Risk-Aware Continuous Control with Neural Contextual Bandits

Jose A. Ayala-Romero[1], **Andres Garcia-Saavedra**[1], **Xavier Costa-Perez**[2, 1]

[1]NEC Laboratories Europe
[2]i2CAT Foundation and ICREA
{jose.ayala, andres.garcia.saavedra, xavier.costa}@neclab.eu

Abstract

Recent advances in learning techniques have garnered attention for their applicability to a diverse range of real-world sequential decision-making problems. Yet, many practical applications have critical constraints for operation in real environments. Most learning solutions often neglect the risk of failing to meet these constraints, hindering their implementation in real-world contexts. In this paper, we propose a risk-aware decision-making framework for contextual bandit problems, accommodating constraints and continuous action spaces. Our approach employs an actor multi-critic architecture, with each critic characterizing the distribution of performance and constraint metrics. Our framework is designed to cater to various risk levels, effectively balancing constraint satisfaction against performance. To demonstrate the effectiveness of our approach, we first compare it against state-of-the-art baseline methods in a synthetic environment, highlighting the impact of intrinsic environmental noise across different risk configurations. Finally, we evaluate our framework in a real-world use case involving a 5G mobile network where only our approach consistently satisfies the system constraint (a signal processing reliability target) with a small performance toll (8.5% increase in power consumption).

Introduction

Recent progress in the domain of decision-making learning techniques has garnered considerable attention owing to their extensive applicability in diverse real-world sequential decision-making problems (Silver et al. 2016; Brown and Sandholm 2019; Meta Fundamental AI Research Diplomacy Team (FAIR) et al. 2022). Nevertheless, the practical deployment of these techniques necessitates careful consideration of critical operational constraints inherent in real environments. Regrettably, existing learning solutions often overlook the risk associated with violating these constraints, thereby impeding their viability in real-world scenarios.

Motivated by many real-world applications, we address the contextual bandit (CB) problem with constraints, which has been applied to many different problems in diverse fields, e.g., industrial control and temperature tuning (Fiducioso et al. 2019), parameter optimization in robotics (Berkenkamp et al. 2021), mobile networks optimization (Ayala-Romero et al. 2019), or video analytics optimization (Galanopoulos et al. 2021). In this framework, one metric needs to be maximized, while one or more other metrics must be bounded at each time step (step-wise constraints). In practice, performance metrics — whether utility or constraints — often possess random components. These can arise from measurement errors or be intrinsic to the metric, a phenomenon called *aleatoric uncertainty*. Such uncertainty hinders constraint satisfaction, which is a crucial aspect in most applications, further complicating the problem.

Previous works address the aforementioned constrained contextual bandit problem considering long-term budget constraints (Badanidiyuru et al. 2014; Agrawal et al. 2014), which does not fit our setting where the constraints must be satisfied at each step. Other works propose linear contextual bandits with safety constraints (Amani et al. 2019; Kazerouni et al. 2017; Daulton et al. 2019). These solutions aim to achieve at least a percentage of the performance of a baseline policy. However, none of these works consider aleatoric uncertainty, which is a key aspect to design risk-aware decision-making algorithms. Berkenkamp et al. (2021) propose a Bayesian optimization algorithm called SafeOPT that handles noisy observation and constraints at each step as we do. Although SafeOPT is data-efficient, it presents important disadvantages over our solution concerning its computational complexity and requirements on prior knowledge, aspects that we discuss in detail later.

In this paper, we present a novel algorithmic framework for risk-aware decision-making. In particular, we propose an actor multi-critic architecture. We use different critics to separately characterize the distribution of each of the metrics — both utility and constraints. We use these critics to train a deterministic actor that enables our solution to operate in continuous action spaces. Previous works adopt the strategy of learning the mean value of the metric of interest (Mnih et al. 2015; Fujimoto et al. 2018; Zhou et al. 2020a). Other works consider a unique utility function capturing the reward with a Lagrangian-like penalty term (Tessler et al. 2018; Solozabal et al. 2020). However, such strategies can lead to constraint violations that depend on the aleatoric uncertainty inherent in the metrics. In contrast, our approach seeks to characterize the aleatoric uncertainty for each performance metric, which allows us to *modulate the risk level in the decision-making process*. To this end, we introduce a

parameter α that balances between risk and performance.

We evaluate our solution against the most relevant baselines in the literature across two distinct environments. Firstly, in a synthetic environment where the performance metrics are non-linear functions and the set of actions that meets the constraints is highly dependent on the context. Within this environment, we assess the impact of aleatoric uncertainty on algorithmic performance. Secondly, we evaluate our framework in a real-world 5G mobile network experimental platform. The primary goal here is to minimize energy consumption subject to specific system performance requirements. In this setting, we experimentally characterize the aleatoric uncertainty inherent in the system metrics. Our solution not only shows superior constraint satisfaction but also exhibits the capability to modulate risk, effectively balancing performance against constraint satisfaction.

Problem Formulation

We consider a contextual bandit formulation with constraints. At each time step $t = 1, \dots, T$ the learner observes the context $s_t \in \mathcal{S}$, where $\mathcal{S}$ is the context space and then selects a d-dimensional continuous action $a_t \in \mathbb{R}^d$. Based on this, the learner observes the reward $r_t(s_t, a_t)$ and M constraint metrics $c_t^{(m)}(s_t, a_t)$, for $m = 1, \dots, M$. All the $M + 1$ observed metrics are intrinsically random, i.e., they can be written as $c_t^{(m)}(s_t, a_t) = E[c_t^{(m)}(s_t, a_t)] + \zeta_t$, where the noise ζ_t at time t is drawn from an unknown distribution with expectation $E[\zeta_t] = 0$. The behavior of the learner is defined by a policy that maps contexts into actions $\pi : \mathcal{S} \mapsto \mathbb{R}^d$. Our objective is to find the optimal policy:

$$\begin{aligned} &\underset{\pi}{\operatorname{argmax}} \sum_{t=1}^{T} r_t(s_t, \pi(s_t)) && (1)\\ &\text{s.t.} \quad c_t^{(m)}(s_t, \pi(s_t)) < c_{\max}^{(m)}, \qquad m = 1, \dots, M \\ &\qquad\qquad\qquad\qquad\qquad\qquad\quad t = 1, \dots, T \end{aligned}$$

where $c_{\max}^{(m)}$ is the maximum value for constraint m.

