ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230320

Reinforcement Learning for Bandits with Continuous Actions and Large Context Spaces

Paul Duckworth[a;*], **Katherine A. Vallis**[b], **Bruno Lacerda**[a] **and Nick Hawes**[a]

[a]Oxford Robotics Institute, Department of Engineering Science, University of Oxford
[b]Department of Oncology, University of Oxford
ORCiD ID: Paul Duckworth https://orcid.org/https://orcid.org/0000-0001-9052-6919, Katherine A. Vallis https://orcid.org/https://orcid.org/0000-0003-4672-5683, Bruno Lacerda https://orcid.org/https://orcid.org/0000-0003-0862-331X, Nick Hawes https://orcid.org/https://orcid.org/0000-0002-7556-6098

Abstract. We consider the challenging scenario of contextual bandits with continuous actions and large context spaces. This is an increasingly important application area in personalised healthcare where an agent is requested to make dosing decisions based on a patient's single image scan. In this paper, we first adapt a reinforcement learning (RL) algorithm for continuous control to outperform contextual bandit algorithms specifically hand-crafted for continuous action spaces. We empirically demonstrate this on a suite of standard benchmark datasets for vector contexts. Secondly, we demonstrate that our RL agent can generalise problems with continuous actions to large context spaces, providing results that outperform previous methods on image contexts. Thirdly, we introduce a new contextual bandits test domain with multi-dimensional continuous action space and image contexts which existing tree-based methods cannot handle. We provide initial results with our RL agent.

1 Introduction

We consider the challenging scenario of contextual bandits with continuous actions and large "context" spaces, for example 2D or 3D images. This setting arises naturally when an agent is repeatedly requested to provide a single continuous action based on observing a context only once. In this scenario, an agent acquires a context, chooses an action from a continuous action space, and receives an immediate reward based on an unknown loss function. The process is then repeated with a new context vector. The agent's goal is to learn how to act optimally. This is usually achieved via minimising regret across actions selected over a fixed number of trials.

Our work is motivated by an increasingly important application area in personalised healthcare: An agent is requested to make dosing decisions based on a patient's single image scan (additional scans after treatment can potentially be damaging to the patient) [16]. This is an unsolved domain; current bandit methods fail to handle the large context space associated with medical scans and the high-dimensional actions required to control dosing decisions.

We consider this problem under the *contextual bandits* framework. Contextual bandits are a model for single-step decision making under uncertainty where both exploration and exploitation are required in unknown environments. They pose challenges beyond classical supervised learning, since a ground truth label is not provided with each training sample. This scenario can also be considered as one-step reinforcement learning (RL), where no transition function is available and environment interactions are considered independent and identically distributed (i.i.d).

There are well-established methods for contextual bandit problems with small, discrete action spaces, often known as *multi-armed bandits with side information*. The optimal trade-off between exploration and exploitation is well studied in this setting and formal bounds on regret have been established [1, 4, 13, 14, 26]. However, there is relatively little research into bandits for continuous action spaces, that frequently arise in real world scenarios. Recent works have focused on *extreme classification* using tree-based methods to sample actions from a discretized action space with smoothing [22, 28]. However, developments in RL for continuous control beg the question: *Can we use single-step policy gradients to solve contextual bandits with continuous actions?*

Real world contextual bandit problems, such as those in healthcare, require solution methods to generalise to large context spaces, i.e. directly from images. We posit that existing tree-based methods are not well suited to this task in their current form [5, 22, 28]. However, recent breakthroughs in deep RL for sequential decision making [3, 15, 27, 31] have successfully demonstrated continuous control abilities directly from pixels. In this regard, neural networks have proven powerful and flexible function approximators, often employed as parametric policy and value networks directly from image inputs. The adoption of nonlinear function approximators often reduces any theoretical guarantees of convergence, but works well in practice. Underpinning this recent work in deep RL for continuous control from pixels is the deterministic policy gradient that can be estimated much more efficiently than the usual stochastic policy gradient [27, 33].

The contributions of our work are as follows:

1. We modify an existing RL algorithm for continuous control and demonstrate state-of-the-art results on contextual bandit problems with continuous actions. We benchmark our approach on four

* Corresponding Author. Email: pduckworth@robots.ox.ac.uk

Please check ArXiv or contact the authors for any appendices or supplementary material mentioned in the paper.

standard OpenML datasets across two settings: online average regret and offline held-out costs.

2. We propose a deep contextual bandit agent that can handle image-based contexts and continuous actions. To the best of our knowledge, we are the first to tackle the challenging setting of multi-dimensional continuous actions and high-dimensional context space. We demonstrate state-of-the-art performance on image contexts.

3. Finally, we propose a new challenging contextual bandits domain for multi-dimensional continuous actions and image contexts. We provide this challenging domain as a new testbed to the community, and present initial results with our RL agent.

Our new contextual bandit test domain is based on a popular 2D game of Tanks: where two opposing tanks are situated at opposite sides of the game screen. The agent (right hand side) must learn to accurately fire trajectories at the enemy tank (left hand side). To do so, the agent has agency over three continuous action dimensions: the agent x-location, its turret angle and its shot power. Three example Tanks domain context images are provided in Figure 2.

