] R Tyrrell Rockafellar, Stanislav Uryasev, et al. 2000. Optimization of conditional value-at-risk. *Journal of risk* 2 (2000), 21–42.

[34] Julien Roy, Roger Girgis, Joshua Romoff, Pierre-Luc Bacon, and Christopher J. Pal. 2022. Direct Behavior Specification via Constrained Reinforcement Learning. In *ICML (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 18828–18843.

[35] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952* (2015).

[36] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[37] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum Learning: A Survey. *Int. J. Comput. Vis.* 130, 6 (2022), 1526–1565.

[38] Bhavya Sukhija, Matteo Turchetta, David Lindner, Andreas Krause, Sebastian Trimpe, and Dominik Baumann. 2022. Scalable Safe Exploration for Global Optimization of Dynamical Systems. *CoRR* abs/2201.09562 (2022).

[39] Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. 2019. Reward Constrained Policy Optimization. In *ICLR (Poster)*.

[40] Matteo Turchetta, Andrey Kolobov, Shital Shah, Andreas Krause, and Alekh Agarwal. 2020. Safe reinforcement learning via curriculum induction. *Advances in Neural Information Processing Systems* 33 (2020), 12151–12162.

[41] Yuxin Wu and Yuandong Tian. 2017. Training Agent for First-Person Shooter Game with Actor-Critic Curriculum Learning. In *ICLR (Poster)*. OpenReview.net.

[42] Xuesu Xiao, Bo Liu, Garrett Warnell, and Peter Stone. 2021. Toward agile maneuvers in highly constrained spaces: Learning from hallucination. *IEEE Robotics and Automation Letters* 6, 2 (2021), 1503–1510.

[43] Zifan Xu, Yulin Zhang, Shahaf S Shperberg, Reuth Mirsky, Yuqian Jiang, Bo Liu, and Peter Stone. 2022. Model-Based Meta Automatic Curriculum Learning. In *Decision Awareness in Reinforcement Learning Workshop at ICML 2022*.

[44] Qisong Yang, T Simão, Nils Jansen, S Tindemans, and M Spaan. 2022. Training and transferring safe policies in reinforcement learning. (2022).