Note that, in contrast to the problem addressed in other works on contextual bandits in the literature (Zhou et al. 2020a; Xu et al. 2022; Amani et al. 2019; Kazerouni et al. 2017), we consider a continuous action space, several performance metrics, and stochastic constraints that should be satisfied at each time step.

Proposed Method

We consider an actor-multi-critic architecture with a deterministic actor to deal with the continuous action space (Lillicrap et al. 2015). In contrast to previous works, we consider $M+1$ distributional critics denoted by $R^m(s, a \mid \eta^m)$, where η^m are the parameters of the critics. Note that the critic with index $m = 0$ approximates the reward function $r_t(s_t, a_t)$ and the critics with indexes $m = 1, \dots, M$ approximate constraint $c_t^{(m)}(s_t, a_t)$. We enable the critics to approximate the distribution of their objective metric using quantile regression.

Distributional Critics

Let $F_Z(z)$ be the cumulative distribution function (CDF) of Z. Note that the quantile function is the inverse of the CDF. Hence, for a given quantile $\tau \in [0, 1]$, the value of the quantile function is defined as $q_\tau = F_Z^{-1}(\tau)$. The quantile regression loss is an asymmetric convex function that penalizes overestimation error with weight τ and underestimation error with weight $1 - \tau$:

$$\mathcal{L}^\tau(\hat{q}_\tau) := \mathbb{E}_{z \sim Z}\left[\rho_\tau(z - \hat{q}_\tau)\right], \text{where} \quad (2)$$

$$\rho_\tau(u) := u \cdot (\tau - \delta_{\{u<0\}}) \quad \forall u \in \mathbb{R}, \quad (3)$$

where $\hat{q}_\tau$ is the estimation of the value of the quantile function, and $\delta_{\{x\}}$ is an indicator function that takes the value 1 when the condition x is satisfied and 0 otherwise. Considering that each critic has N outputs that approximate the set $\{q_{\tau_1}, \dots, q_{\tau_N}\}$, the critic can be trained to minimize the following objective using stochastic gradient descent:

$$\sum_{i=1}^{N} \mathcal{L}^{\tau_i}(\hat{q}_{\tau_i}). \quad (4)$$

Note that the quantile regression loss is not smooth when $u = 0$, limiting the performance of non-linear function approximators such as NNs. To address this issue, we use the *quantile Huber loss* (Huber 1992). This loss function has a squared shape in an interval $[-\kappa, \kappa]$, and reverts to the standard quantile loss outside of this interval:

$$L_\kappa(u) := \begin{cases} \frac{1}{2}u^2 & \text{if } |u| \le \kappa \\ \kappa(|u| - \frac{1}{2}\kappa) & \text{otherwise.} \end{cases} \quad (5)$$

Now, we derive an asymmetric variation of the Huber loss,

$$\rho_\tau^\kappa(u) := |\tau - \delta_{\{u<0\}}| \frac{L_\kappa(u)}{\kappa}. \quad (6)$$

Finally, the quantile Huber loss can be obtained by introducing $\rho_\tau^\kappa(u)$ in eq (2). Note that when $\kappa \to 0$ the quantile Huber loss reverts to the quantile regression loss.

Risk-Aware Actor

In order to capture the information provided by all the critics, we define an aggregate reward signal:

$$R^{agg}(s, a, \alpha \mid \eta) := \quad (7)$$

$$\bar{R}^0(s, a \mid \eta^0) - \sum_{i=1}^{M} \lambda \max\left(\gamma^\alpha(R^i(s, a \mid \eta^i)) - c_{\max}^{(i)}, 0\right),$$

where $\gamma^\alpha(Z)$ is the value of the quantile function of distribution Z at quantile α, λ is the penalty constant for the constraints, $\eta = \{\eta^0, \dots, \eta^M\}$ is the joint set of parameters of the $M+1$ critics, and $\bar{R}^m(\cdot)$ indicates the mean of the distribution provided by critic m. That is, the first term in eq. (7) is the mean of the metric we want to maximize, while the second term captures the penalty incurred when violating each constraint. Note that, when $\gamma^\alpha(R^i(s, a \mid \eta^i)) < c_{\max}^{(i)}$, the value inside the $\max()$ function is negative and the penalty terms are zero.

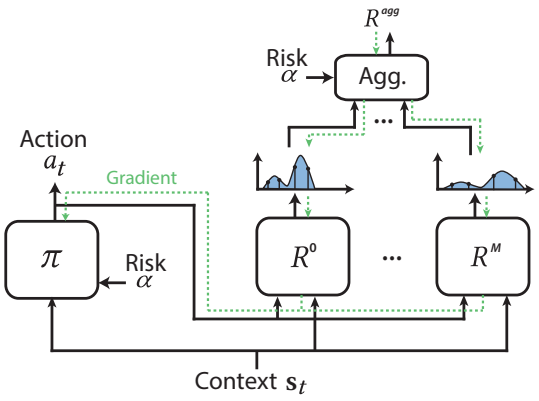

Figure 1: Risk aware decision-making framework comprising a deterministic actor, $M+1$ distributional critics and the aggregation function detailed in eq. (7). The propagation of the gradient to train the actor is shown in green.

Importantly, in contrast to other works using similar weighted penalties in the reward, the use of the quantile function allows us to assure that the tail of the distribution of the constraints (as $\alpha \to 1$) meets the restrictions, making our solution more robust to constraint violations. Note that eq. (7) can be adapted to cases where the constraints set a minimum value by changing the sign to the term inside the maximum operator and choosing a value of α close to zero.

We denote the deterministic actor policy as $\pi(s \mid \theta, \alpha)$, where θ is the set of actor parameters. Then, for a given value of α, we define the actor's objective as

$$J(\pi, \alpha) := \int_{\mathcal{S}} \beta(s)\, R^{agg}(s, \pi(s \mid \theta, \alpha), \alpha \mid \eta)\, \mathrm{d}s \qquad (8)$$
$$= \mathbb{E}_{\mathbf{s}\sim\beta}[R^{agg}(s, \pi(s \mid \theta, \alpha), \alpha \mid \eta)],$$

where $\beta(s)$ is the stationary context distribution. Note that, in a contextual bandit problem, the distribution of the context is not conditioned by the policy.

