Risk-Aware Curriculum Generation for Heavy-Tailed Task Distributions

Cevahir Koprulu [1] **Thiago D. Simão**[2] **Nils Jansen**[2] **Ufuk Topcu**[1]

[1]University of Texas at Austin
[2]Radboud University, Nijmegen

Abstract

Automated curriculum generation for reinforcement learning (RL) aims to speed up learning by designing a sequence of tasks of increasing difficulty. Such tasks are usually drawn from probability distributions with exponentially bounded tails, such as uniform or Gaussian distributions. However, existing approaches overlook *heavy-tailed* distributions. Under such distributions, current methods may fail to learn optimal policies in *rare* and *risky* tasks, which fall under the tails and yield the lowest returns, respectively. We address this challenge by proposing a risk-aware curriculum generation algorithm that simultaneously creates two curricula: 1) a *primary curriculum* that aims to maximize the expected discounted return with respect to a distribution over target tasks, and 2) an *auxiliary curriculum* that identifies and over-samples rare and risky tasks observed in the primary curriculum. Our empirical results evidence that the proposed algorithm achieves significantly higher returns in frequent as well as rare tasks compared to the state-of-the-art methods.

1 INTRODUCTION

The design of task sequences, i.e., curricula, improves the performance of reinforcement learning (RL) agents and speeds up convergence in complex tasks [31]. A curriculum typically begins with easy tasks and gradually increases difficulty toward target tasks. A common approach is to tailor the curricula using human input to identify easy and hard tasks [2, 31]. Recent studies facilitate this process by automating the curriculum generation [13, 33]. In particular, *self-paced* RL uses a *context* to parameterize the dynamics and rewards of the environment, which implicitly defines a task [23, 24, 25, 26]. These methods assume the contexts are drawn from distributions known a priori. Considering problems described by a *target context distribution*, self-paced RL automatically generates a curriculum, represented by a sequence of context distributions, to speed up learning.

In general, curriculum generation methods overlook *heavy-tailed* distributions and focus on target context distributions with exponentially bounded tails, e.g., a normal or uniform distribution. However, heavy-tailed distributions commonly appear in the real world: words in natural language [48] and relationships in social networks [1] follow a power law distribution, while Cauchy distribution appear in risk analysis [32] and rainfall models [29]. Rare events are more likely to occur under heavy-tailed distributions, as the area under the extreme regions is larger than the area under the tails of an exponentially-bounded probability distribution [43]. A possible explanation for such disregard is that simulated RL environments typically have uniform variations [5, 10, 7], and, even when a target distribution is non-uniform, it does not reflect the heavy-tailed nature of the real world [42].

Existing RL algorithms may underperform in a context drawn from the tails of a heavy-tailed distribution [45, 8]. This occurs because an off-the-shelf algorithm would be *underexposed* to rare contexts, i.e., they do not encounter such contexts sufficiently to learn a good policy for them. Curriculum generation methods face the same problem since they do not explicitly address rare contexts, either. These approaches may yield policies that are sub-optimal in rare contexts drawn from exponentially-bounded distributions, as rare contexts have a low impact on the average performance. However, in heavy-tailed distributions, these rare contexts together are more frequent and exacerbate the performance loss. Furthermore, we observe that rare contexts correlate with *risky* contexts, where the agent's return is among the lowest (see Section 5.1). As a result, curriculum learning methods fail to be robust in rare and risky contexts.

We address the challenges faced under heavy-tailed task distributions by developing a *risk-aware* curriculum generation algorithm (RACGEN). To improve the policy in the

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

tails, RACGEN simultaneously creates two curricula: 1) a *primary curriculum* that speeds up the learning of the target context distribution via self-paced RL [25]; and 2) an *auxiliary curriculum* that targets risky and rare contexts under the primary curriculum. The auxiliary curriculum is inspired by a cross entropy method (CEM), which estimates probabilities of rare events [11]. Similar to Greenberg et al.'s work, which does not focus on curriculum learning, we employ CEM to generate a distribution over contexts where the agent's return is below the conditional value of risk (CVaR) of the distribution over returns. In comparison, via CEM, RACGEN generates a sequence of auxiliary context distributions, that identifies rare and risky contexts under primary context distributions produced by the primary curriculum.

Contribution. Our contribution is three-fold, we : 1) identify shortcomings of existing automated curriculum methods under heavy-tailed target context distributions; 2) propose RACGEN, which combines self-paced RL with CEM to simultaneously speed up learning and improve the performance in rare contexts; and 3) demonstrate empirically that, compared to state-of-the-art automated curriculum methods, RACGEN achieves significantly higher returns, with $p<0.001$, in frequent as well as rare and risky contexts.

2 RELATED WORK

We discuss the connections between RL and three subjects related to our work: generalization, curriculum learning, heavy-tail task distributions and risk optimization.

Generalization in RL. We investigate the setting where an RL agent trains on a set of tasks and is deployed to tasks unseen during training. This problem is formulated via contextual Markov decision processes (CMDPs). In this setting, a singleton task refers to an MDP instance described by a context that parameterizes the reward and transition functions [18]. The objective is to maximize the expected discounted return in the MDPs corresponding to the contexts drawn from a probability distribution over the context space of the CMDP. The contexts that an RL agent sees in training and test time are sampled from the same distribution. Therefore, from a generalization perspective, we consider an *interpolation* problem as contexts in test time can be interpolated from contexts seen during training [22]. Under the interpolation subarea, we particularly focus on contextual MDPs where contexts are drawn from a heavy-tailed probability distribution defined over context spaces.

Curriculum learning for RL. Automatically generating curricula in RL aims to accelerate convergence to optimal policies by modifying the configuration of the environment. Numerous works consider curricula as sequences of distributions over such configurations. Florensa et al. [13] focus on distributions over initial states by starting in the neighborhood of the goal state and reversely working towards a target distribution. Other studies propose generating distributions over goal states by optimizing with respect to intrinsic motivation [4, 33], intermediate goal difficulty [14], value disagreement [46], and feasibility and coverage of goal states [34]. Another line of work takes the perspective of generating distributions of levels, i.e., environment instances, that prioritizes higher learning potential [21, 20]. Our work falls under self-paced RL, a curriculum learning approach adopted from supervised learning where training samples are automatically ordered in increasing complexity [28, 19]. Ren et al. [37] consider curricula as a sequence of environment interactions and proposes a self-paced mechanism that minimizes coverage penalty. Eimer et al.'s work generates a sequence of contexts, not distributions, with respect to their capacity of value improvement [12]. Klink et al. [23, 24, 25, 26], Koprulu and Topcu [27] formulate the generation of curricula as interpolations between distributions over contexts. Chen et al. [9] also study interpolations between task distributions, but not under the self-paced RL framework. Although they do not consider risk as a safety metric, Turchetta et al. [41] proposes an approach for generating curricula in safety-critical applications. When the student behaves dangerously, the teacher intervenes by activating reset controllers that take the student to a safe state.

