Prioritizing safety via curriculum learning

Anonymous authors
Paper under double-blind review

Abstract

Curriculum learning for reinforcement learning (RL) aims to accelerate learning by generating sequences of tasks of increasing difficulty. Besides its sample-efficiency benefits, curriculum learning has the potential to address safety-critical settings where an RL agent must adhere to safety constraints. However, existing curriculum generation approaches still overlook such constraints and thus propose tasks that cause RL agents to violate safety constraints during training and behave sub-optimally after. We propose a safe curriculum generation approach (SCG) that aligns the objectives of constrained RL and curriculum learning: improving safety during training and boosting learning speed. SCG generates sequences of tasks where the RL agent can be both safe and performant by initially preferring tasks with minimum safety violations over high-reward ones. In constrained RL environments, we empirically show that compared to the state-of-the-art curriculum learning approaches and their naively modified safe versions, SCG achieves optimal performance and the lowest amount of constraint violations during training.

1 Introduction

Curriculum learning for reinforcement learning (RL) aims to design task sequences that boost RL agents' performance and speed of convergence (Narvekar et al., 2020). A common strategy in curriculum design is to start with easy tasks and adjust the difficulty toward the target tasks as the RL agent collects higher rewards. Manually tailored curricula require human feedback to categorize tasks into easy and hard (Narvekar et al., 2020). Automating curriculum generation alleviates this necessity and increases sample-efficiency benefits for RL in wide-ranging environments (Baranes & Oudeyer, 2010; Florensa et al., 2018; Jiang et al., 2021b).

In addition to improving sample efficiency, curriculum learning has the potential to mitigate the safety challenges faced by RL. In safety-critical settings, unconstrained exploration in RL during training may cause unsafe and undesirable behaviors (Kendall et al., 2019). A curriculum can address this issue by prioritizing tasks with no or low potential for harm so that an RL agent can learn how to accomplish a task without behaving unsafely (Turchetta et al., 2020). For example, for an RL agent learning how to drive, a curriculum can propose a traffic scene without cars and pedestrians to minimize the risk of accidents early on during the training.

Constrained RL addresses safety-critical scenarios where, given a safety threshold, a constraint on the cost function characterizes whether the agent behavior is safe (Altman, 1999). A constrained RL agent aims to maximize its reward while satisfying its cost constraint(Achiam et al., 2017). In the constrained RL framework, a common metric for safety violations during training is the *constraint violation regret*, i.e., accumulated excess cost over the safety threshold (Efroni et al., 2020). As state-of-the-art curriculum learning approaches overlook the constrained RL problem, they fail to consider the cost constraint and propose tasks that yield not only high rewards but also high costs. Such curricula lead to repeated constraint violations during training, as the curriculum generator cannot distinguish whether the agent behaves safely. Therefore, we argue that a safe automated curriculum generation method should minimize constraint violation regret while accelerating learning.

A naive combination of an off-the-shelf curriculum learning approach and a constrained RL algorithm fails to minimize constraint violation regret due to their *misaligned objectives*. A standard curriculum learning method aims to help an RL agent achieve higher rewards faster. In comparison, a constrained RL algorithm searches for policies that primarily satisfy the cost constraint while maximizing reward as much as possible. Given such a combination, the curriculum generator can propose a task that allows the agent to collect high rewards and simultaneously costs higher than the safety threshold, which violates the constraint. To tackle this misalignment, a curriculum learning approach should prioritize tasks where the agent can be performant and safe.

We develop a *safe* curriculum generation method (SCG) (see Fig. 1) that improves performance, accelerates learning, and minimizes safety violations during training. Inspired by CURROT (Klink et al., 2022), given a distribution over target tasks, SCG generates a sequence of task distributions, that allows the current policy to collect higher rewards than a performance threshold and lower costs than a safety threshold. In the initial stages of the training, SCG prioritizes safety over performance by proposing tasks where the agent satisfies the cost constraint. Once the agent behaves safely in all possible tasks under the current distribution, SCG shifts its focus to satisfying the performance constraint. After the agent becomes performant in all contexts in the current support, SCG generates task distributions that approach the target distribution by equally treating safety and performance until the end of the training.

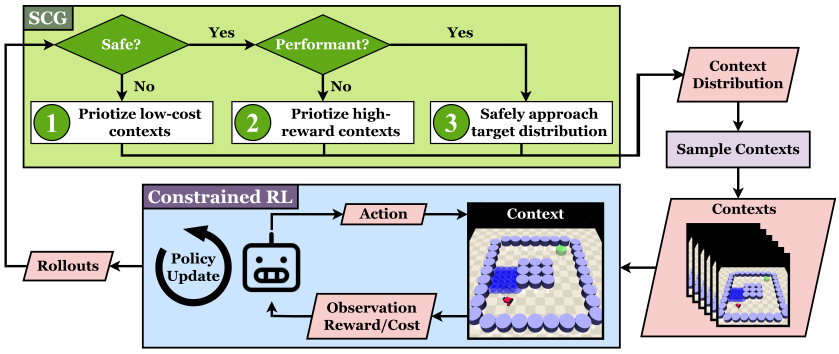


Figure 1: SCG initially prioritizes low-cost contexts to minimize safety violations, then, focuses on high-reward contexts to boost performance. Finally, SCG approaches the target distribution by treating them equally.

Contribution. Our contribution is three-fold: 1) We describe how existing curriculum learning approaches fail to learn an optimal behavior in a constrained environment safely, 2) propose **S**afe **C**urriculum **G**eneration (SCG), an automated curriculum learning approach developed for constrained RL to boost learning speed and minimize constraint violation regret, and lastly 3) our empirical results evidence that, compared to the state-of-the-art curriculum generation approaches and their naively modified versions that account for safety, SCG achieves optimal behavior with the lowest constraint violation regret in constrained RL environments.

2 Related Work

Existing curriculum learning approaches overlook the safety aspect of RL. However, there are methods akin to curriculum learning to ensure safety during training. Wang et al. (2022) develop a curriculum-guided RL approach for real-time bidding systems that relaxes cost constraints to incentivize safe policies early on during training. Eysenbach et al. (2018) learn a reset policy that interferes with the training to prevent the agent from entering dangerous states. Similarly, Turchetta et al. (2020) learns a curriculum policy that chooses an intervention that takes the agent to a safe state if it enters a trigger state. In comparison, existing automated curriculum generation methods do not interfere with the interactions between the student and the environment but only assume that a teacher can set the environment configuration for which the agent learns an optimal behavior (Florensa et al., 2017; 2018; Portelas et al., 2020; Jiang et al., 2021a;b; Klink et al., 2020a;b; 2021; 2022). Similarly, SCG does not assume control over environment dynamics, even when the student violates the cost constraint. SCG also considers a general formulation of constrained environments compared to Turchetta et al.'s specialized notion of cost. Appendix A discusses the literature on automated curriculum generation and constrained RL in detail.

3 Background and Problem Statement

We formulate the environments of interest as contextual constrained Markov decision processes to model a constrained multi-task setting given a distribution over target contexts.

3.1 Contextual Constrained MDPs

Definition 3.1. *We define a* contextual constrained Markov decision process *(CCMDP)* $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{X}, \mathsf{M}, D, \gamma \rangle$ *with a state space* $\mathcal{S}$*, an action space* $\mathcal{A}$*, a context space* $\mathcal{X} \subseteq \mathbb{R}^n$ *for* $n \in \mathbb{Z}^+$*, a mapping from context space to constrained Markov decision process parameters* M*, a safety threshold* $D \in \mathbb{R}_{\geq 0}$*, and a discount factor* $\gamma \in [0, 1]$*.*

A CCMDP $\mathcal{M}$ represents a family of constrained MDPs parameterized by its contexts $\mathbf{x} \in \mathcal{X}$. A context $\mathbf{x}$ provides a constrained MDP $\mathsf{M}(\mathbf{x}) = \langle \mathcal{S}, \mathcal{A}, p_{\mathbf{x}}, r_{\mathbf{x}}, c_{\mathbf{x}}, p_{0,\mathbf{x}}, \gamma \rangle$, where $\mathcal{S}$, $\mathcal{A}$, and γ are the same as in $\mathcal{M}$, but its probabilistic transition function $p_{\mathbf{x}} \colon \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward function $r_{\mathbf{x}} \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, cost function $c_{\mathbf{x}} \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geq 0}$, and initial state distribution $p_{0,\mathbf{x}} \in \Delta(\mathcal{S})$ depend on its context $\mathbf{x}$. A policy $\pi \colon \mathcal{S} \times \mathcal{A} \times \mathcal{X} \to \Delta(\mathcal{A})$, which defines the behavior of an agent in a CCMDP $\mathcal{M}$, outputs a probability simplex over action space $\mathcal{A}$ given $\mathbf{s} \in \mathcal{S}$ and $\mathbf{x} \in \mathcal{X}$. Note that the agent observes the context $\mathbf{x}$. Following policy π, an agent collects a trajectory $\boldsymbol{\tau}_{\mathbf{x}} = \{(\mathbf{s}_t, \mathbf{a}_t, r_t, c_t)\}_{t=0}^{T}$ of length T with an initial state $\mathbf{s}_0 \sim p_{0,\mathbf{x}}$, states $\mathbf{s}_{t+1} \sim p_{\mathbf{x}}(\cdot|\mathbf{s}_t, \mathbf{a}_t)$, actions $\mathbf{a}_t \sim \pi(\cdot|\mathbf{s}_t, \mathbf{x})$, rewards $r_t = r_{\mathbf{x}}(\mathbf{s}_t, \mathbf{a}_t)$, and costs $c_t = c_{\mathbf{x}}(\mathbf{s}_t, \mathbf{a}_t)$ at time steps $t \in [T]$. Resulting in a discounted cumulative reward $G_r(\boldsymbol{\tau}_{\mathbf{x}}) = \sum_{t=0}^{T} \gamma^t r_t$ and a discounted cumulative cost $G_c(\boldsymbol{\tau}_{\mathbf{x}}) = \sum_{t=0}^{T} \gamma^t c_t$.