The actor policy is updated by applying the chain rule to the performance objective defined in eq. (8) with respect to the actor parameters (Silver et al. 2014):

$$\nabla_\theta J(\pi, \alpha) \approx \qquad (9)$$
$$\mathbb{E}_{s\sim\beta}\left[\nabla_a R^{agg}(s, a, \alpha \mid \eta)\mid_{a=\pi(s|\theta,\alpha)} \nabla_\theta \pi(s \mid \theta, \alpha)\right].$$

Note that α is an input of the policy and the aggregated reward (eq. 7). Thus, different values of α can modulate the risk taken by the actor when selecting actions. Specifically, with $\alpha \to 1$ we reduce the probability of violating a constraint. However, this may also imply lower values of reward r_t due to more conservative actions, showing the trade-off between performance and robustness. Moreover, we can configure diverse values of α for each constraint when the constraints have different risk aversion (e.g., some constraints may be more critical than others). This is possible because we consider one critic per constraint that can learn with a different value of α. Then, $R^{agg}(\cdot)$ is computed based on the corresponding values of α from each critic.

Algorithm 1: RANCB training

Input: B, $\mathcal{A}$, α, κ $\mathcal{T}$
Initialize: $\mathcal{D} = \emptyset$, $\mathcal{N}$, θ, η

for $t = 1, \ldots, T$ **do**
  Observe context s_t
  Compute the action $a_t = \pi(s_t, \alpha \mid \theta) + \mathcal{N}_t$
  Observe performance metrics $r_t, c_t^{(1)}, \ldots c_t^{(M)}$
  Store in $\mathcal{D}$ the experience $\langle s_t, a_t, c_t^{(0)} \ldots c_t^{(M)} \rangle$
  Sample a random minibatch of B samples $\langle s_i, a_i, c_i^{(0)} \ldots c_i^{(M)} \rangle$
  for $m = 0, \ldots, M$ **do**
    Update the critic m by minimizing the loss $L = \frac{1}{B}\sum_i \sum_{\tau\in\mathcal{T}} \rho_\tau^\kappa(c_i^{(m)} - \gamma^\tau(R^m(s_i, a_i|\eta^m)))$
  end for
  for all $\alpha_j \in \mathcal{A}$ **do**
    Update the actor with the sampled policy gradient $\frac{1}{B}\sum_i \nabla_a R^{agg}(s_i, a, \alpha_j|\eta)|_{a=\pi(s_i|\theta,\alpha_j)} \nabla_\theta \pi(s_i|\theta, \alpha_j)$
  end for
end for

The proposed framework for risk-aware decision-making is shown in Fig. 1.

Algorithm

We consider a reply buffer $\mathcal{D}$, where the samples of experience $\langle s, a, c^{(0)}(s, a), \ldots, c^{(M)}(s, a)\rangle$ are stored at each time step (Mnih et al. 2015). To simplify the notation, the reward $r(s, a)$ is denoted by $c^{(0)}(s, a)$ in the reply buffer. The gradients used for training are computed using mini-batches of B experience samples randomly gathered from this buffer.

Let α be the default risk value. In most of the applications, a risk-averse policy is desirable, i.e., $\alpha \to 1$. In other cases, we may want to modulate the level of risk to find a different balance between performance and robustness. We define $\mathcal{A}$ as the set of risk values to be used by the algorithm. We define $\mathcal{T}$ as the set of quantiles approximated by all the critics, where $\mathcal{A} \subseteq \mathcal{T}$. For any given minibatch of experience samples, R^{agg} can be computed for different values of risk α (see eq. (7)). Thus, the actor can learn the policy as a function of the risk without collecting extra data. Since the actor policy π is deterministic, we add some noise denoted by $\mathcal{N}$ to the actions to enable exploration during training. Algorithm 1 shows the pseudo-code of our framework, referred to as Risk-Aware Neural Contextual Bandit (RANCB).

Benchmark Algorithms

We first present a set of benchmarks that are variations of our proposal and are inspired by ideas from the literature. In this way, we can conduct an ablation study to evaluate the impact of its different components, i.e., distributional critics and multiple critics. Then, we present SafeOPT (Berkenkamp et al. 2021), the most closely related work to ours. SafeOPT relies on GPs to learn the objective and the constraint functions while handling the intrinsic noise of the observations. To the best of our knowledge, there are no other works in the literature addressing this problem.

Baselines

- **Neural Contextual Bandit (NCB)** is inspired by the actor-critic NN architecture presented by Lillicrap et al. (2015). However, some modifications need to be introduced to adapt this solution to our problem. As NCB only encompasses one critic, we need to define a utility function that captures the constrained problem:

$$u_t(s_t, a_t) := \tag{10}$$
$$r_t(s_t, a_t) - \sum_{i=1}^{M} \lambda \max\left(c_t^{(m)}(s_t, a_t) - c_{\max}^{(i)}, 0\right).$$

  In contrast to the original algorithm (Lillicrap et al. 2015) where future values of reward are also taken into account, the NCB critic approximates the expectation of u_t. For that purpose, we use the MSE loss function:

$$L(\eta) := \mathbb{E}_{s\sim\beta, a\sim\pi'}\left[\left(R(s,a|\eta) - u(s,a)\right)^2\right] \tag{11}$$

  where $R(s,a|\eta)$ denotes the critic and π' any policy that can potentially deviate from the actor's behavior. The actor is updated as indicated by Lillicrap et al. (2015).
- **Single-Critic Distributional NCB (SC-DNCB)** extends NCB with a distributional critic to characterize the distribution of u_t. The critic is trained using the Huber loss in eq. (6) and the actor uses the policy gradient equation proposed by Lillicrap et al. (2015) with respect to the expectation of the distribution provided by the critic. Note that, as the reward and all the constraints are characterized by a single utility function, the level of risk tolerance in the decision-making process cannot be configured. However, it has been widely reported in the literature that the use of distributional critics increases the performance of the algorithms even when the critic is only used to compute the expected value of the distribution (Bellemare et al. 2017; Dabney et al. 2018b).
- **Multi-Critic NCB (MC-NCB)** extends NCB by including $M+1$ non-distributional critics, one per performance metric. Each critic approximates the expectation of its corresponding metric using the MSE loss. To update the actor, an aggregated reward signal is computed based on the output of all the critics, similarly to eq. (7). Then, the actor gradients are computed using eq. (9).