2 Related Works

A naive approach to dealing with continuous actions is to simply discretise the action space [34]. A major limitation of this approach is the curse of dimensionality: the number of possible actions increases exponentially with the number of action dimensions. This is exacerbated for tasks that require fine control of actions, as they require a correspondingly finer grained discretisation, leading to an explosion of the number of discrete actions. A simple fixed discretization of actions over the continuous space has been shown to be wasteful, and has led to adaptive discretization methods, e.g. the Zooming Algorithm [19, 20].

Building upon the discretisation approach, extreme classification algorithms were recently developed [22, 28]. These works use tree-based policies and introduce the idea of smoothing over actions in order to create a probability density function over the entire action space. The authors provide a computationally-tractable algorithm for large-scale experiments with continuous actions. However, their optimal performance guarantees scale inversely with the bandwidth parameter (the uniform smooth region around the discretised action). The authors of these works provide no theoretical or empirical analysis investigating the effects of large context sizes. For these reasons, we propose an RL agent based on a single-step policy gradient [36] that scales well with context size to handle continuous actions and large context spaces.

Prior works have framed personalised healthcare problems as contextual bandits problems [17, 30]. A single-step actor-critic method was proposed for binary actions relating to just-in-time adaptive interventions [25]. However, until now, these methods have been restricted to a small number of discrete actions and small context vectors.

In contrast, the notion of deep contextual bandits has previously been introduced in [40] to handle large context spaces. The authors propose a novel sample average uncertainty method, however, it is only suitable for discrete action spaces. To the best of our knowledge, no prior works have focused on the challenging intersection of continuous actions and large context spaces, and it is an under-explored research area of particular interest in healthcare.

3 Preliminaries

Setting and key definitions:

We consider the i.i.d contextual bandit setting with continuous actions. In this setting, an agent acquires a context vector (or image) x_t from the context space $\mathcal{X}$ by observing the environment E at timestep t. The agent chooses an action a_t from a continuous action space $\mathcal{A} = [0, 1]^N$, and receives an immediate cost based on an unknown loss function $l_t : \mathcal{X} \times \mathcal{A} \to [0, 1]$. The process is then repeated with a new context at time $t + 1$. Unlike the standard RL setting, there is no transition function in the bandit setting.

Existing contextual bandit literature for continuous actions make the restriction that $N = 1$, i.e. a one-dimensional action space is used. This constraint is relaxed for our proposed RL agent in Section 5.2, however for notational simplicity, we restrict ourselves to the $N = 1$ case here.

In general, an agent's behaviour is defined by a policy. We define a stochastic policy π_θ that maps from the context space $\mathcal{X} \to \mathcal{P}(\mathcal{A})$, where $\mathcal{P}(\mathcal{A})$ is the set of probability measures on $\mathcal{A}$.

We define a vector of n parameters as $\theta \in \mathbb{R}^n$, and $\pi_\theta(a_t \mid x_t)$ as the conditional probability density associated with the policy parameterised by θ. We also define a deterministic policy μ_θ that maps contexts $x_t \in \mathcal{X}$ to specific actions in $\mathcal{A}$, similarly parameterised by a vector θ.

For tree-based policy methods [22], a *smoothing operator* is required: $\texttt{Smooth}_h : \mathcal{A} \to \mathcal{P}(\mathcal{A})$, maps each action a to a Uniform distribution over the interval $\{a' \in \mathcal{A} : |a - a'| \leq h\} = [a - h, a + h] \cap \mathcal{A}$.

Regret: Online vs Offline:

Over T rounds, an agent accumulates a history of interactions. For example, after t rounds, this experience is comprised of tuples $\{(x_i, a_i, l_t(x_i, a_i))\}_{i=1}^t$. Note that unlike in supervised learning, we do not have access to the true label or optimal action.

One assumption in the contextual bandit setting is the availability of an oracle to provide the loss value for a given context and action pair. Traditionally, an agent's goal is to minimise the *expected cumulative regret* across a fixed number of T trials, which is equivalent to minimising the online loss: $J \triangleq \sum_{i=1}^T \mathbb{E}[l_t(x_i, a_i)]$. Note that this expectation is under both the policy and a potentially stochastic loss function, i.e. not a transition function.[1] In this online setting, an agent has access to each trial only once and no prior knowledge of the environment is assumed, and no burn-in phases is allowed.

Whilst online cumulative or average regret over a fixed number of trials is a popular setting, it is also common in machine learning to allow a dedicated learning or planning phase followed by an execution phase to exploit the learning. In these cases, *simple regret* [10] can also be a sensible consideration. In this offline setting a dataset of experience is collected in advance and an agent relies on multiple passes (or epochs) over the dataset of experience followed by an evaluation on a held-out, unseen portion. We know that for enough training samples offline validation will obey the Hoeffding

[1] We provide an example of a stochastic loss function, induced by stochastic action outcomes, in our new Tanks domain in Section 5.2.