Given a CCMDP $\mathcal{M}$ and a target context distribution φ, i.e., a probability simplex $\Delta(\mathcal{X})$, *contextual constrained RL* aims to maximize expected return subject to a cost constraint:

$$\pi^* \doteq \arg\max_{\pi} \mathbb{E}_{\varphi}[V_r^{\pi}(\mathbf{x})], \quad \text{s.t.} \quad \mathbb{E}_{\varphi}\left[V_c^{\pi}(\mathbf{x})\right] \leq D, \tag{1}$$

where $V_r^{\pi} = \mathbb{E}_{\pi, p_{\mathbf{x}}, p_{0,\mathbf{x}}}[G_r(\boldsymbol{\tau}_{\mathbf{x}})]$ and $V_c^{\pi} = \mathbb{E}_{\pi, p_{\mathbf{x}}, p_{0,\mathbf{x}}}[G_r(\boldsymbol{\tau}_{\mathbf{x}})]$ are the expected discounted cumulative reward and cost, respectively, induced by policy π in context $\mathbf{x}$ drawn from φ.

Figure 2 shows *safety-goal*, a CCMDP we study. Safety threshold D determines how much the agent needs to avoid the hazards. An episode terminates when the agent reaches the goal. The state is the LIDAR output for the goal, hazards, and columns. The action consists of forces applied to wheel actuators. The target context distribution is uniform over the top section inside the wall.

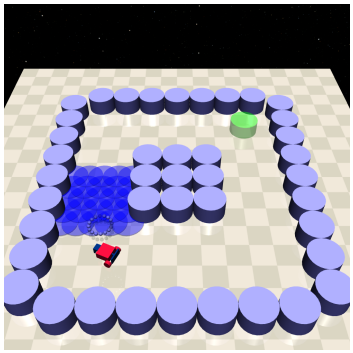


Figure 2: *Safety-goal* in safety-gymnasium (Ji et al., 2023a): A context specifies the position of the goal and its tolerance, i.e., the distance to the goal for success. The agent starts on the bottom left and must reach the goal (green) by avoiding the hazards (navy blue) where it receives a cost.

3.2 Contextual Constrained RL

Contextual-constrained RL (CCRL) is an online multi-task-constrained RL framework that does not assume access to the transition, reward, and cost functions. As the optimal policy maximizes the expected discounted cumulative reward while being safe, i.e., satisfying a cost constraint, a CCRL algorithm should focus on sample efficiency as well as safety. To measure safety, we use *constraint violation regret*, which is the difference between the safety threshold and the value of a learned policy (Efroni et al., 2020). Given that a CCRL algorithm runs for L-many episodes during training, we define the training regret of an algorithm Λ as

$$\mathrm{Reg}^{tr}(L, \{\varrho_l\}_{l=1}^{L}, D) \doteq \sum_{l=1}^{L} \left[\mathbb{E}_{\varrho_l}[V_c^{\pi_l}(\mathbf{x})] - D\right]_+, \tag{2}$$

where $[y]_+ = \max\{y, 0\}$, π_l refers to the policy at the l^{th} update of Λ, and ϱ_l is the context distribution from which $\mathbf{x}$ is drawn at episode l. The regret is non-zero only when the expected

discounted cumulative cost is larger than the safety threshold D. Thus, training regret Reg^{tr} only considers the *safety violations* of an algorithm Λ with respect to context distributions $\{\varrho_l\}_{l=1}^{L}$. Once Λ converges to an optimal policy, its training regret converges, as well. Our problem of interest is not only to learn an optimal policy but to achieve it with the minimum constrained violation regret.

Problem statement. Given a CCMDP $\mathcal{M}$ to describe the parameterization of a set of constrained tasks, and a target context distribution φ to specify their probability of occurrence, generate a sequence of context distributions $\{\varrho_l\}_{l=1}^{L}$ that allow an RL agent to *sample-efficiently* learn an optimal policy (1) with minimal constraint violation regret (2).

Traditionally, a curriculum learning approach generates the sequence of context distributions $\{\varrho_l\}_{l=1}^{L}$, while a non-curriculum approach draws contexts directly from the target context distribution. Thus, curriculum learning approaches can choose a context distribution ϱ_l prioritizing contexts with low expected cost $V_c^{\pi}(\mathbf{x})$ to minimize constraint violation regret.

4 Curriculum Learning and Constrained RL

Now, we present a curriculum learning algorithm and discuss its limitations for constrained RL.

4.1 Curricula via Optimal Transport

Curricula via Optimal Transport (CURROT, Klink et al., 2022) is an automated curriculum generation method that, given a target context distribution φ, creates a sequence of context distributions $\{\varrho_k\}_{k=0}^{K}$ to obtain an optimal policy for a contextual MDP Hallak et al. (2015) $\tilde{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, \mathcal{X}, \tilde{\mathsf{M}}, \gamma \rangle$. Compared to a contextual constrained MDP, a contextual MDP $\tilde{\mathcal{M}}$ does not have a *cost* function, as $\tilde{\mathsf{M}}(\mathbf{x}) = \langle \mathcal{S}, \mathcal{A}, p_{\mathbf{x}}, r_{\mathbf{x}}, p_{0,\mathbf{x}}, \gamma \rangle$. An optimal policy π^* in a contextual MDP $\tilde{\mathcal{M}}$ only maximizes the expected discounted cumulative reward, i.e., $\pi^* \doteq \arg\max_{\pi} \mathbb{E}_{\varphi}[V_r^{\pi}(\mathbf{x})]$.

At curriculum iteration $k \in [K]$, CURROT draws contexts $\{\mathbf{x}_i\}_{i=0}^{M}$ from context distribution ϱ_{k-1}, and collects trajectories $\mathcal{D}_k = \{\boldsymbol{\tau}_{\mathbf{x}_i}\}_{i=1}^{M}$, where $\boldsymbol{\tau}_{\mathbf{x}_i} = \{(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}, r_{i,t}, \mathbf{s}_{i,t+1})\}_{t=0}^{|\boldsymbol{\tau}_{\mathbf{x}_i}|}$. Then, an RL algorithm updates policy π_{k-1} via $\mathcal{D}_k$. CURROT generates the next context distribution via

$$\arg\min_{\varrho} \; \mathcal{W}_2(\varrho, \varphi) \quad \text{s.t.} \quad \varrho(\mathbf{x}) > 0 \Rightarrow V_r^{\pi_k}(\mathbf{x}) \geq \zeta, \forall \mathbf{x} \in \mathcal{X}, \text{ and } \mathcal{W}_2(\varrho, \varrho_+) \leq \epsilon, \tag{3}$$

where $\mathcal{W}_2(\cdot, \cdot)$ is the Wasserstein distance and ϱ_+ is a particle-based distribution based on contexts with return $G_r(\boldsymbol{\tau}_{\mathbf{x}})$ higher than performance threshold ζ, which a buffer of successful contexts ($\mathcal{B}_+$) keeps. A failure buffer ($\mathcal{B}_-$) in parallel maintains the remaining contexts. Furthermore, CURROT estimates $V_r^{\pi_k}$ based on $\mathcal{B}_-$ and $\mathcal{B}_+$. The constraint on $V_r^{\pi_k}$ ensures that the next distribution will have support over contexts where the agent collects sufficiently high rewards. The Wasserstein distance constraint avoids diverging from the current successful contexts since such a scenario can cause performance loss. For more details, we refer the reader to Klink et al. (2022).

4.2 Failure of Curricula to Ensure Safety

The state-of-the-art curriculum learning methods, e.g., CURROT focus on the standard multi-task RL problem, i.e., maximizing $\mathbb{E}_{\varphi}[V_r^{\pi}(\mathbf{x})]$. These approaches fail to address the constrained contextual RL problem, and output curricula that prioritize contexts $\mathbf{x} \sim \varrho_k$ where policy π_k achieves high $V_r^{\pi}(\mathbf{x})$ but violate the constraint on $V_c^{\pi}(\mathbf{x})$.

Imagine the safety-goal environment where safety threshold $D = 0$ and the initial *easy* contexts are around the bottom left corner. As the agent improves, CURROT will generate a context distribution closer to the target distribution. However, as CURROT minimizes Wasserstein distance, it will move its context distribution over the hazards. Although such contexts result in goals closer to the initial position, hence high V_r^{π}, they can cause high V_c^{π} and constraint violations. The agent will choose to pass through the hazards or stay out. The former scenario results in unsafe behavior that can reach the goal with high costs, whereas the latter yields a failed conservative behavior.

Figure 3 demonstrates curricula generated by CURROT in *safety-maze*, a constrained environment similar to safety-goal, but has simpler dynamics. Starting from the bottom left corner (green), the agent needs to avoid the hazards (red) and reach the goal. CURROT moves the particles, sampled from the context distributions $\{\varrho_k\}_{k=0}^{K}$, from the bottom row towards the target context distribution (top white row). As CURROT ignores the cost, it places goals mostly over the red region in the early stages of the training, which causes the aforementioned suboptimal behaviors.

Such scenarios are not unique to CURROT. They occur under curriculum learning algorithms that overlook the constrained nature of a safety-critical setting. Therefore, to assure safety, a curriculum learning algorithm should have its objective aligned with the constrained RL problem. However, by construction, existing approaches suffer from misaligned objectives in constrained RL.

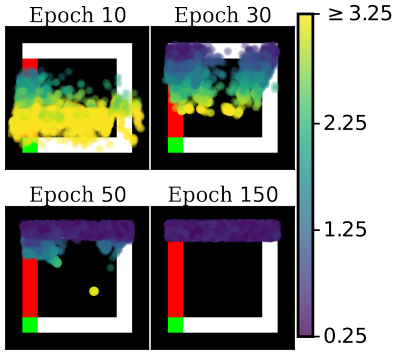


Figure 3: CURROT's progression in *safety-maze* where the context determines the position and tolerance (brighter for higher) of the goal. Contexts from ϱ_k with $k \in \{10, 30, 50, 150\}$.