RL under heavy-tailed task distributions. Some supervised learning algorithms have considered learning under heavy-tailed distributions, such as in computer vision by *Long-tailed Image Net* benchmark [30] and only a few works that particularly concentrate on rare events or heavy-tailed task distributions in RL. Frank et al. [15] devise an importance sampling approach to alter probabilities of rare events in simulation data for a tabular setting. Chan et al. [8] is the first work that investigates the shortcomings of Deep RL algorithms in rare events, sampled from Zipfian distributions, which are heavy-tailed and fall under the family of power law distributions. In addition, Zhuang and Sui [47] propose no-regret RL algorithms for settings with rewards that follow heavy-tailed distributions. To our knowledge, our work is the first work that proposes an automated curriculum learning method to address heavy-tailed task distributions.

Risk optimization in RL. Minimizing risk in RL aims to learn policies that maximize performance while satisfying safety requirements during training and test time [16]. To this aim, Tamar et al. [39] proposes a policy gradient algorithm for general coherent risk measures, among which CVaR is very popular [40, 36, 44]. Greenberg et al. [17] focuses on CVaR optimization in a multi-task setting and presents a risk-averse RL algorithm that combines risk-optimizing policy gradient methods with CEM that identifies and samples risky tasks. Although we take inspiration from Greenberg et al. [17], we do not optimize risk in RL. Instead, we utilize CEM in curriculum generation to sample rare and risky contexts, namely, tasks, under context distributions generated by primary curricula.

3 CONTEXTUAL MDP

We formalize our problem of interest as a contextual RL problem, which uses contextual Markov decision processes (CMDPs) to model a multi-task setting given a distribution over target contexts. Upon introducing these concepts, we continue laying the foundations for self-paced RL and cross-entropy methods, which we adopt to generate primary and auxiliary curricula, respectively.

Definition 1 *A* contextual Markov decision process (CMDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{C}, \mathsf{M}, \gamma \rangle$ *is defined by a state space* $\mathcal{S}$*, an action space* $\mathcal{A}$*, a context space* $\mathcal{C} \subseteq \mathbb{R}^n$ *for* $n \in \mathbb{Z}^+$*, a mapping from context space to Markov decision process parameters* M*, and a discount factor* γ*.*

A CMDP $\mathcal{M}$ represents a family of MDPs parameterized by its contexts $\mathcal{C}$. Given a context $\mathbf{c} \in \mathcal{C}$, we obtain an MDP $\mathsf{M}(\mathbf{c}) = \langle \mathcal{S}, \mathcal{A}, p_{\mathbf{c}}, r_{\mathbf{c}}, p_{0,\mathbf{c}}, \gamma \rangle$, where $\mathcal{S}$, $\mathcal{A}$, and γ are the same state space as in $\mathcal{M}$, but its probabilistic transition function $p_{\mathbf{c}}$, reward function $r_{\mathbf{c}}$, and initial state distribution $p_{0,\mathbf{c}}$ depend on its context $\mathbf{c}$. A policy $\pi : \mathcal{S} \times \mathcal{C} \to \Delta(\mathcal{A})$, which defines the behavior of an agent in a CMDP $\mathcal{M}$, outputs a probability simplex over action space $\mathcal{A}$ given state $\mathbf{s} \in \mathcal{S}$ and context $\mathbf{c} \in \mathcal{C}$. Note that the agent observes the context $\mathbf{c}$. Following policy π, an agent collects a trajectory $\boldsymbol{\tau} = \{(\mathbf{s}_t, \mathbf{c}, \mathbf{a}_t, r_t)\}_{t=0}^{T}$ of length T with an initial state $\mathbf{s}_0 \sim p_{0,\mathbf{c}}$, states $\mathbf{s}_{t+1} \sim p_{\mathbf{c}}(\cdot|\mathbf{s}_t, \mathbf{a}_t)$, actions $\mathbf{a}_t \sim \pi(\cdot|\mathbf{s}_t, \mathbf{c})$, and rewards $r_t = r_{\mathbf{c}}(\mathbf{s}_t, \mathbf{a}_t)$ for times steps $t \in [T]$.

Given a CMDP $\mathcal{M}$ and a target context distribution φ, i.e., a probability simplex $\Delta(\mathcal{C})$ over context space $\mathcal{C}$, *contextual RL* aims to learn a policy that maximizes the expected discounted return in contexts $\mathbf{c}$ drawn from φ:

$$\max_{\pi} J(\pi, \varphi) = \max_{\pi} \mathop{\mathbb{E}}_{\mathbb{P}_{\mathbf{c}}^{\pi}(\boldsymbol{\tau}), \varphi(\mathbf{c})} [G(\boldsymbol{\tau})], \tag{1}$$

where $G(\boldsymbol{\tau}) = \sum_{t=0}^{T} \gamma^t r_{\mathbf{c}}(\mathbf{s}_t, \mathbf{a}_t)$ is the discounted return for trajectory $\boldsymbol{\tau}$, and $\mathbb{P}_{\mathbf{c}}^{\pi}(\boldsymbol{\tau})$ is the probability distribution of trajectory $\boldsymbol{\tau}$ induced by policy π in context $\mathbf{c}$ as: $\mathbb{P}_{\mathbf{c}}^{\pi}(\boldsymbol{\tau}) = p_{0,\mathbf{c}}(\mathbf{s}_0) \prod_{t=0}^{T} \pi(\mathbf{a}_t|\mathbf{s}_t, \mathbf{c}) p_{\mathbf{c}}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$.

Contextual RL formulates an optimal decision-making problem that we attempt to solve in this paper. Particularly, we focus on heavy-tailed target context distributions under which rare and risky contexts pose a challenge: An agent requires more samples in a risky context, as it is non-trivial to acquire an optimal behavior. When this context falls under the tails of the target context distribution φ, simply using the target distribution φ prevents the agent from obtaining sufficiently many samples in a sample-efficient manner. In addition, a learning algorithm can get stuck in local optima while maximizing the expected discounted return $J(\pi, \varphi)$ by overlooking rare contexts. The literature on automated curriculum generation fails to address this phenomenon by solely focusing on exponentially-bounded target context distributions where contexts are either equally likely or not so spread out, e.g. under a uniform or a normal distribution, respectively (see Section 5.1 for a detailed discussion).

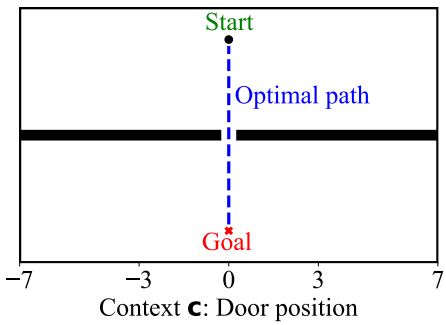


Figure 1: Point-mass environment with 1D context space: Context $\mathbf{c}$ determines the position of the door.

Problem statement. Given a CMDP $\mathcal{M}$ to describe the parameterization of a set of tasks via contexts, and a *heavy-tailed* target context distribution φ to specify their probability of occurrence, *sample-efficiently* learn a policy π that maximizes the expected discounted return $J(\pi, \varphi)$ in $\mathcal{M}$.