All these baseline solutions use the same exploration approach used by RANCB in the training phase. Note that none of these baselines allow us to configure the level of risk during the decision-making process as RANCB does.

Bayesian Optimization with Constraints

Finally, we also use SafeOPT as a benchmark (Berkenkamp et al. 2021), which is a Bayesian online learning algorithm that handles constraints and noisy observations. SafeOPT comprises $M + 1$ GPs that characterize each one of the performance and constraint metrics. We implement the contextual version of SafeOPT and follow the implementation provided in Sec. 4.3 of that paper, where the confidence intervals of the GPs are used to compute the safe set of actions (actions that satisfy the constraint for a given context). The confidence is determined by a scalar β (see eq. (10) in (Berkenkamp et al. 2021)). We consider this practical version of the algorithm because we assume that, in general, the Lipschitz continuity properties of the performance and constraint functions are unknown.

We configure SafeOPT with a combination of the anisotropic version of the Matér kernel with $\nu = \frac{3}{2}$ and a white kernel to model the noise (Duvenaud 2014). We use UCB as an acquisition function since it optimizes the reward and expands the safe set of actions at the same time. We found that this strategy provides higher performance compared to the exploration strategy proposed originally by SafeOPT, which expands the safe set explicitly. We also note that this issue has been reported in other previous works (Fiducioso et al. 2019; Ayala-Romero et al. 2021).

During the execution of SafeOPT, it may happen that none of the actions satisfies the conditions to be in the safe set, e.g., due to a large value of β or high noise in the observations. The original algorithm does not consider that this event may happen in practice. To address this issue, we modified SafeOPT to select the action that minimizes the estimation of the accumulated constraint violation (across all the constraints) when the safe set is empty.

Evaluation

We evaluate all the aforementioned algorithms in two settings: (i) using a synthetic environment with non-linear functions and variable noise in the observations, and (ii) in a real-world resource allocation problem in wireless networks implemented on a real system. The source code of our solution and all the baselines is available online[1].

In our evaluation, we configure all actor and critic NNs with two hidden layers of 256 units. The critics that approximate the reward function learn the set of quantiles $\mathcal{T} = \{i/N \mid i = 1, \ldots, N\}$, where $N = 21$. For the critics that approximate constraints, we consider two different configurations depending on whether the constraint sets a maximum value (synthetic environment) or a minimum value (resource allocation problem in wireless networks). For the former, we use $\mathcal{T}^{\max} = \{0.1, 0.3, 0.5, 0.7, 0.8, 0.9, 0.99, 0.995, 0.999\}$ and $\alpha^{\max} = 0.995$; and for the latter $\mathcal{T}^{\min} = \{1 - \tau \mid \tau \in \mathcal{T}^{\max}\}$ and $\alpha^{\min} = 0.005$. We used Adam to learn the NN parameters with a learning rate of 10^{-4} and 10^{-3} for the actor and critics, respectively. For the exploration noise $\mathcal{N}$, we use an Ornstein-Uhlenbeck process with the parameters $\theta_{\text{noise}} = 0.15$ and $\sigma_{\text{noise}} = 0.15$ to generate temporally correlated perturbations to the selected action (Lillicrap et al. 2015). We use a reply buffer $\mathcal{D}$ with a memory of 2000 samples. Finally, we configure $\kappa = 1$, a minibatch size of $B = 64$ samples, and $\lambda = 2.5$ (see the Appendix for a detailed evaluation).

For all the results shown in this section, we consider 10 independent runs. The figures with shadowed area show the average and the 15$^{\text{th}}$ and 85$^{\text{th}}$ percentiles. The figures with error bars show the mean values and the confidence intervals with a confidence level of 0.95.

[1] https://github.com/jaayala/risk_aware_contextual_bandit

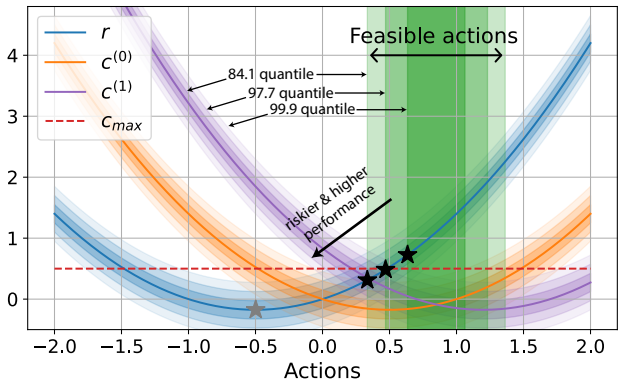

Figure 2: Representation of the synthetic environment defined in eq. (12) for a fixed context $s = (0.7, 0.7, 0.7)$ and $\sigma_{\text{env}} = 0.15$. We depict the 84.1$^{\text{th}}$, 97.7$^{\text{th}}$, and 99.9$^{\text{th}}$ quantiles of the functions with different transparency levels and their corresponding sets of feasible actions. The markers show the optimal values of the unconstrained (grey) and constrained (black) problems.

Synthetic Environment

In the first set of experiments, we consider a synthetic environment with 3-dimensional contexts and a one-dimensional action space ($d = 1$). The reward and the constraints are given by the following quadratic functions:

$$\begin{aligned} r_t(s_t, a_t) &= s_t^{(0)} \cdot a_t^2 + s_t^{(1)} \cdot a_t + \xi_t^{(0)} \\ c_t^{(1)}(s_t, a_t) &= s_t^{(0)} \cdot a_t^2 - s_t^{(1)} \cdot a_t + \xi_t^{(1)}, \\ c_t^{(2)}(s_t, a_t) &= s_t^{(0)} \cdot (a_t - s_t^{(2)})^2 - s_t^{(1)} \cdot (a_t - s_t^{(2)}) + \xi_t^{(2)}, \end{aligned} \tag{12}$$

where $\xi_t^{(i)} \sim N(0, \sigma_{\text{env}}^2)$ for $i = 0, 1, 2$. In our experiments, the contexts are generated as i.i.d. uniform random variables in $[0, 1]^3$. We set the constraint bounds as $c_{\text{max}}^{(0)} = c_{\text{max}}^{(1)} = 0.3$.