5 Safe Curriculum Generation

We develop **Safe Curriculum Generation (SCG)**, an automated curriculum generation method that minimizes constraint violation regret and sample-efficiently learns a policy optimizing the CCRL objective (1). Algorithm 1 is a pseudocode for SCG. At curriculum iteration k, SCG samples contexts $\{\mathbf{x}_i\}_{i=0}^{M}$ from context distribution ϱ_{k-1} (Line 5), and collects trajectories $\mathcal{D}_k = \{\boldsymbol{\tau}_{\mathbf{x}_i}\}_{i=1}^{M}$, where each transition includes the received cost (Line 6). Then, a constrained RL algorithm updates policy π_{k-1} (Line 7). Next, based on $\mathcal{D}_k$, the UpdateSuccessfulContexts() function determines *successful* contexts ($\mathcal{B}_+$) according to SCG's three phases (Line 8): 1) prioritizing safety, 2) prioritizing performance, and 3) safely approaching the target context distribution. Finally, SCG updates V_r^{π} and V_c^{π} based on $\mathcal{B}_+$ and $\mathcal{B}_-$ (Line 9) and generates the next context distribution ϱ_k (Line 10) via

$$\Phi_{\text{SCG}}^{\varphi}(\pi_k, \varrho_+, \tilde{D}, \zeta) = \arg\min_{\varrho} \mathcal{W}_2(\varrho, \varphi) \quad \text{s.t.} \quad \begin{aligned} &\varrho(\mathbf{x}) > 0 \Rightarrow V_r^{\pi_k}(\mathbf{x}) \geq \zeta, \forall \mathbf{x} \in \mathcal{X}, \\ &\varrho(\mathbf{x}) > 0 \Rightarrow V_c^{\pi_k}(\mathbf{x}) \leq \tilde{D}, \forall \mathbf{x} \in \mathcal{X}, \\ &\mathcal{W}_2(\varrho, \varrho_+) \leq \epsilon, \end{aligned} \tag{4}$$

where, in contrast to CURROT, SCG imposes a constraint on V_c^{π} to ensure that the support of the next distribution will be over low-cost contexts to minimize safety violations. Note that constrained RL algorithm Λ utilizes safety threshold D for the constraint on the expected cumulative cost (1), SCG uses cost threshold $\tilde{D}$ for a constraint on individual contexts under the support of ϱ_k. For the remainder of this section, we describe SCG's three phases, while Appendix C provides more details.

1) Prioritizing safety. Early on in training, an RL agent likely collects high costs or low rewards during exploration. In a safety-critical setting, this period can rapidly increase constraint violation regret until the agent discovers how to behave safely. Therefore, SCG initially proposes *easy* contexts where the agent can behave safely without much exploration. To achieve that, UpdateSuccessfulContexts() labels safe contexts as *successful*. A context $\mathbf{x}$ is *safe* if the discounted cumulative cost $G_c(\boldsymbol{\tau}_{\mathbf{x}})$ is less than the median cost C_{med} of $\mathcal{B}_+$. SCG updates $\mathcal{B}_+$ with safe contexts only, and generates ϱ_+, a Gaussian mixture model (GMM), around contexts in $\mathcal{B}_+$.

2) Prioritizing performance. Once C_{med} is less than the cost threshold $\tilde{D}$, SCG focuses on performant contexts. A *performant* context has discounted cumulative reward $G_r(\boldsymbol{\tau}_{\mathbf{x}})$ greater than the median reward R_{med} of $\mathcal{B}_+$. In the first two phases, $\mathcal{B}_+$ and $\mathcal{B}_-$ get updated cyclically. SCG generates ϱ_+ to be a GMM centered in contexts from $\mathcal{B}_+$, as in the previous phase.

Algorithm 1 **S**afe **C**urriculum **G**eneration (SCG)

Input: Target and initial context distributions φ and ϱ_0
Parameters: Safety threshold D, cost threshold $\tilde{D}$, performance threshold ζ, Wasserstein distance bound ϵ, number of curriculum iterations K, number of rollouts per iteration M, buffer size N
Output: Policy π

Initialize policy π_0
$\mathcal{B}_-, \mathcal{B}_+ \leftarrow (), ()$ ▷ *initialize buffers of size N*
IsSafe, IsPerf $\leftarrow$ False, False ▷ *to search safe and performant contexts*
for $k = 1$ **to** K **do**
  $\mathbf{x}_i \sim \varrho_{k-1}$, $i \in [M]$ ▷ *sample contexts*
  $\mathcal{D}_k = \{\boldsymbol{\tau}_{\mathbf{x}_i} = (\mathbf{s}_{i,t}, \mathbf{x}_i, \mathbf{a}_{i,t}, \mathbf{s}_{i,t+1}, r_{i,t}, c_{i,t})_{t=0}^{T}\}_{i=1}^{M}$ ▷ *collect rollouts via policy π_{k-1}*
  $\pi_k \leftarrow \Lambda(\mathcal{D}_k, \pi_{k-1}, D)$ ▷ *policy update via a constrained RL algorithm Λ*
  $\mathcal{B}_+, \mathcal{B}_-, \varrho_+,$ IsSafe, IsPerf $\leftarrow$ UpdateSuccessfulContexts$(\mathcal{B}_+, \mathcal{B}_-,$ IsSafe, IsPerf, $\mathcal{D}_k)$
  Update value functions $V_r^{\pi_k}$ and $V_c^{\pi_k}$ with $\mathcal{B}_+$ and $\mathcal{B}_-$
  $\varrho_k \leftarrow \Phi_{\text{SCG}}^{\varphi}(\pi_k, \varrho_+, \tilde{D}, \zeta)$ ▷ *new context distribution (4)*
end for
return π

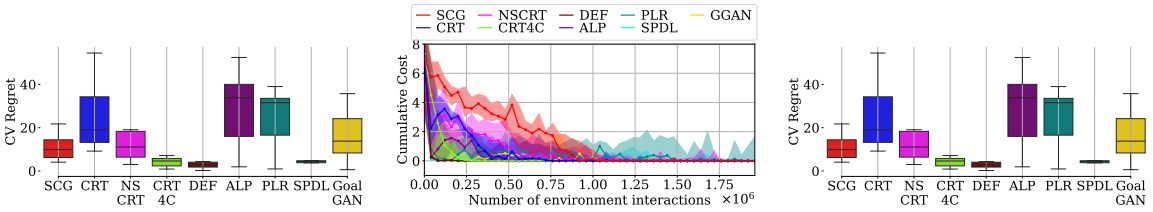
3) Safely approaching the target context distribution. When R_{med} exceeds ζ, SCG moves to the final phase. Here, UpdateSuccessfulContexts() labels a context $\mathbf{x}$ as successful if the policy π_{k-1} collects discounted cumulative reward greater than or equal to ζ and a discounted cumulative cost less than or equal to $\tilde{D}$. Similar to CURROT, to update a full success buffer, SCG generates a particle-based context distribution $\varrho_+(\mathbf{x}) = \frac{1}{|\mathcal{B}_+|}\sum_{j=1}^{|\mathcal{B}_+|}\delta_{\mathcal{B}_+}(\mathbf{x})$, where $\delta_{\mathcal{B}_+}$ is a Dirac delta at contexts in $\mathcal{B}_+$. Next, it replaces contexts in $\mathcal{B}_+$ with new ones from a distribution that minimizes the Wasserstein distance $\mathcal{W}_2(\varrho_+, \varphi)$. In contrast, $\mathcal{B}_-$ gets updated cyclically.

6 Empirical Results

We set up experiments in constrained RL domains to investigate the benefits of SCG. Qualitatively, we demonstrate the evolution of curricula generated by SCG to evidence that SCG proposes safe and performant contexts. Quantitatively, we consider **three metrics**: 1) constraint violation regret $\text{Reg}^{tr}(L, \{\varrho_l\}_{l=1}^{L}, D)$ in (2), 2) expected discounted cumulative cost $\mathbb{E}_{\varphi}[V_c^{\pi}(\mathbf{x})]$, and expected success or discounted cumulative reward $\mathbb{E}_{\varphi}[V_r^{\pi}(\mathbf{x})]$, both with respect to target context distribution φ. We compare SCG with **five state-of-the-art curriculum learning methods**: CURROT (Klink et al., 2022), SPDL (Klink et al., 2021), PLR (Jiang et al., 2021b), GoalGAN (Florensa et al., 2018), and ALP-GMM (Portelas et al., 2020). Appendix B provides details about these algorithms. Finally, we include **three baseline methods**: Default, CURROT4Cost, and NaiveSafeCURROT. Default draws contexts from the target context distribution without generating a curriculum. CURROT4Cost is a version of CURROT that replaces the performance constraint with a cost constraint $V_c^{\pi}(\mathbf{x}) \leq \tilde{D}$. NaiveSafeCURROT is a naively safe CURROT that penalizes the reward with the cost to have an augmented performance constraint: $V_r^{\pi}(\mathbf{x}) - V_c^{\pi}(\mathbf{x}) \geq \zeta$. The baselines serve as ablation studies to understand whether generating a curriculum boosts learning performance and/or improves safety during training and whether focusing only on the cost or a naive penalization using the cost is sufficient, respectively. We utilize PPO-Lagrangian, a constrained RL algorithm proposed by Achiam & Amodei (2019).