Figure 1 shows an example domain, called the *point-mass* environment, where *a context specifies the position of the door*. The agent must reach the goal position by passing through the door. An episode terminates when the agent hits the wall or reaches the goal. The state space, i.e., all possible positions of the agent, and the action space, i.e., forces applied to the point mass along two axes, are independent of the context. However, the context affects the transitions, e.g., whether the agent ends up in the wall, and the rewards, e.g., if the agent receives a reward for approaching the goal position without hitting the wall.

For the point-mass environment, an example target context distribution is a univariate normal distribution $\mathcal{N}(\mu, \sigma^2)$, not heavy-tailed, with mean μ and standard deviation σ over context values, i.e., door positions, in context space $\mathcal{C}$.

4 CONTEXTUAL RL

In this section, we review two methods that aim to solve CMDPs from different perspectives.

4.1 SELF-PACED RL

Self-paced RL [25] is an automated curriculum generation approach that creates a sequence $\{\varrho_k\}_{k=1}^{K}$ of context distributions ϱ_k to learn a policy π that maximizes $J(\pi, \varphi)$ given a CMDP $\mathcal{M}$ with a target context distribution φ. The algorithm starts with an initial context distribution ϱ_0, under which *easy* contexts are more likely to occur. Looking at the

Algorithm 1 Self-paced RL [25]

Input: Target and initial context distributions φ, ϱ_0
Parameters: Performance constraint δ, KL divergence bound ϵ, number of curriculum iterations K, number of rollouts per policy update M
Output: Policy π

1: Initialize policy π
2: **for** $k = 1$ **to** K **do**
3: $\quad \mathbf{c}_i \sim \varrho_{k-1}, i \in [M]$ ▷ *sample contexts*
4: $\quad \mathcal{D}_k = \{(\mathbf{c}_i, \boldsymbol{\tau}_i) | \boldsymbol{\tau}_i \sim \mathbb{P}^{\pi}_{\mathbf{c}_i}(\boldsymbol{\tau})\}_{i=1}^{M}$ ▷ *collect trajectories*
5: $\quad \pi \leftarrow \Psi(\mathcal{D}_k, \pi)$ ▷ *update policy with RL algorithm* Ψ
6: $\quad \varrho_k \leftarrow \Phi_{\varphi}(\pi, \mathcal{D}_k, \varrho_{k-1})$ ▷ *new context distribution (2)*
7: **end for**
8: **return** π

Algorithm 2 CEM variant [17]

Input: Context distribution ϱ, risk level q_α, policy π
Parameters: Number of iterations I, batch size N, risk level α, smoothing risk level β
Output: Auxiliary context distribution $\tilde{\varrho}$

$\tilde{\varrho} \leftarrow \varrho$ ▷ *Initialize auxiliary context distribution*
for $i = 1$ **to** I **do**
$\quad (\mathbf{c}_n, \boldsymbol{\tau}_n) \sim \mathbb{P}^{\pi}_{\tilde{\varrho}}(\mathbf{c}_n, \boldsymbol{\tau}_n), n \in [N]$ ▷ *collect trajectories*
$\quad \mathcal{G} \leftarrow \{G(\boldsymbol{\tau}_n)\}_{n=1}^{N}$ ▷ *compute returns*
$\quad \omega_n \leftarrow \varrho(\mathbf{c}_n)/\tilde{\varrho}(\mathbf{c}_n), n \in [N]$ ▷ *compute IS weights*
$\quad q \leftarrow \max\{q_\alpha(\mathcal{G}), q_\beta(\mathcal{G})\}$ ▷ *estimate quantile*
$\quad \tilde{\varrho} \leftarrow \arg\max_{\tilde{\varrho}'} \sum_{n=1}^{N} \omega_n \mathbf{1}_{G(\boldsymbol{\tau}_n) \leq q} \log \tilde{\varrho}'(\mathbf{c}_n)$ ▷ *new auxiliary context distribution*
end for
return $\tilde{\varrho}$

point-mass environment, an easy context has a door positioned in the middle of the room and thus yields the highest return under an optimal policy.

Algorithm 1 provides more details about the method. At iteration k, first, the algorithm samples contexts $\{\mathbf{c}_i\}_{i=1}$ from the current context distribution ϱ_{k-1} (Line 3), and rolls out policy π to collect a set of trajectories $\mathcal{D}_k$ (Line 4). Then, using $\mathcal{D}_k$, policy π is updated via an RL algorithm of choice. Finally, the algorithm generates the next context distribution ϱ_k, which minimizes the KL divergence to the target context distribution φ:

$$\begin{aligned} \Phi_{\varphi}(\pi, \mathcal{D}_k, \varrho_{k-1}) = \arg\min_{\varrho_k} \; & D_{\mathrm{KL}}(\varrho_k || \varphi) \\ \text{s.t.} \; & J(\pi, \varrho_k) \geq \delta, \\ & D_{\mathrm{KL}}(\varrho_{k-1} || \varrho_k) \leq \epsilon, \end{aligned} \tag{2}$$

where there are two constraints: 1) the expected discounted return $J(\pi, \varrho_k)$ under the next context distribution ϱ_k should be equal to or greater than the desired level of performance δ, and 2) the maximum KL divergence between the current context distribution ϱ_{k-1} and the next context distribution ϱ_k should be less then the divergence bound ϵ. The performance constraint guarantees that the agent collects sufficiently large returns. In parallel, the KL divergence constraint prevents the curriculum from diverging too much from the previous context distribution, which could result in performance loss as past experience becomes less valuable. To estimate $J(\pi, \varrho_k)$, self-paced RL uses the following unbiased sample average given M trajectories:

$$J(\pi, \varrho_k) = \frac{1}{M} \sum_{i=1}^{M} \frac{\varrho_k(\mathbf{c}_i)}{\varrho_{k-1}(\mathbf{c}_i)} \sum_{t=0}^{T_i} \gamma^t r_{\mathbf{c}_i}(\mathbf{s}_t, \mathbf{a}_t),$$

where T_i is the length of the i-th trajectory. Equation (2) can be solved via any constrained optimization algorithm, such as trust-region, as adapted by Klink et al. [23, 24, 25].

Self-paced RL fails under heavy-tailed target context distributions. The existing literature on self-paced RL merely focuses on exponentially-bounded target context distributions and generates a curriculum by taking the expected discounted return $J(\pi, \varrho_k)$ into account. Therefore, they do not address the challenges caused by risky and rare contexts that appear under heavy-tailed target context distributions. We propose a risk-aware curriculum generation method that tackles these challenges by integrating the cross entropy method, which we explain next, into self-paced RL.

4.2 CROSS ENTROPY METHOD

The cross entropy method (CEM) is a generic approach to rare event simulation and optimization [11]. We use CEM to identify and sample risky contexts from the primary context distribution, thus CEM does not aim to learn a policy.

We call a context $\mathbf{c}$ *risky* if the discounted return $G(\boldsymbol{\tau})$ of trajectory $\boldsymbol{\tau}$ in $\mathbf{c}$ is below the CVaR of the return distribution. CVaR is a popular risk measure defined as $\mathrm{CVaR}_\alpha(X) = \mathbb{E}[X | X \leq q_\alpha(X)]$, where $q_\alpha(X) = \min\{x | F_X(x) \geq \alpha\}$ is the α-quantile of a random variable X with cumulative distribution function F_X. To adapt CVaR to our setting, we take inspiration from Greenberg et al. [17] and define CVaR as $\mathrm{CVaR}_\alpha(G|\pi, \varrho) = \mathbb{E}[G | G \leq q_\alpha(G|\pi, \varrho)]$, where $q_\alpha(G|\pi, \varrho) = \min\{g | F_{G|\pi,\varrho}(g) \geq \alpha\}$. In other words, CVaR is the expectation of the lowest q_α-fraction of returns obtained by policy π in contexts drawn from context distribution ϱ.