Fig. 2 shows an example of the functions in eq. (12) for a fixed context $s = (0.7, 0.7, 0.7)$ and $\sigma_{\text{env}} = 0.15$. In this example, the lower value of the feasible action set is outlined by $c_t^{(2)}$ and the higher value is delimited by $c_t^{(1)}$. Note that the location and shape of all of these functions are highly dependent on the context s.

We plot with different transparencies the 84.1$^{\text{th}}$, 97.7$^{\text{th}}$, and 99.9$^{\text{th}}$ quantiles of the functions in eq. (12). We also plot the feasible sets of actions obtained when considering that each of those quantiles needs to satisfy the constraint. Note that when considering higher quantiles, the safe set becomes smaller but safer, i.e., the probability of satisfying the constraints is higher and vice versa. The optimal values of the reward r for each feasible set of actions are marked with a black star, and we use a grey star for the optimal unconstrained value. Note that the riskier the set of feasible actions, the higher the performance of the optimal action within the set, i.e., we get closer to the grey star.

In other words, a higher reward implies a higher probability of violating the constraint. Therefore, if we want to satisfy the constraint with high probability, we need to be more conservative in decision-making, which has a cost in terms of reward. This trade-off also depends on the variance of the noise of the performance metrics, modeled by σ_{env}^2.

Fig. 3 compares the performance of all the solutions during training. The right plot shows the instantaneous reward ($-r_t(s_t, a_t)$), and the left plot shows the accumulated constraint violation defined as follows:

$$\Gamma_t := \sum_{m=1}^{M} \sum_{t'=0}^{t} \max \left\{ c_{t'}^{(m)} - c_{\text{max}}^{(i)}, 0 \right\}. \tag{13}$$

We observe that RANCB with $\alpha = 0.995$ not only provides the minimum values of Γ_t but also the slope of Γ_t tends to zero. This means that the constraint violation after convergence is very small in this setting. Obviously, this outstanding reliability performance comes at a cost in terms of reward as depicted in the right plot. Conversely, RANCB with $\alpha = 0.5$ provides the highest reward but pays the price of higher accumulated constraint violations. This result illustrates how RANCB can adapt to any application reliability target by setting α appropriately. The rest of the benchmarks are unable to adjust the level of risk and, therefore, they converge to intermediate solutions.

Note that Fig. 3 does not include SafeOPT. The reason is that it is not feasible to provide a fair comparison of the training performance between SafeOPT and the rest of the algorithms due to some fundamental differences. On the one hand, differently than our approach, SafeOPT needs some previous knowledge before starting the training phase. First, SafeOPT needs a dataset to optimize the hyperparameters of the kernels. As the kernels encode the smoothness of the metric function, this step is critical. A suboptimal hyperparameter optimization (e.g., due to a poor dataset) may have serious consequences on training performance. In this particular example, there are 5 hyperparameters to optimize for each GP, i.e., 4 dimensions (3-dimensional contexts and 1-dimensional action) plus the noise level. In our evaluations, we optimized the kernels using 1000 samples obtained randomly from the environment. Second, we need to define an initial safe set of actions, which will be used at the beginning of the training phase. This step requires some domain knowledge and can be very challenging as the safe set of actions can be highly dependent on the context. We found that, if the actions in the initial safe set violate the constraint, the algorithm does not converge. To avoid this, we use eq. (12) in the first iterations of the training phase to compute an initial safe set of actions. Note that this gives SafeOPT some advantage over the other benchmarks, hindering a fair comparison. Moreover, this strategy to generate the initial safe set is not realistic in a real-world application.

On the other hand, GP-based learning algorithms are known to be more data efficient than NN-based algorithms, that is, they need fewer data samples to converge. However, the computational complexity of GP-based solutions is $O(n^3)$ with the sample size (Williams and Rasmussen 2006). To illustrate this, Fig. 4 shows the execution times of SafeOPT and RANCB in an Intel i7-11700 @ 2.5GHz with 15Gb of RAM. The execution time of SafeOPT increases exponentially with the sample size, while the inference time of

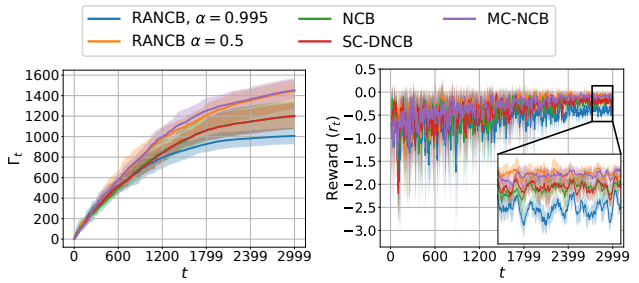

Figure 3: Evaluation of training phase in synthetic environment with $\sigma_{\text{env}} = 0.2$. Accumulated constraint violation Γ_t (left); instantaneous reward $-r_t(s_t, a_t)$ (right).

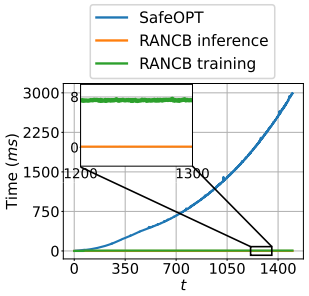

Figure 4: Evaluation of execution time in a Intel i7-11700 @ 2.5GHz and 15Gb or RAM.

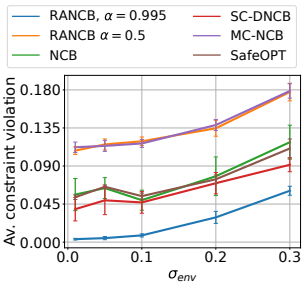

Figure 5: Evaluation of inference performance. Average constraint violation per step as a function of σ_{env}.