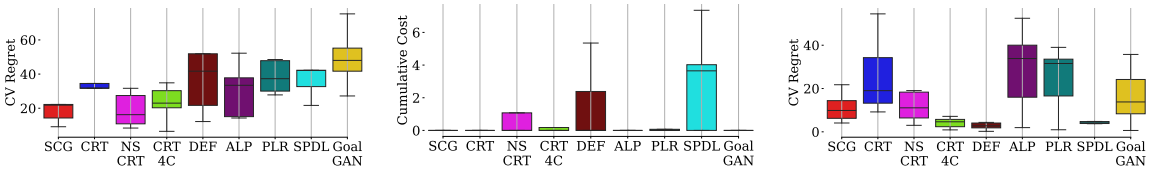
6.1 Safety-Maze

Safety-maze is a constrained version of the maze environment proposed by Klink et al. (2022). The objective of an agent and the context are the same in safety-goal (Section 3). The context space $\mathcal{X} = [-9, 9] \times [-9, 9] \times [0.25, 5.0]$ is over positions and tolerances of the goal, respectively. The target distribution is uniform over the top white row (see Fig. 5) where the agent can move, unlike the black areas.



(a) Final CV regret (b) Cost progression (c) Final success

Figure 4: Safety-maze results from runs in 10 seeds: a) Constraint violation regret at the final curriculum iteration. Box plots show the minimum, the first quartile, the median, the third quartile, and the maximum, from bottom to top. b) Progression of expected discounted cumulative cost in contexts drawn from the target context distribution. The bold lines are the median and the shaded regions cover the first and third quartiles. c) Expected success of the final policies in contexts from the target context distribution. Due to limited space, we use CRT for CURROT, NSCRT for NaiveSafeCURROT, DEF for Default, ALP for ALP-GMM, and GGAN for GoalGAN.



(a) Final CV regret (b) Final cost (c) Final success

Figure 6: Safety-goal results from runs in 5 seeds: a) Constraint violation regret at the final curriculum iteration. b) Expected discounted cumulative cost of the final policies in target contexts. c) Expected discounted cumulative reward of the final policies in target contexts.

Curriculum generation. Figure 5 shows the progression of SCG's curricula. Early on in the training, SCG prioritizes contexts with high tolerance (lighter color) and positions over the bottom white row, as they are *easy*. If the vertical goal coordinates are above the vertical half, then the agent cannot reach the goal without going above the bottom white row, even when the context has the highest tolerance. In this case, moving over the left column leads to costs. Therefore, SCG moves contexts over the right white column with gradually decreasing tolerances (see Epoch 50). Once the agent learns how to reach the top row from the right, SCG moves its contexts towards the target context distribution. In contrast, CURROT cannot distinguish whether the agent behaves safely, and thus chooses to move contexts from the left causing high constraint violation regret (see Fig. 3).

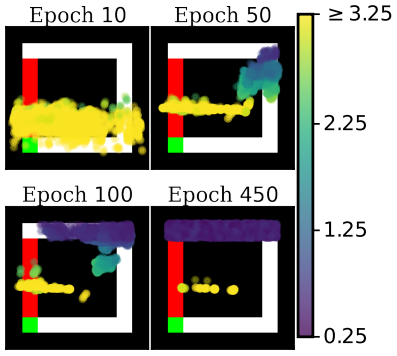


Figure 5: SCG's curricula in safety-maze: Contexts from context distributions ϱ_k with $k \in \{10, 50, 100, 450\}$.

Safety during training and final performance. Figure 4a shows the constraint violation regret at the end of the training. Although SCG doesn't have the lowest constraint violation regret, every other approach, other than CURROT, fails to achieve an optimal behavior as the success rates indicate in Figure 4c. However, CURROT has the third highest constraint violation regret, as it suffers from misaligned objectives with constrained RL (see Section 4.2). CURROT4Cost and NaiveSafeCURROT yield similar or lower constraint violation regret compared to SCG, but they have highly varying success rates. As Default fails to consistently learn an optimal policy, we can argue that a curriculum boosts performance in safety-maze.

6.2 Safety-Goal

The objective of an agent and the effect of a context in safety-goal are similar to safety-maze. However, our experiments in safety-goal aim to demonstrate that the benefits of SCG are not due to simple dynamics or low dimensional states. The context space $\mathcal{X} = [-1.5, 1.5] \times [-1.5, 1.5] \times [0.25, 0.75]$ is goal positions and tolerances, respectively. See Section 3.1 for more details.

Curriculum generation. Similar to safety-maze, SCG prioritizes contexts on the right side of the environment, as they allow the agent to learn how to reach the target contexts by avoiding the hazards (see Fig. 7) As SCG's curricula approach the target context distribution, the tolerance of goal on the hazards or the columns drop, because the agent is already able to reach these goals by stopping right next to the hazards or the columns without collecting costs.

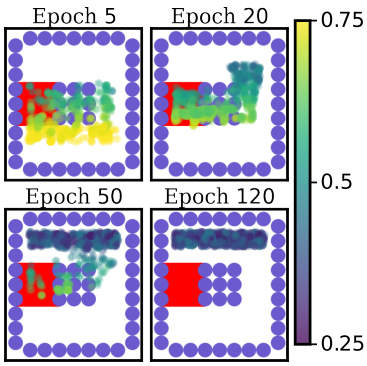


Figure 7: SCG's curricula in safety-goal: Contexts from context distributions ϱ_k with $k \in \{5, 20, 50, 120\}$.

Safety during training and final performance. Figure 6 evidences that, in safety-goal, SCG achieves the lowest constraint violation regret at the end of the training, as well as the lowest expected discounted cumulative cost and the highest expected success in target contexts. NaiveSafeCURROT and CURROT4Cost achieve similar levels of constraint violation regret, but not as robustly, and also they yield higher costs and lower success rates in target contexts. Default fails to learn safe and performant policies, similar to the other state-of-the-art methods. Our empirical results in safety-goal support our previous observation that aligning curriculum learning with constrained RL via SCG boosts performance and safety during training.

7 Conclusion

In this work, we study safe automated curriculum generation in multi-task cost-constrained settings with distributions over target tasks. We propose a safe curriculum generation approach (SCG) developed for constrained RL to minimize constraint violation regret and accelerate learning. SCG initially prioritizes tasks with low costs over high-reward ones, to ensure that the agent learns a policy that satisfies the cost constraint. Next, SCG proposes tasks where the agent can collect high rewards. Finally, SCG takes safety and performance into account together. Our empirical evaluation evidence that state-of-the-art curriculum learning approaches fail to learn an optimal behavior safely and stably as they suffer from misaligned objectives with constrained RL. In contrast, SCG obtains optimal behavior with the lowest constraint violation regret in all constrained RL domains we study. SCG achieves this in domains with low or high dimensional state spaces, or in settings where a safe curriculum learning approach does not have a trivial advantage.

Limitations. SCG aims to minimize constraint violation regret while preserving the benefits of curriculum learning in boosting learning speed. However, SCG does not provide any guarantees for the constraint violation regret achieved at the end of the training. This is primarily because we do not employ a constrained RL algorithm that guarantees safety during training, as in Simão et al. (2021). In addition, we do not assume to access the dynamics or to intervene with the interactions between an environment and an RL agent, as in Eysenbach et al. (2018); Turchetta et al. (2020).

Future work. Our current plan is to extend SCG by providing safety guarantees without limiting assumptions. As the state-of-the-art curriculum learning methods do not assume to interfere with the training, except by setting environment configurations at the beginning of each episode, we aim to follow the same direction. A possible way is to exploit the properties of a context space, as proposed by Simão et al. (2021) in a non-curriculum study so that both the RL agent and the curriculum generator can achieve regret guarantees.

References

Joshua Achiam and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning, 2019. URL https://cdn.openai.com/safexp-short.pdf.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. PMLR, 2017.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *AAAI*, volume 32, 2018.

Eitan Altman. *Constrained Markov decision processes*. Routledge, 1999.

Adrien Baranes and Pierre-Yves Oudeyer. Intrinsically motivated goal exploration for active motor learning in robots: A case study. In *IROS*, pp. 1766–1773, 2010.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *NeurIPS*, 30, 2017.

Jiayu Chen, Yuanxin Zhang, Yuanfan Xu, Huimin Ma, Huazhong Yang, Jiaming Song, Yu Wang, and Yi Wu. Variational Automatic Curriculum Learning for Sparse-Reward Cooperative Multi-Agent Problems. In *NeurIPS*, pp. 9681–9693, 2021.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *NeurIPS*, 31, 2018.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *NeurIPS*, 33:13049–13061, 2020.

Yonathan Efroni, Shie Mannor, and Matteo Pirotta. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*, 2020.

Theresa Eimer, André Biedenkapp, Frank Hutter, and Marius Lindauer. Self-paced context evaluation for contextual reinforcement learning. In *ICML*, pp. 2948–2958, 2021.

Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In *ICLR*, 2018.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse Curriculum Generation for Reinforcement Learning. In *CoRL*, pp. 482–495. PMLR, 2017.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic Goal Generation for Reinforcement Learning Agents. In *ICML*, pp. 1514–1523. PMLR, 2018.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.

Yannick Hogewind, Thiago D Simão, Tal Kachman, and Nils Jansen. Safe reinforcement learning from pixels using a stochastic latent representation. In *ICLR*, 2022.

Peide Huang, Mengdi Xu, Jiacheng Zhu, Laixi Shi, Fei Fang, and Ding Zhao. Curriculum reinforcement learning using optimal transport via gradual domain adaptation. *NeurIPS*, 35:10656–10670, 2022.

Nils Jansen, Bettina Könighofer, Sebastian Junges, Alex Serban, and Roderick Bloem. Safe reinforcement learning using probabilistic shields. In *CONCUR*, 2020.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Juntao Dai, and Yaodong Yang. Safety-gymnasium: A unified safe reinforcement learning benchmark. *arXiv preprint arXiv:2310.12567*, 2023a.