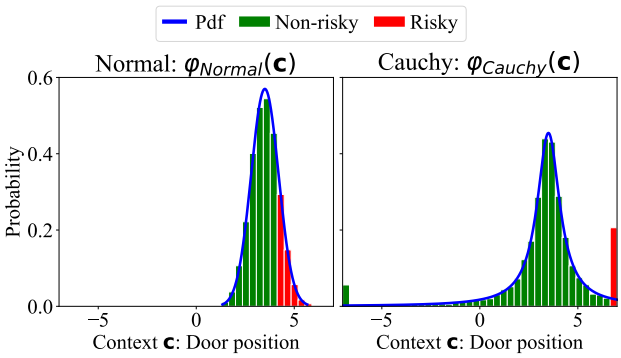
Our goal is to sample context-trajectory pairs $(\mathbf{c}, \boldsymbol{\tau})$ from the distribution $\mathbb{P}^{\pi}_{\varrho,\alpha}(\mathbf{c}, \boldsymbol{\tau}) = \alpha^{-1} \mathfrak{F}(\boldsymbol{\tau}, \pi, \alpha) \mathbb{P}^{\pi}_{\varrho}(\mathbf{c}, \boldsymbol{\tau})$, where $\mathfrak{F}(\boldsymbol{\tau}, \pi, \alpha) = \mathbf{1}[G(\boldsymbol{\tau}) \leq q_\alpha(G|\pi)]$ and $\mathbb{P}^{\pi}_{\varrho}(\mathbf{c}, \boldsymbol{\tau}) = \varrho(c) \mathbb{P}^{\pi}_{\mathbf{c}}(\boldsymbol{\tau})$ is the probability of drawing context $\mathbf{c}$ from context distribution ϱ and collecting trajectory $\boldsymbol{\tau}$ via policy π. In short, given risk-level α, we want to find the distribution $\mathbb{P}^{\pi}_{\varrho,\alpha}$, that is the closest distribution to the tail of the distribution $\mathbb{P}^{\pi}_{\varrho}$. We employ CEM to find a context distribution $\tilde{\varrho}$ for which the distribution $\mathbb{P}^{\pi}_{\tilde{\varrho}}$ is similar to $\mathbb{P}^{\pi}_{\varrho,\alpha}$. To this end, CEM solves the following KL divergence minimization

problem:

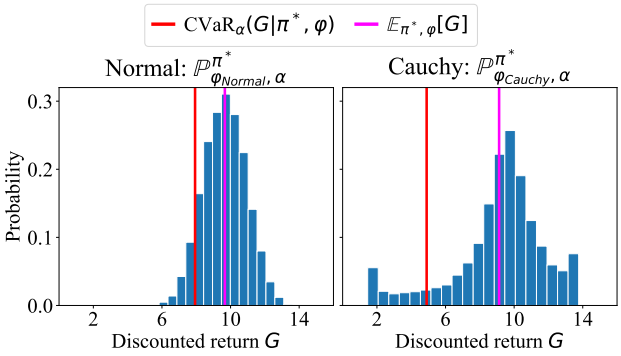
$$\begin{aligned}\tilde{\varrho} &\in \arg\min_{\tilde{\varrho}'} D_{\mathrm{KL}}(\mathbb{P}^{\pi}_{\varrho,\alpha} || \mathbb{P}^{\pi}_{\tilde{\varrho}'}) \\ &= \arg\max_{\tilde{\varrho}'} \mathop{\mathbb{E}}_{(\mathbf{c},\boldsymbol{\tau})\sim\mathbb{P}^{\pi}_{\varrho}} \left[\frac{\mathfrak{F}(\boldsymbol{\tau},\pi,\alpha)\log \tilde{\varrho}'(\mathbf{c})}{\alpha} \right] \\ &= \arg\max_{\tilde{\varrho}'} \mathop{\mathbb{E}}_{(\mathbf{c},\boldsymbol{\tau})\sim\mathbb{P}^{\pi}_{\tilde{\varrho}'}} \left[\frac{\omega(\mathbf{c},\boldsymbol{\tau})\mathfrak{F}(\boldsymbol{\tau},\pi,\alpha)\log \tilde{\varrho}'(\mathbf{c})}{\alpha} \right],\end{aligned} \tag{3}$$

where $\omega(\mathbf{c},\boldsymbol{\tau}) = {}^{\mathbb{P}^{\pi}_{\varrho}(\mathbf{c},\boldsymbol{\tau})}/_{\mathbb{P}^{\pi}_{\tilde{\varrho}'}(\mathbf{c},\boldsymbol{\tau})} = {}^{\varrho(\mathbf{c})}/_{\tilde{\varrho}'(\mathbf{c})}$ is the importance sampling (IS) weight for the context-trajectory pair $(\mathbf{c},\boldsymbol{\tau})$. As the distribution over which the expectation is computed changes from $\mathbb{P}^{\pi}_{\varrho}$ to $\mathbb{P}^{\pi}_{\tilde{\varrho}'}$ in Equation (3), an IS weight is necessary. We provide the pseudocode of CEM variant for sampling risky contexts by Greenberg et al. [17] in Algorithm 2. Given a smoothing risk level $\beta > \alpha$, Line 6 enables smooth updates of context distribution $\tilde{\varrho}$.

Integrating CEM into curriculum generation to sample risky contexts. Greenberg et al. [17] use CEM to identify and sample risky contexts under the target context distribution φ, which can be achieved in Algorithm 2 by replacing input ϱ with φ. This method does not focus on curriculum generation, hence it does not benefit from performance and convergence advantages that come with curriculum learning in RL [31]. In the next section, we address this gap by proposing a risk-aware curriculum generation method, where CEM takes the context distribution ϱ_k from curriculum iteration k to generate an auxiliary distribution $\tilde{\varrho}_k$ identifying the risky contexts under ϱ_k.



(a) Normal and Cauchy target context distributions



(b) Distribution of returns under Normal and Cauchy target

Figure 2: Pitfalls of heavy-tailed target context distributions in the point-mass environment: (a) Non-risky and risky contexts for $\alpha = 0.2$; and (b) distributions of returns for an optimal policy under φ_{Normal} and φ_{Cauchy}.

5 RISK-AWARE CURRICULUM GENERATION

Building upon our problem of interest, contextual RL, we first discuss the challenges that emerge with heavy-tailed context distributions. Then, we present a risk-aware curriculum generation algorithm that adopts self-paced RL and CEM to address these challenges.