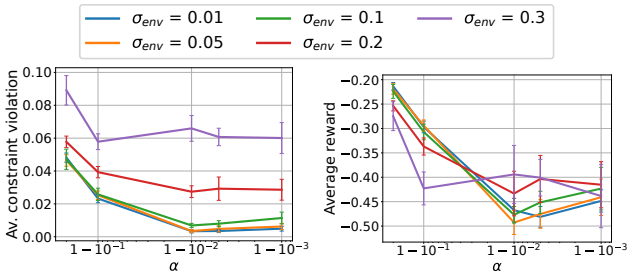

Figure 6: Impact α on the execution performance of RANCB. Average constraint violation per step (left) and average reward (right).

RANCB is $0.106 \pm 5 \cdot 10^{-4}$ ms^2. Additionally, we measured that the execution time for the hyperparameter optimization of SafeOPT is 160.7 ± 15.1 seconds for each GP. Therefore, GP-based solutions can be unfeasible in scenarios where the computational capacity is limited or the decisions need to be made synchronously or in a timely manner (as in the wireless network example that we show later).

Let us now compare the performance of all the benchmarks during the inference operation (after the training phase). Fig. 5 shows the average constraint violation as a function of σ_{env}. Note that, with larger values of σ_{env}, there are contexts for which the penalty term in eq. (7) is not zero for any action, i.e., the constraint violation is inevitable due to the high variance. Hence, we observe a general increase of the constraint violation with σ_{env}. In such cases, the algorithms need to select the action that minimizes the cost due to constraint violation. In all the cases, RANCB with $\alpha = 0.995$ obtains the minimum constraint violation. Moreover, we found that optimal values of the hyperparameter β of SafeOPT are highly dependent on σ_{env}. We evaluated several values of β and selected for each σ_{env} the one attaining the lowest constraint violation, $\beta = \{90, 15, 10, 3.5, 2\}$ for each value in the x-axis of Fig. 5, respectively.

Finally, Fig. 6 shows the impact of α on the inference performance of RANCB for different values of σ_{env}. As expected, when α increases, the constraint violations decrease (left plot). We also observe that lower values of α are associated with higher reward (right plot), which shows again the trade-off between constraint satisfaction and performance.

Resource Assignment in Mobile Networks

For every Transmission Time Interval (TTI) of 1 ms or lower, wireless processors such as those in 5G must process signals that encode data, known as Transport Blocks (TB), within hard time deadlines. Failing to meet such deadlines may result in TB data loss (Foukas et al. 2021). To provide industry-grade reliability, today's wireless processors use hardware accelerators (HAs) that can swiftly process these signals. However, it is well-known that HAs are energy-hungry, and energy consumption is nowadays a major concern for mobile operators (GSMA Association 2020; China Mobile Limited 2021). Alternatively, software processors, which use inexpensive CPUs, are more energy-efficient but are slower than HAs, potentially risking deadline violations.

Importantly, the processing time of these signals (in a software processor or an HA) is difficult to predict as it depends on several and potentially unknown variables, i.e., the TB size, and the signal quality, among others (Foukas et al. 2021). Thus, we face a resource assignment problem where we need to decide between energy-efficient CPUs or high-performing HAs to process incoming signals with uncertain processing times. In this context, an overuse of CPUs to save energy may cause that many TBs are not processed within their deadlines leading to data loss, which has serious implications for the mobile operator. In other words, there is a trade-off between processing constraints (deadlines when processing signals) and energy consumption.

Current Open RAN (O-RAN) systems support third-party applications for resource control at 100 millisecond timescales (Garcia-Saavedra et al. 2021). This timescale brings an additional challenge since the decisions cannot be made per TB but with a coarser time granularity. As shown in Fig. 7, our algorithm makes resource allocation decisions every 100 ms, which are implemented as *rules* that are then applied to each TB in real-time in the computing platform.

We hence formulate this problem as the following constrained contextual bandit. We define s_t as the traffic char-

[2]Similar times are observed when using a GPU NVIDIA A100-SXM4-80GB. Due to the small size of the NNs, there is no noticeable gain in execution time when using a GPU.

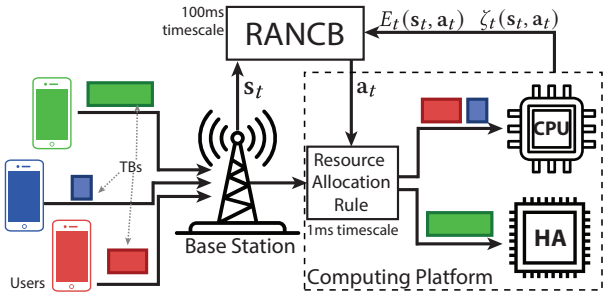

Figure 7: Simplified scheme of the resource assignment problem in mobile networks.

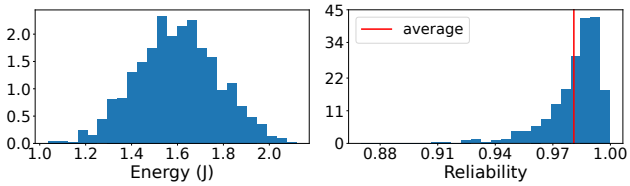

Figure 8: Empirical probability density function of the $E(\cdot)$ and $\zeta(\cdot)$ for a fixed $a = 0.6$ and stationary traffic load.

acterization at time t (context). The offloading decision is denoted by $a_t \in [0, 1]$. The energy consumption of the system (in Joules) for a given context and action is denoted by $E_t(s_t, a_t)$. The ratio of TBs that have been processed within their deadline (reliability) is denoted by $\zeta_t(s_t, a_t) \in [0, 1]$. We formulate the problem as follows:

$$\min_{\{a_t\}_{t=1}^T} \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} E_t(s_t, a_t) \tag{14}$$

$$\text{s.t.} \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \zeta_t(s_t, a_t) \geq 1 - \epsilon.$$

where ϵ sets the target reliability. More details about the formulation are provided in the Appendix. We would like to highlight that both $E_t(\cdot)$ and $\zeta_t(\cdot)$ are very complex functions whose closed-form expressions are unavailable. They characterize the high complexity of the system (i.e., the number of users and their mobility patterns, signal quality, app data generation, etc.) during a period of 100 ms and also depend on the specific hardware of the system. For these reasons, they need to be learned from observations.