Jiaming Ji, Jiayi Zhou, Borong Zhang, Juntao Dai, Xuehai Pan, Ruiyang Sun, Weidong Huang, Yiran Geng, Mickel Liu, and Yaodong Yang. Omnisafe: An infrastructure for accelerating safe reinforcement learning research. *arXiv preprint arXiv:2305.09304*, 2023b.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. Self-paced curriculum learning. In *AAAI*, pp. 2694–2700. AAAI Press, 2015.

Minqi Jiang, Michael Dennis, Jack Parker-Holder, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Replay-guided adversarial environment design. *NeurIPS*, pp. 1884–1897, 2021a.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *ICML*, pp. 4940–4950. PMLR, 2021b.

Sebastian Junges, Nils Jansen, Christian Dehnert, Ufuk Topcu, and Joost-Pieter Katoen. Safety-constrained reinforcement learning for MDPs. In *TACAS*, pp. 130–146. Springer, 2016.

Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *ICRA*, pp. 8248–8254. IEEE, 2019.

Pascal Klink, Hany Abdulsamad, Boris Belousov, and Jan Peters. Self-paced contextual reinforcement learning. In *CoRL*, pp. 513–529. PMLR, 2020a.

Pascal Klink, Carlo D' Eramo, Jan R Peters, and Joni Pajarinen. Self-paced deep reinforcement learning. In *NeurIPS*, pp. 9216–9227. Curran Associates, Inc., 2020b.

Pascal Klink, Hany Abdulsamad, Boris Belousov, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. A probabilistic interpretation of self-paced learning with applications to reinforcement learning. *JMLR*, 22:182:1–182:52, 2021.

Pascal Klink, Haoyi Yang, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. Curriculum reinforcement learning via constrained optimal transport. In *ICML*, pp. 11341–11358. PMLR, 2022.

Cevahir Koprulu and Ufuk Topcu. Reward-machine-guided, self-paced reinforcement learning. In *UAI*, volume 216, pp. 1121–1131. PMLR, 2023.

Cevahir Koprulu, Thiago D. Simão, Nils Jansen, and Ufuk Topcu. Risk-aware curriculum generation for heavy-tailed task distributions. In *UAI*, volume 216, pp. 1132–1142. PMLR, 2023.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NIPS*, pp. 1189–1197. Curran Associates, Inc., 2010.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *JMLR*, 21: 181:1–181:50, 2020.

Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In *CoRL*, pp. 835–853. PMLR, 2020.

Sebastien Racaniere, Andrew K Lampinen, Adam Santoro, David P Reichert, Vlad Firoiu, and Timothy P Lillicrap. Automated curricula through setter-solver interactions. In *ICLR*, 2020.

Zhipeng Ren, Daoyi Dong, Huaxiong Li, and Chunlin Chen. Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning. *IEEE TNNLS*, pp. 2216–2226, 2018.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *ICLR*, 2016.

Thiago D. Simão, Nils Jansen, and Matthijs T. J. Spaan. Alwayssafe: Reinforcement learning without safety constraint violations during training. In *AAMAS*, pp. 1226–1235, Richland, SC, 2021. IFAAMAS.

Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with gaussian processes. In *ICML*, pp. 997–1005. PMLR, 2015.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration for interactive machine learning. *NeurIPS*, 32, 2019.

Matteo Turchetta, Andrey Kolobov, Shital Shah, Andreas Krause, and Alekh Agarwal. Safe reinforcement learning via curriculum induction. In *NeurIPS*, pp. 12151–12162, 2020.

Georgios Tzannetos, Bárbara Gomes Ribeiro, Parameswaran Kamalaruban, and Adish Singla. Proximal curriculum for reinforcement learning agents. *TMLR*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=8WUyeeMxMH.

Akifumi Wachi and Yanan Sui. Safe reinforcement learning in constrained Markov decision processes. In *ICML*, pp. 9797–9806. PMLR, 2020.

Haozhe Wang, Chao Du, Panyan Fang, Shuo Yuan, Xuming He, Liang Wang, and Bo Zheng. Roi-constrained bidding via curriculum-guided bayesian reinforcement learning. In *SIGKDD*, pp. 4021–4031, 2022.

Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Projection-based constrained policy optimization. *arXiv preprint arXiv:2010.03152*, 2020.

Yunzhi Zhang, Pieter Abbeel, and Lerrel Pinto. Automatic curriculum learning through value disagreement. In *NeurIPS*, pp. 7648–7659. Curran Associates, Inc., 2020.

A Extension of Related Work

Curriculum learning for RL. Automated curriculum generation in RL aims to accelerate convergence to optimal policies by changing the environment configuration according to agent performance. A common curriculum scheme is to create sequences of distributions over such configurations. Florensa et al. (2017) propose generating distributions over initial states where early on during the training, the agent starts in the proximity of the goal state. Another line of work focuses on goal states by optimizing for value disagreement (Zhang et al., 2020), intrinsic motivation (Baranes & Oudeyer, 2010; Portelas et al., 2020), feasibility and coverage of goal states (Racaniere et al., 2020), and intermediate task difficulty (Florensa et al., 2018; Tzannetos et al., 2023). Dennis et al. (2020) proposes unsupervised level design as an alternative curriculum learning paradigm and an approach that adversarially generates environment configurations while avoiding infeasible ones. Others study generating distributions over levels, namely, environment instances, that allow the agent to have high learning potential (Jiang et al., 2021b;a). We study *self-paced RL*, a method adopted from supervised learning to order training samples in increasing complexity (Kumar et al., 2010; Jiang et al., 2015). Eimer et al. generate sequences of tasks that have a high capacity for value improvement. Ren et al. (2018) develops a self-paced mechanism that minimizes coverage penalty by generating sequences of environment interactions. Klink et al. (2020a;b; 2021; 2022); Koprulu & Topcu (2023); Koprulu et al. (2023); Huang et al. (2022) formulate the curriculum generation problem as interpolations between task distributions. Although (Chen et al., 2021) investigates a similar formulation, they do not follow the self-paced RL framework.

Constrained RL. Constrained RL studies safety-critical settings where errors during exploration may cause constraint violations (Kendall et al., 2019). Therefore, a constrained RL approach aims to achieve safe behavior during and after training (Simão et al., 2021). Constrained RL approaches that guarantee zero safety violation during training propose using Gaussian processes as transition models (Sui et al., 2015; Berkenkamp et al., 2017; Turchetta et al., 2019; Wachi & Sui, 2020), Lyapunov functions for ensuring global constraints (Chow et al., 2018), or formal methods (Junges et al., 2016; Alshiekh et al., 2018; Jansen et al., 2020). To address environments with high dimensional state and action spaces, Achiam et al. (2017); Tessler et al. (2018); Yang et al. (2020); Hogewind et al. (2022) develop safe policy search algorithms with soft guarantees of not violating the constraints, whereas Achiam & Amodei (2019) combine a Lagrangian approach with popular RL algorithms.

B Automated Curriculum Generation Algorithms

In this section, we provide short descriptions of the state-of-the-art curriculum learning methods evaluated in the experiments.

- GoalGAN (Florensa et al., 2018): *Goal Generative Adversarial Network* is a curriculum learning approach developed for goal-conditioned RL. GoalGAN trains a goal discriminator to classify goals that are at the intermediate difficulty for the policy of the RL agent, and a goal generator to generate goals at that difficulty to boost learning performance.
- ALP-GMM (Portelas et al., 2020): *Absolute Learning Progress with Gaussian Mixture Models* uses the absolute learning progress of a task to measure whether a task would improve the learning process of an RL agent. ALP-GMM learns a Gaussian mixture model over the absolute learning progress where a multi-armed bandit samples a Gaussian as an arm based on its utility, which is the absolute learning progress. The Gaussian distribution that the arm corresponds to draws a task, namely, the context in our setting.
- SPDL (Klink et al., 2021): *Self-paced Deep Reinforcement Learning* formulates the automated curriculum generation problem similarly to CURROT, except that SPDL generates context distributions that minimize the KL divergence to the target context distribution. The constraints in the optimization problem solved in SPDL are on minimum expected discounted cumulative reward and maximum KL divergence to the previous context distribu-

tion. SPDL does not include an initial search procedure and generates context distributions as Gaussian distributions.

- PLR (Jiang et al., 2021b): *Prioritized Level Replay* is a curriculum learning method developed for procedural context generation environments, where a *level* corresponds to a task, i.e., an environment instance. PLR prioritizes levels that have a high average magnitude of generalized advantage estimate (Schulman et al., 2016), namely, the discounted sum of temporal-difference errors.
- CURROT (Klink et al., 2022): We propose SCG based on *Curriculum RL via Constrained Optimal Transport*, which we describe and discuss in Sections 4.1 and 4.2.

C Details of SCG

To support Section 5, here we provide a closer look into how the UpdateSuccessfulContexts() function in SCG works (See Algorithm 2 for a pseudocode). First, we note that SCG does not sample contexts from ϱ_+ but uses it as a source distribution to approach the target context distribution φ (4). Inspired by CURROT, SCG generates ϱ_+ in the first two phases as a GMM to allow for exploration in the context space. In the final phase, SCG models ϱ_+ as a particle-based distribution since exploration is not as critical and $\mathcal{B}_+$ well represents where the agent is safe and performant.