5.1 PITFALLS OF HEAVY-TAILED TASK DISTRIBUTIONS

A probability distribution is *heavy-tailed* if its tails are not exponentially-bounded; intuitively, they are heavier than the tails of the exponential probability distribution [3]. Extreme events or outliers are more likely to occur under heavy-tailed distributions, as the area under the extreme regions of the distribution is larger than the area under the tails of an exponentially-bounded probability distribution [43]. In addition, some moments of a heavy-tailed distribution do not exist. For instance, a Cauchy distribution has no finite moments of order 1 or higher, which causes its mean and variance to be undefined. Therefore, a Cauchy distribution is described by its median l, i.e., location parameter, and its median absolute deviation s, i.e., scale parameter. We focus on Cauchy distributions since they look similar to normal distributions; at the same time, they are considered pathological due to their heavy-tailed nature. For a random variable X drawn from a Cauchy distribution with location l and scale s, the corresponding probability density function is $f(x|l,s) = {}^{1}/_{\pi s}[1 + ({}^{x-l}/_{s})^2]^{-1}$.

Consider the point-mass environment from Figure 1 with one-dimensional context space $\mathcal{C} = [-7, 7]$, corresponding to possible door positions. In Figure 2, we analyze two target context distributions: a normal distribution φ_{Normal} with mean $\mu = 3.5$ and standard deviation $\sigma = 0.7$ (Figure 2a left), and a Cauchy distribution φ_{Cauchy} with location $l = 3.5$ and scale $s = 0.7$ (Figure 2a right).

Let us assume the reward is the negative of the exponential distance to the goal position $r_{\mathbf{c}}(\mathbf{s}_t, \mathbf{a}_t) = -\exp \|\mathbf{s}_t - \mathbf{g}\|_2$, where $\mathbf{g}$ is the position of the goal. Figure 2b shows the distributions of discounted returns of the optimal policy with $\gamma = 0.95$, i.e., $\mathbb{P}^{\pi^*}_{\varphi,\alpha}$, under normal (left) and Cauchy (right) target context distributions, respectively. The distribution of discounted returns under φ_{Cauchy} is

Algorithm 3 Risk-aware curriculum generation (**RACGEN**)

Input: Target context distribution φ

Parameter: Initial context distribution ϱ_0, performance constraint δ, KL divergence bound ϵ, number of curriculum iterations K, number of rollouts per policy update M, smoothing risk level β, final risk level α, initial risk level α_0, risk level scheduling factor ρ

Output: Policy π

1: Initialize policy π.
2: $\tilde{\varrho}_0 \leftarrow \varrho_0$ ▷ *initialize auxiliary context distribution*
3: **for** $k = 1$ **to** K **do**
4: $\{\mathbf{c}_i^{pri} | \mathbf{c}_i^{pri} \sim \varrho_{k-1}\}_{i=1}^{M^{pri}}$ ▷ *sample primary contexts*
5: $\{\mathbf{c}_j^{aux} | \mathbf{c}_j^{aux} \sim \tilde{\varrho}_{k-1}\}_{j=1}^{M^{aux}}$ ▷ *sample auxiliary contexts*
6: $\mathcal{D}_k^{pri} \leftarrow \{(\mathbf{c}_i^{pri}, \boldsymbol{\tau}_i) | \boldsymbol{\tau}_i \sim \mathbb{P}_{\mathbf{c}_i^{pri}}^{\pi}\}_{i=1}^{M^{pri}}$ ▷ *collect primary trajectories*
7: $\mathcal{D}_k^{aux} \leftarrow \{(\mathbf{c}_j^{aux}, \boldsymbol{\tau}_j) | \boldsymbol{\tau}_j \sim \mathbb{P}_{\mathbf{c}_j^{aux}}^{\pi}\}_{j=1}^{M^{aux}}$ ▷ *collect auxiliary trajectories*
8: $\pi \leftarrow \Psi(\mathcal{D}_k, \pi)$, for $\mathcal{D}_k = \mathcal{D}_k^{pri} \cup \mathcal{D}_k^{aux}$ ▷ *update policy with RL algorithm* Ψ
9: $\varrho_k \leftarrow \Phi_\varphi(\pi, \mathcal{D}_k^{pri}, \varrho_{k-1})$ ▷ *new context distribution (2)*
10: $q \leftarrow \max\{\hat{q}_{\alpha_{k-1}}(\{G(\boldsymbol{\tau}) | (\mathbf{c}, \boldsymbol{\tau}) \in \mathcal{D}_k^{pri}\}), \hat{q}_\beta(\{G(\boldsymbol{\tau}) | (\mathbf{c}, \boldsymbol{\tau}) \in \mathcal{D}_k\})\}$ ▷ *estimate quantile*
11: $\Omega \leftarrow \{(\omega, \mathbf{c}, \boldsymbol{\tau}) | \omega = {}^{\varrho_k(\mathbf{c})}/_{\tilde{\varrho}_k(\mathbf{c})}, (\mathbf{c}, \boldsymbol{\tau}) \in \mathcal{D}_k\}$ ▷ *compute IS weights*
12: $\tilde{\varrho}_k \leftarrow \arg\max_{\tilde{\varrho}'} \sum_{(\omega, \mathbf{c}, \boldsymbol{\tau}) \in \Omega} \omega \mathbf{1}_{G(\boldsymbol{\tau}) \leq q} \log \tilde{\varrho}'(\mathbf{c})$ ▷ *new auxiliary context distribution*
13: $\alpha_k \leftarrow \max\{\alpha, 1 - {}^{(1-\alpha)k}/_{(\rho K)}\}$ ▷ *apply soft risk scheduling*
14: **end for**

more spread than its counterpart under φ_{Normal}. Similarly, the expectation $\mathbb{E}[G|\pi, \varphi]$ and conditional value-at-risk $\text{CVaR}_{\alpha=0.2}(G|\pi, \varphi)$ are further apart under the Cauchy distribution. Figure 2a supports this observation by illustrating risky contexts, namely, contexts with returns lower than $\text{CVaR}_\alpha(G)$, in red and non-risky contexts in green. Risky contexts under φ_{Cauchy} pile up on the borders as we clip every sample with respect to the boundaries of the context space. In comparison, φ_{Normal} has risky contexts only under its right tail, closer to its mean.

In a multi-task setting, generalization from one task to another becomes challenging as the environment configuration changes drastically. Similarly, in the point-pass environment, generalizing the behavior learned from one context to another requires the policy to learn how the reward function and the transition function change with respect to the context. If the contexts are further apart in the context space, the generalization will be poorer in comparison to transferring behavior to a context that is similar to the source [45].

Figure 2a highlights that risky contexts can cause challenges in generalization under a Cauchy distribution, as the likely contexts and contexts under tails are quite different. Generalization is less critical under a normal distribution, where 99.617% of the samples occur in the interval between three standard deviations from the mean, i.e., $\mathcal{I} = [\mu - 3\sigma, \mu + 3\sigma]$. In contrast, in a Cauchy distribution with $l = 3.5$ and $s = 0.7$, only 35.0828% of the samples fall into the interval $\mathcal{I} = [l - 3s, l + 3s]$.

Therefore, we argue that to improve generalization under heavy-tailed context distributions, an automated curriculum learning algorithm should identify and oversample risky and rare contexts.