Using our experimental platform detailed in the Appendix (Salvat et al. 2023), we characterized experimentally the distribution of the energy ($E(\cdot)$) and the system's reliability ($\zeta(\cdot)$) using a fixed offloading decision $a = 0.6$. Fig. 8 shows that these metrics are random in nature. In particular, if we only considered the average value of the reliability to satisfy the constraint, the reliability would be below its minimum required value with a probability of 0.35 for the distribution in Fig. 8. Our proposal aims at minimizing this risk.

The risk level (i.e., the tolerance to constraint violation) is determined by the specific application scenario. Khan et al. discuss different risk levels (reliability targets) for different

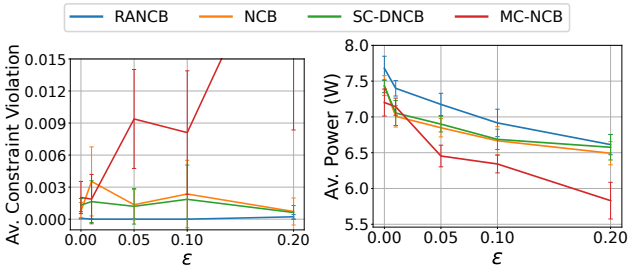

Figure 9: Performance evaluation in our wireless network experimental platform. Average constraint violation per step (left) and average power consumption in watts (right).

scenarios (e.g., broadband communication services in cities vs. industry communication in factories). Moreover, network operators may want to reduce reliability in exchange for lower costs (energy consumption in this example) in some situations. For that reason, we evaluate various risk levels, showing our framework's flexibility to meet different application demands, balancing between risk and cost.

We consider 1500 iterations for training and 500 iterations for inference. Note that SafeOPT cannot be evaluated in this use case because, as shown in Fig 4, its inference time exceeds the system requirements (100 ms). Fig. 9 depicts the mean system unreliability, i.e., $\frac{1}{T}\sum_{t=1}^{T} \zeta_t(s_t, a_t) - (1 - \epsilon)$ (left) and the mean power consumption (right) in inference across all the solutions under study for various reliability targets ϵ. Notably, RANCB consistently outperforms its benchmarks in terms of reliability, providing near-zero unreliability. Importantly, RANCB's superior reliability comes at a low price in terms of power consumption, just 8.5% higher than that of MC-NCB on average.

Related Work

There is a significant body of literature available on contextual bandit algorithms. Some works assume a structure in the reward function, e.g., a linear relationship between the contexts and the reward (Li et al. 2010; Abbasi-Yadkori et al. 2011; Abeille et al. 2017). Other works use neural networks (NN) to learn non-linear reward functions. For example, some use an NN to learn the embeddings of the actions and then apply Thompson Sampling on the last layer of the NN for exploration (Riquelme et al. 2018). Others propose an NN-based algorithm with a UCB exploration (Zhou et al. 2020b). Ban et al. (2022) propose novel exploration strategies based on neural networks. However, none of these works consider constraints in the problem formulation.

Other studies consider contextual bandits with budget constraints (bandits with knapsack) (Badanidiyuru et al. 2014; Agrawal et al. 2014). However, the constraints in both of these papers are cumulative resources (budget) and deterministic, in contrast to the constraints in our problem that are stepwise and stochastic. Others consider the linear contextual bandit problem with safety constraints (Amani et al. 2019; Kazerouni et al. 2017; Daulton et al. 2019). The goal of these works is to obtain at least a percentage of the per-

formance of a baseline policy. Nevertheless, they do not consider the intrinsic random noise of the performance metrics and learn their mean value. Moreover, some of these (Amani et al. 2019; Kazerouni et al. 2017) assume a structure in the reward function (linear), limiting their applicability to environments with non-linearities.

The most closely related work to ours (Berkenkamp et al. 2021) proposes a Bayesian optimization algorithm called SafeOPT that, like in this work, handles constraints and noisy observations. This work generalizes the proposal of Sui et al. by considering multiple constraints and contexts. In contrast to our approach, SafeOPT relies on Gaussian Processes (GPs) that are used to learn the objective and constraint functions. The GPs model the noise and the uncertainty in the estimation, which allows SafeOPT to compute a safe set of actions for each context. Besides, the use of GPs makes the algorithm very data-efficient.

However, SafeOpt has important drawbacks. First, the use of GPs is computationally expensive. Specifically, the complexity scales as $O(n^3)$ with the number of data samples (Williams and Rasmussen 2006). This hinders its application to settings requiring a large amount of data (e.g., environments with high dimensionality) and their deployment in computationally constrained platforms. Second, SafeOPT requires an initial set of actions that satisfy the constraints at the beginning of the training phase. As this set of actions can be highly dependent on the context, its computation can be very challenging, requiring some domain knowledge of the specific application, which limits the applicability of this solution. SafeOPT is objectively evaluated in comparison with our approach in the evaluation section.

Finally, in the Reinforcement Learning (RL) arena, there exist some works that do characterize the distribution of the value/Q function instead of its mean value (Bellemare et al. 2017; Dabney et al. 2018b,a; Nguyen et al. 2020). These approaches bring performance improvements even when only the mean of the distribution is used in the learning process. Moreover, some other works propose a distributional approach to optimize value-at-risk metrics in RL (Tang, Zhang, and Salakhutdinov 2020). However, to the best of our knowledge, these ideas have not been applied yet in the contextual bandit setting nor have they been used to design risk-aware decision-making algorithms in constrained environments.

Conclusions

This paper proposed a risk-aware decision-making framework for constrained contextual bandit problems. The solution relies on an actor-multi-critic architecture, where the multiple critics characterize the distributions of the performance and constraint metrics, and a deterministic actor enables continuous control. Our solution can adapt to different levels of risk to address the trade-off between constraint satisfaction and performance. We evaluated our solution in a synthetic environment and a real-world mobile network testbed, showing its effectiveness.