1) Prioritizing safety. Initially, SCG sets FoundSafeXs and FoundPerfXs to false to enable the UpdateSuccessfulContexts() function to search for safe contexts first. Lines 2-3 indicate that a successful context in this phase yields a discounted cumulative cost less than or equal to the median cost C_{med} in success buffer $\mathcal{B}_+$. Cyclically, $\mathcal{B}_+$ gets updated with the successful contexts in trajectory set $\mathcal{D}_k$, as $\mathcal{B}_-$ gets updated with the rest of the contexts. Then, UpdateSuccessfulContexts() generates ϱ_+ as a Gaussian mixture model using $\mathcal{B}_+$ (Line 4)

$$\Xi^{\text{INIT}}_{\text{SAFE}}(\mathcal{B}_+) = \sum_{\mathbf{x}_i \in \mathcal{B}_+} \omega_i^c \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \sigma^2_{\text{SAFE},i}\mathbf{I}), \tag{5}$$
$$\text{where} \quad \omega_i^c \propto \max\{0, C_{\text{med}} - G_c(\boldsymbol{\tau}_{\mathbf{x}_i})\},$$
$$\sigma_{\text{SAFE},i} = \max\left\{\sigma_{\min}, 2\frac{G_c(\boldsymbol{\tau}_{\mathbf{x}_i}) - \tilde{D}}{C_{\max} - \tilde{D}}\right\}.$$

$C_{\max}$ is the maximum cost in all contexts until curriculum iteration k. A weight ω_i^c of this GMM is proportional to how below the discounted cumulative cost is from the median cost C_{med}. SCG searches for such safe contexts until C_{med} is less than or equal to cost threshold $\tilde{D}$ (Line 5).

2) Prioritizing performance. Once the contexts in $\mathcal{D}_k$ satisfy this safety condition, UpdateSuccessfulContexts() switches its focus to finding performant contexts. Similarly, SCG begins by updating $\mathcal{B}_+$ with contexts where the discounted cumulative reward is greater than or equal to the performance threshold ζ (Lines 7-8). Then, SCG uses $\Xi^{\text{INIT}}_{\text{PERF}}(\mathcal{B}_+)$ to generate ϱ_+, which differs from $\Xi^{\text{INIT}}_{\text{SAFE}}(\mathcal{B}_+)$ in terms of GMM weights ω^r and standard deviation $\sigma_{\text{PERF},i}$.

$$\Xi^{\text{INIT}}_{\text{PERF}}(\mathcal{B}_+) = \sum_{\mathbf{x}_i \in \mathcal{B}_+} \omega_i^r \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \sigma^2_{\text{PERF},i}\mathbf{I}), \tag{6}$$
$$\text{where} \quad \omega_i^r \propto \max\{0, G_r(\boldsymbol{\tau}_{\mathbf{x}_i}) - R_{\text{med}}\}, \text{ and}$$
$$\sigma_{\text{PERF},i} = \max\left\{\sigma_{\min}, 2\frac{\zeta - G_r(\boldsymbol{\tau}_{\mathbf{x}_i})}{\zeta - R_{\min}}\right\}.$$

Note that $R_{\min}$ is the minimum reward until curriculum iteration k. SCG prioritizes performant contexts until R_{med} is greater than or equal to ζ. During the initial search, SCG updates $\mathcal{B}_+$ and $\mathcal{B}_-$ in a cyclic fashion.

3) Safely approaching the target context distribution. Section 5 already provides information about how the last phase of SCG works. This phase operates similarly to the main phase of CURROT. For a detailed description, we refer the reader to Klink et al. (2022).

Algorithm 2 UPDATESUCCESSFULCONTEXTS()

Input: $\mathcal{B}_+, \mathcal{B}_-, \text{IsSafe}, \text{IsPerf}, \mathcal{D}_k$
Parameters: Cost threshold $\tilde{D}$, performance threshold ζ
Output: $\mathcal{B}_+, \mathcal{B}_-, \varrho_+, \mathcal{D}_k, \text{IsSafe}, \text{IsPerf}$

if not FOUNDSAFEXS **then**
  Add $\{\mathbf{x}_i | G_c(\boldsymbol{\tau}_{\mathbf{x}_i}) > C_{\text{med}})\}$ to $\mathcal{B}_-$
  Add $\{\mathbf{x}_i | G_c(\boldsymbol{\tau}_{\mathbf{x}_i}) \leq C_{\text{med}}\}$ to $\mathcal{B}_+$
  $\varrho_+ \leftarrow \Xi^{\text{INIT}}_{\text{SAFE}}(\mathcal{B}_+)$ ▷ *prioritize safety*
  FOUNDSAFEXS $\leftarrow C_{\text{med}} \leq \tilde{D}$
else if not FOUNDPERFXS **then**
  Add $\{\mathbf{x}_i | G_r(\boldsymbol{\tau}_{\mathbf{x}_i}) < R_{\text{med}}\}$ to $\mathcal{B}_-$
  Add $\{\mathbf{x}_i | G_r(\boldsymbol{\tau}_{\mathbf{x}_i}) \geq R_{\text{med}}\}$ to $\mathcal{B}_+$
  $\varrho_+ \leftarrow \Xi^{\text{INIT}}_{\text{PERF}}(\mathcal{B}_+)$ ▷ *prioritize performance*
  FOUNDPERFXS $\leftarrow R_{\text{med}} \geq \zeta$
else
  Add $\{\mathbf{x}_i | G_r(\boldsymbol{\tau}_{\mathbf{x}_i}) < R_{\text{med}}$ or $G_c(\boldsymbol{\tau}_{\mathbf{x}_i}) > C_{\text{med}}\}$ to $\mathcal{B}_-$
  $\mathcal{B}^{\text{TEMP}}_+ \leftarrow \{\mathbf{x}_i | G_r(\boldsymbol{\tau}_{\mathbf{x}_i}) \geq R_{\text{med}}$ and $G_c(\boldsymbol{\tau}_{\mathbf{x}_i}) \leq C_{\text{med}}\}$
  $\mathcal{B}_+, \varrho_+ \leftarrow \Xi^{\text{MAIN}}(\mathcal{B}^{\text{TEMP}}_+, \mathcal{B}_+, \varphi)$ ▷ *main phase*
end if
return $\mathcal{B}_+, \mathcal{B}_-, \varrho_+, \text{IsSafe}, \text{IsPerf}, \mathcal{D}_k$

Table 1: Parameters used for SCG, CURROT, NAIVESAFECURROT, and CURROT4COST.

Environment	ζ	$\tilde{D}$	ϵ_{KL}	ϵ	K	M
Safety-maze	0.6	0.25	0.25	1.25	500	40
Safety-goal	0.6	1	0.25	0.5	150	20

Table 2: Selected values for parameters of PLR, GOALGAN and ALP-GMM

Environment	ρ	β	p	δ_{noise}	$n^{\text{GG}}_{\text{rollout}}$	p_{success}	p_{rand}	$n^{\text{AG}}_{\text{rollout}}$	s_{buffer}
Safety-maze	0.45	0.15	100	0.1	200	0.2	0.2	200	500
Safety-goal	0.45	0.15	100	0.1	200	0.2	0.2	200	500

D Experimental Details

We discuss the process of hyperparameter selection for the curriculum learning approaches evaluated in this work and additional details about the constrained RL environments in the experiments.

D.1 Algorithm Hyperparameters

SCG has five main parameters: performance threshold ζ, cost threshold $\tilde{D}$, Wasserstein distance threshold ϵ, number of curriculum iterations K and number of rollouts per curriculum updates M. CURROT and NAIVESAFECURROT share the same parameters except the cost threshold $\tilde{D}$, whereas CURROT4COST shares all except the performance threshold ζ. We chose ζ to be approximately the midpoint between the minimum and maximum possible discounted cumulative reward or success rate. To select the Wasserstein distance threshold ϵ, we ran a grid search over $\{0.25, 0.5\}$ for safety-goal, and over $\{1.0, 1.25\}$ for safety-maze. For the number of rollouts per curriculum updates M, we ran grid searches over $\{20, 40\}$ for all settings. Although SPDL shares ζ, K, and M, it has a KL divergence threshold ϵ_{KL}, for which we ran a grid search over $\{0.25, 0.5\}$ in all environments. Table 1 provides all parameter values. For the initial search procedure in SCG, we set the minimum standard deviation $\sigma_{\min}$ of the Gaussian mixture model to 0.001.

As parameters to tune, PLR has the score temperature β, the staleness coefficient ρ, and the replay probability p. We ran a grid search over $(\rho, \beta, p) \in \{0.15, 0.45\} \times \{0.15, 0.45\} \times \{0.55, 0.85\}$.

GoalGAN has three parameters: the number of rollouts between curriculum updates $n^{\text{GG}}_{\text{rollout}}$, the random noise on drawn contexts δ_{noise}, and the percentage of contexts to draw from the success buffer p_{success}. We ran a grid search over $(\delta_{\text{noise}}, n^{\text{GG}}_{\text{rollout}}, p_{\text{success}}) \in \{0.05, 0.1\} \times \{100, 200\} \times \{0.1, 0.2\}$. ALP-GMM has three parameters: the buffer size s_{buffer}, the number of rollouts between curriculum updates $n^{\text{AG}}_{\text{rollout}}$, and the probability of randomly sampling contexts p_{rand}. We ran a grid search over $(p_{\text{rand}}, n^{\text{AG}}_{\text{rollout}}, s_{\text{buffer}}) \in \{0.1, 0.2\} \times \{50, 100\} \times \{500, 1000\}$. Table 2 shows the final parameter used for PLR, GoalGAN, and ALP-GMM.

D.2 Environment Descriptions

Safety-maze environment. Inspired by the maze environment in Klink et al. (2022), we design safety-maze, where the agent receives rewards from of -1 until it reaches the goal, and costs of 0.25 when it enters the hazardous area (see Figure 5). The observation of the RL agent is its coordinates on the 2D plane. The action of the agent is its displacement along the horizontal and vertical axes. We use the PPO-Lagrangian algorithm Achiam & Amodei (2019) to train a constrained RL agent. The implementation we integrate into our codebase is from OmniSafe Ji et al. (2023b). The parameters of the PPO-Lagrangian are fixed to their default values in OmniSafe, except the number of steps to update the policy is 4000 and the number of iterations to update the policy is 12.

Safety-goal environment. We create an environment with a high-dimensional state space in Safety-Gymnasium Ji et al. (2023a). We use pillars, purple columns, and hazards, blue circles, as objects with which the agent, a car, interacts in the environment (see Figure 2). The rewards and costs come from the safety-gymnasium implementation. Similar to safety-maze, we only change the parameters of the PPO-Lagrangian implementation in OmniSafe by setting the number of steps and iterations to update the policy to 10000 and 15, respectively.