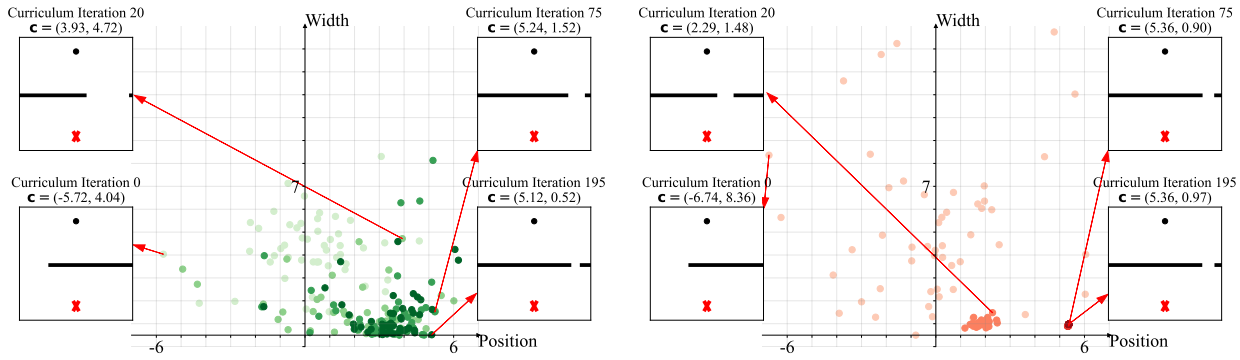
5.2 RISK-AWARE CURRICULUM GENERATION

We propose a risk-aware curriculum generation algorithm, RACGEN, that simultaneously creates two curricula: 1) a *primary* curriculum, i.e., a sequence $\{\varrho_k\}_{k=0}^K$ of context distributions, via a self-paced RL algorithm, and 2) an *auxiliary* curriculum that identifies risky and rare contexts in the primary curriculum via a variant of CEM.

Primary curriculum. Given a target context distribution φ, a self-paced RL algorithm [25] generates a sequence of context distributions $\{\varrho_k\}_{k=0}^K$ by optimizing Equation (2).

Auxiliary curriculum. Upon generating the next primary context distribution ϱ_k at iteration k of the primary curriculum, the auxiliary curriculum outputs the next auxiliary context distribution $\tilde{\varrho}_k$. We propose a CEM variant that achieves this by solving Equation (3) given the current risk-level α_k and the primary context distribution ϱ_k, which corresponds to the reference context distribution in Algorithm 2.

Algorithm 3 presents the pseudocode for the RACGEN method. In summary, at each iteration, the algorithm generates trajectories based on contexts sampled from the primary and auxiliary curricula and updates the policy and the two curricula. More specifically, at iteration k, RACGEN samples M^{pri}-many primary $\mathbf{c}_i^{pri}$ and M^{aux}-many auxiliary $\mathbf{c}_j^{aux}$ contexts from the current primary ϱ_{k-1} and auxiliary $\tilde{\varrho}_{k-1}$ distributions, respectively (Lines 4 and 5). Then, rolling out policy π, it collects two sets of trajectories: primary $\mathcal{D}_k^{pri}$ and auxiliary $\mathcal{D}_k^{aux}$ (Lines 6 and 7). The union $\mathcal{D}_k$ of these sets are used to update policy π via an RL algorithm of choice (Line 8). To generate the next primary context distribution ϱ_k, RACGEN optimizes Equation (2) with the primary trajectory set $\mathcal{D}_k^{pri}$ (Line 9), which completes the



(a) RACGEN's primary context distributions ϱ_k.

(b) RACGEN's auxiliary context distributions $\tilde{\varrho}_k$.

Figure 3: Point-mass environments from contexts sampled at iterations $k \in \{0, 20, 75, 195\}$, which determine the position and the width of the door. Figures 3a and 3b demonstrate how primary and auxiliary contexts (green and red dots, respectively) evolve during the training. The shade of a dot indicates the curriculum iteration, whereas darker shades are of later iterations.

primary curriculum update. The auxiliary curriculum update begins with estimating a risk quantile q (Line 10). Following Greenberg et al.'s approach [17], RACGEN uses a smooth quantile update (Line 10). Then, RACGEN computes IS weights of sampled context-trajectory pairs $(\mathbf{c}, \boldsymbol{\tau}) \in \mathcal{D}_k$ (Line 11). Finally, it generates the next auxiliary context distribution $\tilde{\varrho}_k$ using the estimated quantile q (Line 12), note that this optimization problem has a closed-form solution for some probability distributions, such as a normal or a Cauchy distribution.

Soft-risk scheduling. RACGEN uses soft-risk scheduling to linearly decrease α_k from an initial risk level α_0 to a final risk level $\alpha \leq \alpha_0$. Originally, Greenberg et al. [17] proposes soft-risk scheduling for CVaR policy gradient algorithms to enable policies to learn in contexts with high returns. In contrast, the soft-risk scheduling in RACGEN allows the generation of auxiliary context distribution that focuses on contexts with high returns at first, which allows faster learning at the initial phase of the training. Then, as α_k decreases, risky and rare contexts become the focal point of the auxiliary curriculum. Our empirical results evidence that soft-risk scheduling facilitates not only faster performance increase, but also higher returns at the training's end.

6 EMPIRICAL RESULTS

We set up experiments in **two domains** to investigate the benefits of RACGEN under heavy-tailed target context distributions. We demonstrate the evolution of the primary and auxiliary curricula. Furthermore, we consider **two performance metrics**: 1) the distribution of discounted returns ($G(\boldsymbol{\tau})$) with respect to the target context distribution and 2) its expectation ($J(\pi, \varphi)$). We compare RACGEN with **six state-of-the-art algorithms** for automated curriculum generation: CURROT [26], SPDL [25], PLR [21], VDS [46], GOALGAN [14], and ALP-GMM [33]. Appendix A provides more details about each algorithm. Finally, we include **two baseline methods**: DEFAULT and DEFAULT-CEM. DEFAULT draws contexts from the target context distribution without generating a curriculum. DEFAULT-CEM extends DEFAULT with an auxiliary curriculum generated by inputting the target context distribution to CEM (Algorithm 2). These baselines serve as ablation studies to understand whether generating a curriculum indeed boosts learning performance and speed, and whether targeting rare and risky contexts without a primary curriculum is sufficient, respectively.

6.1 POINT-MASS ENVIRONMENT

We begin with a point-mass environment (Figure 1), that has a two-dimensional context space $\mathcal{C} = [-7, 7] \times [0.5, 14]$, where the context determines the position and the width of the door. Klink et al. [23, 24, 25] study settings where the target context distribution is Gaussian and narrow, referring to small variance, whereas Klink et al. [26] focuses on a bi-modal target context distribution with small variance around each mode. In contrast, the target context distribution in our experimental setting is a Cauchy distribution with location $l = (3.5, 0.5)$ and scale $s = \text{diag}(0.7^2, 0.5^2)$. The initial context distribution is a Cauchy distribution with location $l = (0, 4.25)$ and scale $s = \text{diag}(2^2, 1.875^2)$. Appendix B.1 provides more details. The code to reproduce the experiments is available online[1].