Acknowledgments

The work was supported by the European Commission through Grants No. SNS-JU-101097083 (BeGREEN), 101139270 (ORIGAMI), and 101017109 (DAEMON). Additionally, it has been partially funded by the Spanish Ministry and the European Union – NextGeneration EU (Call UNICO I+D 5G 2021, ref. number TSI-063000-2021-3) and the CERCA Programme.

References

Abbasi-Yadkori, Y.; et al. 2011. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24.

Abeille, M.; et al. 2017. Linear thompson sampling revisited. In *Artificial Intelligence and Statistics*, 176–184. PMLR.

Agrawal, S.; et al. 2014. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*, 989–1006.

Amani, S.; et al. 2019. Linear stochastic bandits under safety constraints. *Advances in Neural Information Processing Systems*, 32.

Ayala-Romero, J. A.; Garcia-Saavedra, A.; Costa-Perez, X.; and Iosifidis, G. 2021. Orchestrating energy-efficient vrans: Bayesian learning and experimental results. *IEEE Transactions on Mobile Computing*.

Ayala-Romero, J. A.; Garcia-Saavedra, A.; Gramaglia, M.; Costa-Perez, X.; Banchs, A.; and Alcaraz, J. J. 2019. vrAIn: A deep learning approach tailoring computing and radio resources in virtualized RANs. In *The 25th Annual International Conference on Mobile Computing and Networking*, 1–16.

Badanidiyuru, A.; et al. 2014. Resourceful contextual bandits. In *Conference on Learning Theory*, 1109–1134. PMLR.

Ban, Y.; Yan, Y.; Banerjee, A.; and He, J. 2022. EE-Net: Exploitation-Exploration Neural Networks in Contextual Bandits. In *International Conference on Learning Representations*.

Bellemare, M. G.; et al. 2017. A distributional perspective on reinforcement learning. In *International conference on machine learning*, 449–458. PMLR.

Berkenkamp, F.; et al. 2021. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, 1–35.

Brown, N.; and Sandholm, T. 2019. Superhuman AI for multiplayer poker. *Science*, 365(6456): 885–890.

China Mobile Limited. 2021. 2021 Sustainability Report. white Paper.

Dabney, W.; Ostrovski, G.; Silver, D.; and Munos, R. 2018a. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, 1096–1105. PMLR.

Dabney, W.; Rowland, M.; Bellemare, M.; and Munos, R. 2018b. Distributional reinforcement learning with quantile

regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Daulton, S.; Singh, S.; Avadhanula, V.; Dimmery, D.; and Bakshy, E. 2019. Thompson sampling for contextual bandit problems with auxiliary safety constraints. *arXiv preprint arXiv:1911.00638*.

Duvenaud, D. 2014. *Automatic model construction with Gaussian processes*. Ph.D. thesis, University of Cambridge.

Fiducioso, M.; Curi, S.; Schumacher, B.; Gwerder, M.; and Krause, A. 2019. Safe Contextual Bayesian Optimization for Sustainable Room Temperature PID Control Tuning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 5850–5856. International Joint Conferences on Artificial Intelligence Organization.

Foukas, X.; et al. 2021. Concordia: Teaching the 5G vRAN to share compute. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 580–596.

Fujimoto, S.; et al. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, 1587–1596. PMLR.

Galanopoulos, A.; Ayala-Romero, J. A.; Leith, D. J.; and Iosifidis, G. 2021. AutoML for video analytics with edge computing. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, 1–10. IEEE.

Garcia-Saavedra, A.; et al. 2021. O-RAN: Disrupting the Virtualized RAN Ecosystem. *IEEE Communications Standards Magazine*, 5(4): 96–103.

GSMA Association. 2020. 5G energy efficiencies: green is the new black. white Paper.

Huber, P. J. 1992. Robust estimation of a location parameter. *Breakthroughs in statistics: Methodology and distribution*, 492–518.

Kazerouni, A.; Ghavamzadeh, M.; Abbasi Yadkori, Y.; and Van Roy, B. 2017. Conservative contextual linear bandits. *Advances in Neural Information Processing Systems*, 30.

Khan, B. S.; Jangsher, S.; Ahmed, A.; and Al-Dweik, A. 2022. URLLC and eMBB in 5G industrial IoT: A survey. *IEEE Open Journal of the Communications Society*, 3: 1134–1163.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Meta Fundamental AI Research Diplomacy Team (FAIR); Bakhtin, A.; Brown, N.; Dinan, E.; Farina, G.; Flaherty, C.; Fried, D.; Goff, A.; Gray, J.; Hu, H.; et al. 2022. Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science*, 378(6624): 1067–1074.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Nguyen, T. T.; et al. 2020. Distributional reinforcement learning with maximum mean discrepancy. *Association for the Advancement of Artificial Intelligence (AAAI)*.

Riquelme, C.; et al. 2018. Deep Bayesian Bandits Showdown: An Empirical Comparison of Bayesian Deep Networks for Thompson Sampling. In *International Conference on Learning Representations*.

Salvat, J. X.; Ayala-Romero, J. A.; Zanzi, L.; Garcia-Saavedra, A.; and Costa-Perez, X. 2023. Open Radio Access Networks (O-RAN) Experimentation Platform: Design and Datasets. *IEEE Communications Magazine*.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In *International conference on machine learning*, 387–395. Pmlr.

Solozabal, R.; et al. 2020. Constrained combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:2006.11984*.

Sui, Y.; Gotovos, A.; Burdick, J.; and Krause, A. 2015. Safe exploration for optimization with Gaussian processes. In *International conference on machine learning*, 997–1005. PMLR.

Tang, Y. C.; Zhang, J.; and Salakhutdinov, R. 2020. Worst Cases Policy Gradients. In *Conference on Robot Learning*, 1078–1093. PMLR.

Tessler, C.; et al. 2018. Reward Constrained Policy Optimization. In *International Conference on Learning Representations*.

Williams, C. K.; and Rasmussen, C. E. 2006. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Xu, P.; Wen, Z.; Zhao, H.; and Gu, Q. 2022. Neural Contextual Bandits with Deep Representation and Shallow Exploration. In *International Conference on Learning Representations*.

Zhou, D.; et al. 2020a. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, 11492–11502. PMLR.

Zhou, D.; et al. 2020b. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, 11492–11502. PMLR.