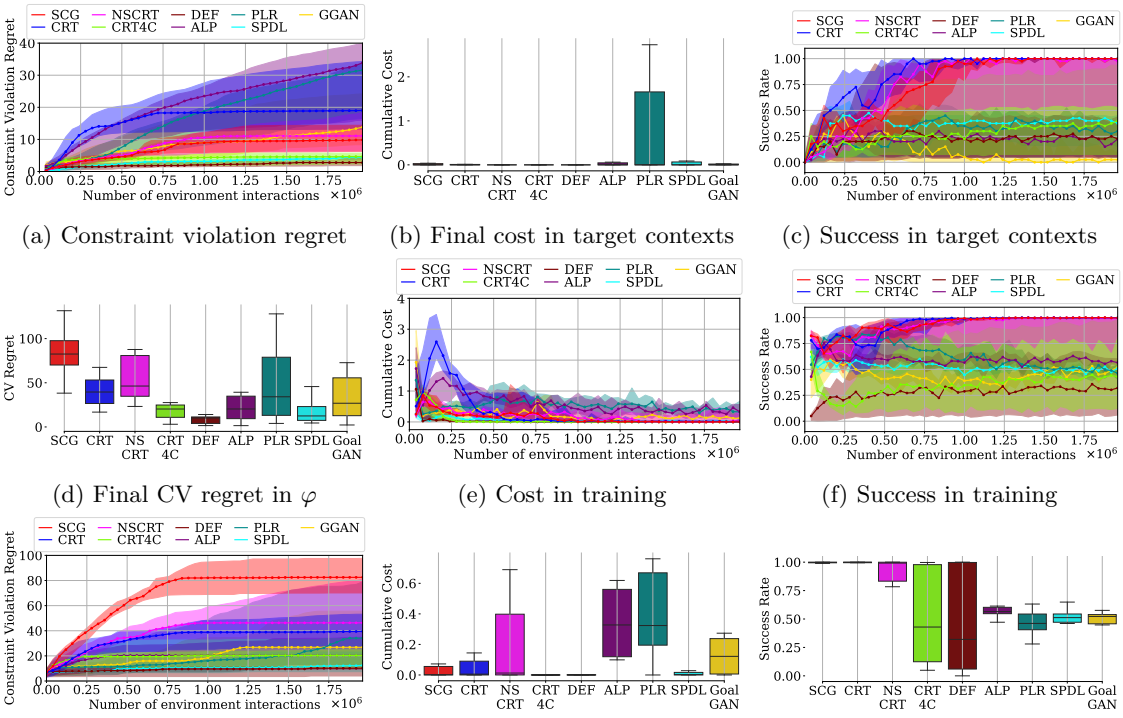
E Detailed Analysis of Results

E.1 Safety-maze

Figure 8 demonstrates additional plots that provide detailed information about safety and performance during and after training. We observe that SCG, CURROT, NaiveSafeCURROT, CURROT4Cost, and Default achieve optimal behavior in at least one run out of 10. However, SCG and CURROT consistently get optimal policies with converged constraint violation regret during training and with respect to the target context distribution. We highlight that, as SCG paces the curriculum according to how safely the agent behaves, its constraint violation regret in φ converges the last. Nevertheless, it achieves the lowest constraint violation regret in training out of all approaches that more or so reliably learn an optimal policy. ALP-GMM, PLR, SPDL, and GoalGAN achieve similar success rates throughout the training and they all fail in target contexts. However, the constraint violation regret of ALP-GMM and PLR in training increases very rapidly, with GoalGAN following behind. In contrast, the rest of the approaches have converged constraint violation regret in training.

E.2 Safety-goal

The results in Figure 9 demonstrate that SCG generates curricula that achieve the highest success rates and the lowest costs during training time and when deployment after. CURROT4Cost and Default can achieve similar success rates, but not as reliably, in target contexts. Default has its best performance out of all three constrained environments we study because safety-goal has a dense reward function, which eases learning without a curriculum. NaiveSafeCURROT and CURROT4Cost also yield as low constraint violation regret as SCG has at the final iteration of the training. However, NaiveSafeCURROT is less stable at learning policies that accomplish the task both during and after training. CURROT, ALP-GMM, SPDL, and GoalGAN follow NaiveSafeCURROT in terms of success in target contexts. An important interpretation to make in all three settings is that, in contrast to learning directly in target contexts, a curriculum learning



(a) Constraint violation regret (b) Final cost in target contexts (c) Success in target contexts

(d) Final CV regret in φ (e) Cost in training (f) Success in training

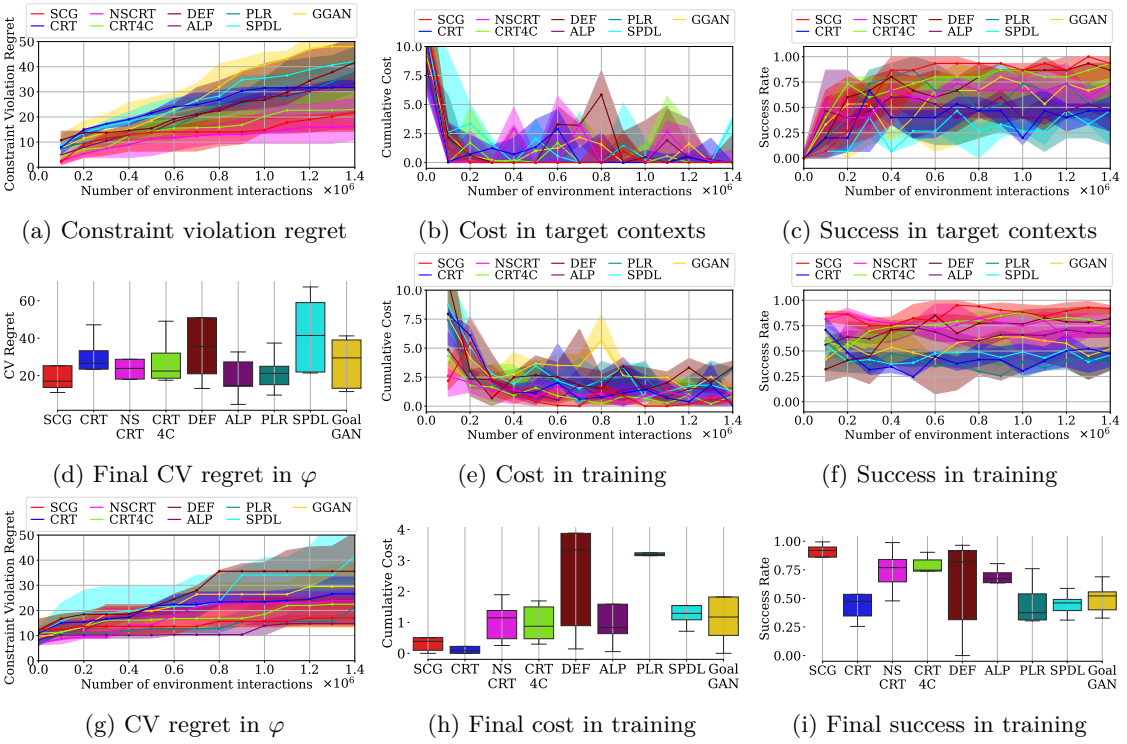
(g) CV regret in φ (h) Final cost in training (i) Final success in training

Figure 8: Safety-maze results from runs in 10 seeds: a) Evolution of constraint violation regret during. b) Expected discounted cumulative cost of the final policies in target contexts. c) Progression of expected success rate in contexts drawn from the target context distribution. d) Constraint violation regret with respect to the target context distribution at the final curriculum iteration. e) Progression of expected discounted cumulative cost in contexts sampled during training. f) Progression of expected success rate in contexts sampled during training. g) Evolution of constraint violation regret with respect to the target context distribution. h) Expected discounted cumulative cost of the final policies in contexts sampled during training. i) Expected success rate of the final policies in contexts sampled during training.

approach can cause an agent to behave unsafely if the approach does not consider the constrained nature of the task.

F Curriculum Progression Results

Figures 10, 11, 12, 13, 14, 15, and 16 demonstrate the progression of curricula generated by CURROT, NAIVESAFECURROT, CURROT4COST, SPDL, PLR. ALP-GMM, and GOALGAN, respectively, in safety-maze, safety-door, and safety-goal environments. Figure 17 demonstrates the contexts drawn from the target context distribution during training runs of DEFAULT. NAIVESAFECURROT has similar curricula to SCG, as it considers reward and cost simultaneously but through a penalized reward signal, which takes away the flexibility that SCG provides in prioritizing safe or performant contexts separately and sometimes together. CURROT4COST takes cost into account, only, but fails to recognize that goals on the hazards in safety-maze and safety-goal can lead to high constraint violation regret. Although SPDL and ALP-GMM have Gaussian and Gaussian mixture models for context distributions, only SPDL converges to the target context distribution in safety-door, as ALP-GMM assumes that the target context distribution is uniform over the context space, which is a limitation that PLR and GOALGAN suffer from, as well.