Curriculum generation. Figures 3a and 3b show the progress of the primary and auxiliary context distributions during training, respectively. The primary curriculum starts sampling easy contexts where the door is in the center of the room, and its width is high. The auxiliary curriculum

[1] https://github.com/cevahir-koprulu/risk-aware-curriculum-generation

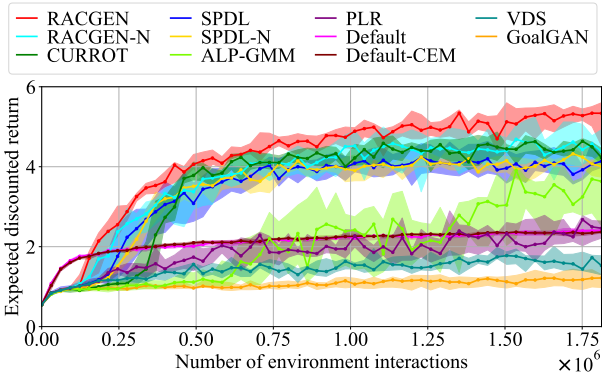


Figure 4: Expected discounted return with respect to the target context distribution in the point-mass environment. The bold lines show the median and the shaded regions cover the first and third quartiles of 10 independent runs.

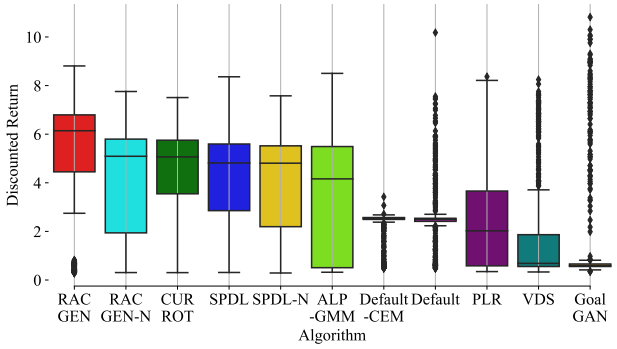


Figure 5: Distribution of the discounted return with respect to contexts drawn from the target context distribution in the point-mass environment over 10 independent training runs. Box plots show the minimum, the first quartile, the median, the third quartile, and the maximum of all returns, from bottom-to-top, whereas the rhombus data samples correspond to outlier values.

follows a similar pattern as $\alpha_0 = 1$, which prevents it from identifying risky and rare contexts, as the objective is to allow the agent to learn the task as quickly as possible. As the training continues, the primary curriculum generates context distributions ϱ_k that approach to the target context distribution φ. In comparison, as α_k decreases linearly, the auxiliary curriculum outputs context distributions $\tilde{\varrho}_k$ that identify the rare and risky contexts under the tails of their corresponding primary context distributions. Figure 3b validates this argument as the last auxiliary contexts (darker shades) are centered around the context $\mathbf{c} \approx (5.4, 0.9)$, approximately 3 median standard deviations away from the median of the target context distribution along the x-axis (door position).

Performance progression. Figure 4 shows the progression of the discounted expected return in the target context distribution during training. We introduce two algorithms in this experiment: RACGEN-N and SPDL-N, which generate normal context distributions only. We evaluate these algorithms because the original SPDL algorithms have a Gaussian assumption [23, 24, 25]. In contrast, RACGEN and SPDL assume the target distribution is Cauchy. We observe that although DEFAULT and DEFAULT-CEM achieve higher returns faster than other algorithms, they stop improving at the early phases of the training. RACGEN attains the highest expected returns and even continues to improve toward the end of the training. CURROT and RACGEN-N perform similarly, despite the fact that CURROT has no risk-aware mechanism. SPDL and SPDL-N also achieve similar expected discounted returns, though Figure 5 shows that SPDL performs slightly better due to having the correct assumption about the type of the context distributions. ALP-GMM, PLR, VDS, and GOALGAN fail to learn policies that receive higher expected returns than DEFAULT and DEFAULT-CEM in the median.

Final performance. Figure 5 shows that RACGEN outperforms all state-of-the-art algorithms, their variations, and the baselines in the experiment. Furthermore, it achieves returns that are significantly higher than the returns of all algorithms according to a Welch's t-test with $p < 0.0001$.

6.2 LUNAR LANDER ENVIRONMENT

In the lunar lander environment [7], the agent must land a pod on planets with varying gravity and wind disturbances. We consider a context space $\mathcal{C} = [-12, -0.01] \times [0, 10]$ that determines the gravity and wind power. We use a Cauchy target context distribution with location $l = (-7, 5)$ and scale $s = \text{diag}(1, 1)$. The initial context distribution is a Cauchy distribution with location $l = (-3.7, 0.)$ and scale $s = \text{diag}(0.25^2, 0.25^2)$, where the median corresponds to a no wind condition in Mars. More details on Appendix B.2. Our analysis focuses on the final performance (Figures 6 and 7), and Appendix B.3 provides the training curves.

Final performance. Figure 6 demonstrates the distribution of the discounted return obtained by the final policies in contexts drawn from the target context distribution. By identifying rare and risky contexts via a CEM module, RACGEN and DEFAULT-CEM achieve discounted return distributions significantly higher than DEFAULT and the rest with $p < 0.01$ and $p < 0.0001$, respectively, according to a Welch's t-test. RACGEN has a higher median and a tighter range than DEFAULT-CEM. In addition, low outlier values are not as spread out as in DEFAULT-CEM, which is an informative observation because outlier low returns particularly occur in rare and risky contexts. Therefore, RACGEN is advantageous over DEFAULT-CEM by generating a primary curriculum in addition to an auxiliary curriculum.

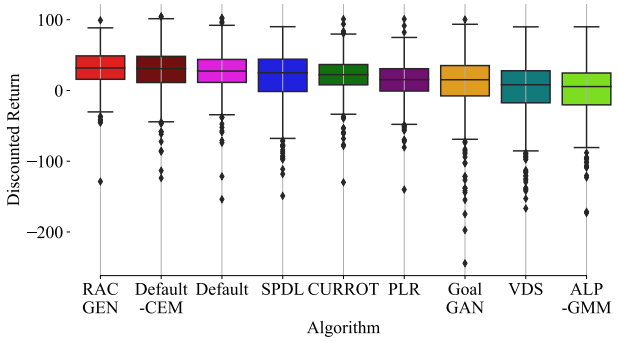


Figure 6: Distribution of the discounted return with respect to contexts drawn from the target context distribution in the lunar lander environment over 5 independent training runs.

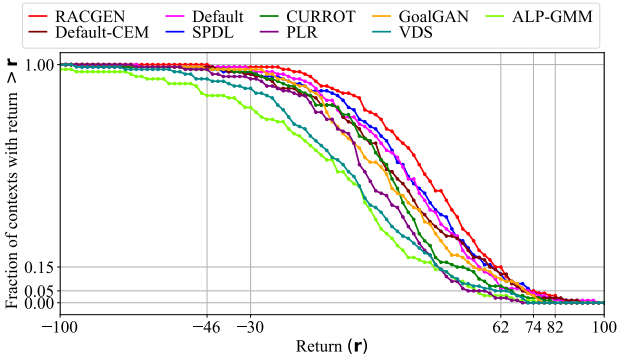


Figure 7: Performance profiles of evaluated algorithms in the lunar lander environment: the fraction of episodes where the final policies achieve discounted returns greater than $\mathbf{r}$. It presents the median over 5 independent training runs.

Performance in risky contexts. We also note that RACGEN does not achieve the highest maximum discounted return. This is likely because RACGEN may overlook more trivial contexts by allocating a portion of its sample budget to auxiliary ones, which are non-trivial and yield low-return. In addition, due to an average of 65% success rate, the lunar lander is a challenging domain under the given target context distribution, which likely results in policies trained via RACGEN to attend risky contexts more. Nevertheless, in terms of first and third quartiles, median, and minimum values, we conclude that RACGEN outperforms the state-of-the-art methods and the baselines in this environment under a heavy-tailed target context distribution.

Performance profiles. Figure 7 further demonstrates that RACGEN achieves higher returns in high and medium-risk contexts than the remaining methods. The figure shows the fraction of contexts (y-axis) where an algorithm learns a policy that achieves a return higher than the return $\mathbf{r}$ (x-axis). The curves show the median over 5 runs. First, we notice that RACGEN almost always achieves returns higher than -46, with DEFAULT following closely and the rest achieving lower returns in high-risk contexts. At $\mathbf{r} = -30$, DEFAULT starts to perform worse than RACGEN, which supports our previous argument that RACGEN achieves the highest minimum returns. The curve of RACGEN stays on the top until $\mathbf{r} = 62$, which demonstrates that RACGEN performs the best in most of the contexts. However, as we previously discussed in Figure 6, RACGEN does not yield the highest returns in low-risk contexts since its curve goes under the others in terms of the portion of contexts with high returns, more specifically for returns $\mathbf{r} \in [62, 74] \cup [82, 100]$.

7 CONCLUSIONS

In this paper, we investigate how to generate curricula in a multi-task setting where the task distribution has a heavy tail. We propose the risk-aware curriculum generation method (RACGEN) that oversamples rare and risky tasks, improving the agent's performance in such tasks. Our empirical evaluation shows that, under a heavy-tail task distribution, RACGEN outperforms state-of-the-art curriculum generation methods that do not take the heavy tail distributions into account. Furthermore, RACGEN has a fast convergence rate, comparable to the state-of-the-art curriculum generation methods, despite deliberately sampling risky tasks.

Limitations. The algorithms that RACGEN employ to generate primary and auxiliary curricula, SPDL and CEM, respectively, search over a fixed parametric family of distributions. Therefore, RACGEN is limited to producing primary and auxiliary context distributions of the same parametric type. Given an arbitrary target context distribution, RACGEN needs to assume that it belongs to a certain parametric family to generate primary and auxiliary context distributions. There, it is likely that the likelihood of some primary or auxiliary contexts would be over or under-estimated. As a result, RACGEN may return sub-optimal policies.

Future Work. We are planning to extend RACGEN to address arbitrary target context distributions. CURROT addresses such limitation of SPDL by replacing KL divergence with Wasserstein distance. Similarly, the generalized version of CEM [6] extends CEM for arbitrary distributions. We can combine CURROT and the generalized CEM to tackle the limitations of RACGEN.

Acknowledgements

This work is supported by the Office of Naval Research (ONR) under grant number N00014-22-1-2254, the National Science Foundation (NSF) under grant number 1646522, the European Research Council (ERC) under the starting grant 101077178 (DEUCE), and the Dutch Research Council (NWO) under the grant NWA.1160.18.238 (PrimaVera).

References

[1] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[2] Minoru Asada, Shoichi Noda, Sukoya Tawaratsumida, and Koh Hosoda. Purposive Behavior Acquisition for a Real Robot by Vision-Based Reinforcement Learning. *Mach. Learn.*, 23(2-3):279–303, 1996.

[3] Søren Asmussen. Steady-state properties of of gi/g/1. In *Applied Probability and Queues*, pages 266–301. Springer New York, 2003.

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

[5] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *JAIR*, 47: 253–279, 2013.

[6] Zdravko I Botev and Dirk P Kroese. The generalized cross entropy method, with applications to probability density estimation. *Methodology and Computing in Applied Probability*, 13(1):1–27, 2011.

[7] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[8] Stephanie CY Chan, Andrew Kyle Lampinen, Pierre Harvey Richemond, and Felix Hill. Zipfian environments for reinforcement learning. In *Conference on Lifelong Learning Agents*, pages 406–429. PMLR, 2022.

[9] 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*, pages 9681–9693, 2021.

[10] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pages 1282–1289. PMLR, 2019.

[11] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134 (1):19–67, 2005.

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

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

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

[15] Jordan Frank, Shie Mannor, and Doina Precup. Reinforcement learning in the presence of rare events. In *ICML*, pages 336–343. ACM, 2008.

[16] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *JMLR*, 16 (1):1437–1480, 2015.

[17] Ido Greenberg, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Efficient Risk-Averse Reinforcement Learning. In *NeurIPS*, 2022.

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

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

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

[21] Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *ICML*, pages 4940–4950. PMLR, 2021.

[22] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *JAIR*, 76:201–264, 2023.

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

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

[25] 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.

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

[27] Cevahir Koprulu and Ufuk Topcu. Reward-machine-guided, self-paced reinforcement learning. In *UAI*. arXiv preprint arXiv:1708.04782, 2023.

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

[29] Philip Kibet Langat, Lalit Kumar, and Richard Koech. Identification of the most suitable probability distribution models for maximum, minimum, and mean streamflow. *Water*, 11(4):734, 2019.

[30] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pages 2537–2546. Computer Vision Foundation / IEEE, 2019.

[31] 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.

[32] Bright O Osu and Johnson Ohakwe. Financial risk assessment with cauchy distribution under a simple transformation of dividing with a constant. *Theoretical Mathematics & Applications*, 1:73–89, 2011.

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

[34] 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.

[35] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *JMLR*, 22(268):1–8, 2021.

[36] Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. EPOpt: Learning Robust Neural Network Policies Using Model Ensembles. In *ICLR*, 2017.

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

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

[39] Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. In *NIPS*, volume 28, 2015.

[40] Aviv Tamar, Yonatan Glassner, and Shie Mannor. Optimizing the cvar via sampling. In *AAAI*, pages 2993–2999. AAAI Press, 2015.

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

[42] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

[43] Rand R Wilcox. Probability and related concepts. In *Applying contemporary statistical techniques*. Elsevier, 2003.

[44] Qisong Yang, Thiago D. Simão, Simon H. Tindemans, and Matthijs T. J. Spaan. Safety-constrained reinforcement learning with a distributional safety critic. *Machine Learning*, 112(3):859–887, 2023.

[45] Amy Zhang, Shagun Sodhani, Khimya Khetarpal, and Joelle Pineau. Learning Robust State Abstractions for Hidden-Parameter Block MDPs. In *ICLR*, 2021.

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

[47] Vincent Zhuang and Yanan Sui. No-regret reinforcement learning with heavy-tailed rewards. In *AISTATS*, pages 3385–3393. PMLR, 2021.

[48] George Kingsley Zipf. *The psycho-biology of language: An introduction to dynamic philology*. Routledge, 2013.