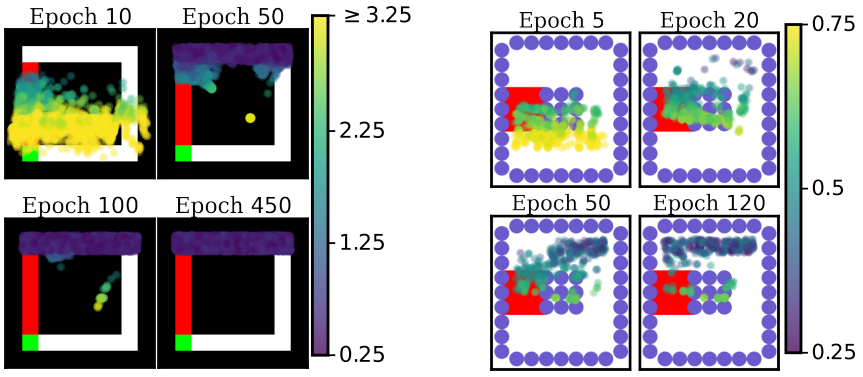


(a) Constraint violation regret (b) Cost in target contexts (c) Success in target contexts

(d) Final CV regret in φ (e) Cost in training (f) Success in training

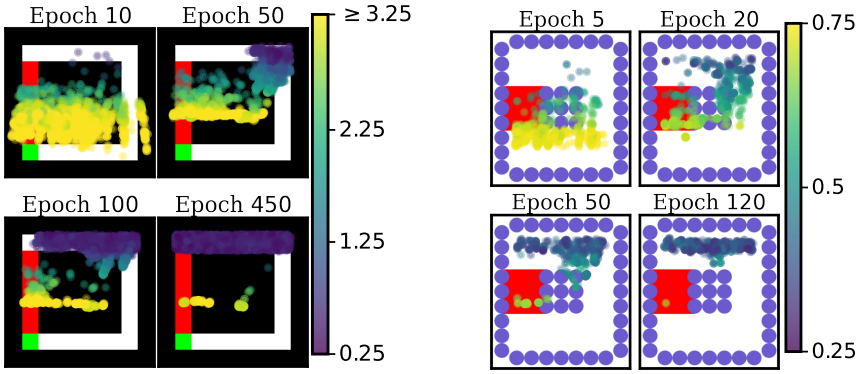
(g) CV regret in φ (h) Final cost in training (i) Final success in training

Figure 9: Safety-goal results from runs in 10 seeds: a) Evolution of constraint violation regret during. b) Progression of expected discounted cumulative cost in contexts drawn from the target context distribution. c) Progression of expected success rate in contexts drawn from the target context distribution. d) Constraint violation regret with respect to the target context distribution at the final curriculum iteration. e) Progression of expected discounted cumulative cost in contexts sampled during training. f) Progression of expected success rate in contexts sampled during training. g) Evolution of constraint violation regret with respect to the target context distribution. h) Expected discounted cumulative cost of the final policies in contexts sampled during training. i) Expected success rate of the final policies in contexts sampled during training.



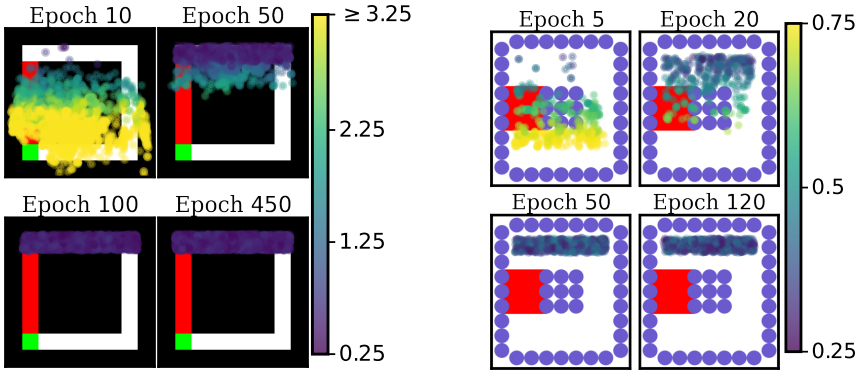
(a) Safety-maze (b) Safety-goal

Figure 10: Curricula generated by CURROT in safety-maze and safety-goal.



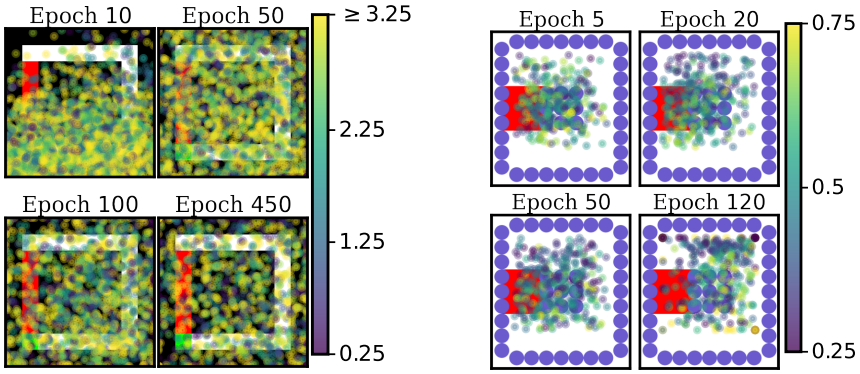
(a) Safety-maze (b) Safety-goal

Figure 11: Curricula generated by NaiveSafeCURROT in safety-maze and safety-goal.



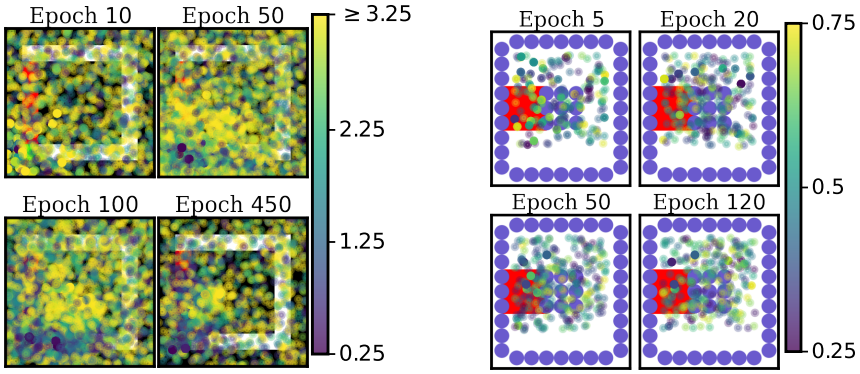
(a) Safety-maze (b) Safety-goal

Figure 12: Curricula generated by CURROT4Cost in safety-maze and safety-goal.



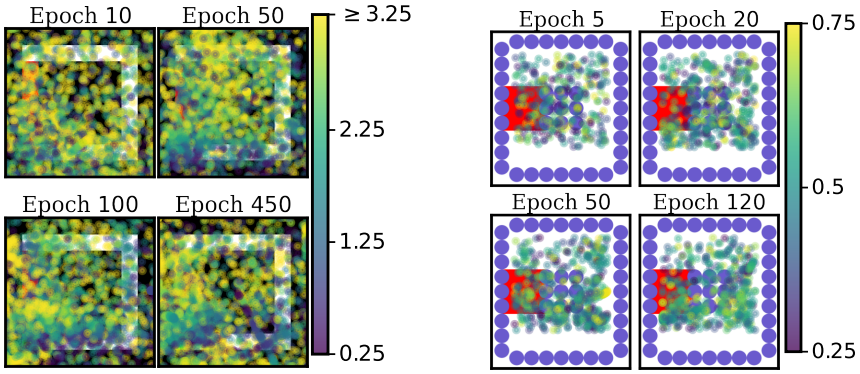
(a) Safety-maze (b) Safety-goal

Figure 13: Curricula generated by SPDL in safety-maze and safety-goal.



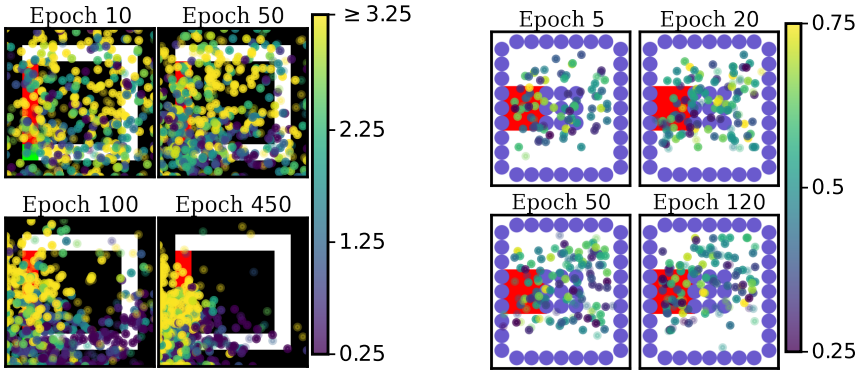
(a) Safety-maze (b) Safety-goal

Figure 14: Curricula generated by PLR in safety-maze and safety-goal.



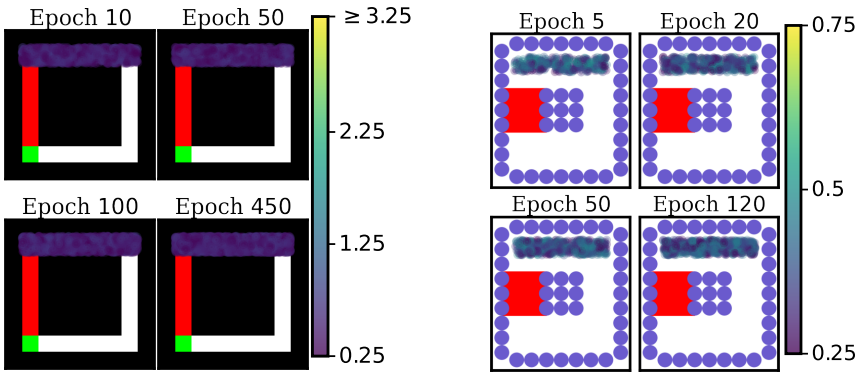
(a) Safety-maze (b) Safety-goal

Figure 15: Curricula generated by ALP-GMM in safety-maze and safety-goal.



(a) Safety-maze (b) Safety-goal

Figure 16: Curricula generated by GoalGAN in safety-maze and safety-goal.



(a) Safety-maze (b) Safety-goal

Figure 17: Curricula generated by DEFAULT in safety-maze and safety-goal.