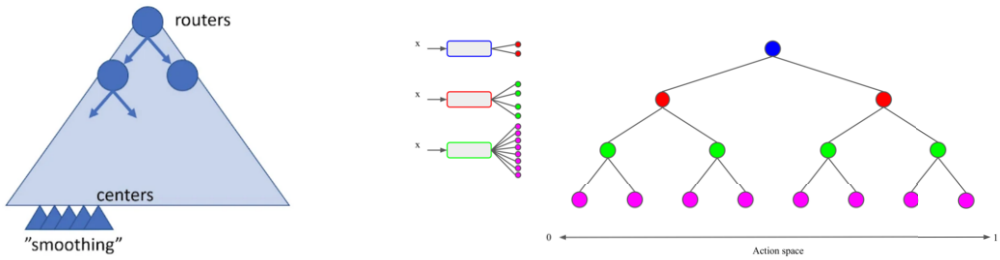

Figure 1. (Left:) CATS binary tree policy where each internal node routes a context vector left or right until a leaf node that represents an action center. Smoothing is then applied to create a probability distribution over continuous action space $\mathcal{A}$, and an action is sampled. Image taken from [38]. (Right): CATX tree policy: re-implementation with multi-class classification networks at each tree-layer. Action discretization of 8 (pink leaves), tree depth of three with blue, red and green classification neural networks, for input context x. Image taken from [5]. Best viewed in colour.

bound [7], guaranteeing that with high probability the error discrepancy between the training estimate and hold-out estimate will be small. Therefore, in the offline setting, we report average costs (or loss) on a held-out portion of the dataset.

Methods such as UCB1 are often used to minimise online cumulative regret in discrete action settings [4, 9], or to identify the best action in a best-arm identification setting. However, we cannot maintain statistics over the infinite actions available in the continuous setting.

Continuous Armed Bandits:

Continuous armed bandits are an extension to the multi-armed bandit (MAB) setting that have an infinite number of action values available in a continuous range. Tractable algorithms, with provable bounds on cumulative regret, often make use of a Lipschitz assumption[2] [2, 18]. However, a simple fixed discretization of arms over the continuous space has been shown to be wasteful, and adaptive discretization has become a popular approach, e.g. the Zooming Algorithm [19, 20]. Adaptive discretization has the same worst-case regret as fixed discretization, but often performs better in practice. However, the regret bounds for these algorithms rely on prior knowledge of the Lipschitz constant.

A recent state-of-the-art method proposed smoothing over a discretised action space in order to create a density over the entire continuous action space [22, 28]. Continuous Actions + Trees + Smoothing (CATS) is a recent algorithm that uses a pre-specified bandwidth and number of discrete action bins to perform the task of *extreme classification* for the continuous action setting [28]. We demonstrate in Section 5.2 that it does not scale well to image contexts. We also evaluate an open-source re-implementation of the CATS algorithm (CATX) [5]. CATX additionally shares parameters within its tree-layers using multi-class classification neural networks instead of binary trees (as the original implementation). We compare our RL agent to both the CATS and CATX algorithms in Section 5.3, and describe each in more detail next.

Continuous Action + Trees + Smoothing (CATS):

The CATS algorithm [28] was recently introduced as state-of-the-art for the contextual bandits with continuous action spaces. It is based on the idea of extreme classification using a decision tree structure where internal nodes route a context vector to a probability distribution over continuous actions. That is, the algorithm performs successive classifications within a binary tree structure, where the number of nodes in each subsequent layer increases exponentially. The authors define a Tree policy $\mathcal{T}$ as a complete binary tree of depth D with $K = 2^D$ leaves, where each leaf v has a label function $\texttt{label}(v) = \frac{0}{K}, \ldots, \frac{K-1}{K}$ from left to right respectively, and where each internal node v maintains a binary classifier $f^v \in \mathcal{F}$ the class of binary classifiers from $\mathcal{X} \rightarrow \{\texttt{left, right}\}$.

Successive binary classifications route an input context to an output bin, or action "center". A CATS tree policy is depicted in Figure 1 (left). An action is then sampled ϵ-greedily from a uniformly smooth region around the bin (defined by a bandwidth parameter). The key idea is to smooth the actions: each action a is mapped to a distribution over actions using a smoothing operator introduced in Section 3. When the action space is the interval $[0, 1]$, $\mathcal{P}(\mathcal{A}) = [a - h, a + h] \cap [0, 1]$. The authors state that the loss function for smoothed actions is always well-behaved [22]. We point readers to [28] for more details.

CATX:

A JAX re-implementation [5] builds and improves upon the original CATS algorithm and is released open-source. The authors make two key contributions: i) they integrate nonlinear function approximators into the tree structure. Instead of binary classifiers at each internal node they implement multi-class classification networks at each tree layer; ii) each layer is represented by a network with output shape equivalent to the nodes in that tree-layer, meaning that each layer *shares parameters*. This is intended to stabilise training.

Figure 1 (right) depicts a CATX tree policy of depth three comprising of three multi-class classification neural networks (blue, red and green), with increasing number of action-dimentions providing a final action discretisation of $2^3 = 8$ (bins). Similarly to CATS, an action is sampled ϵ-greedily from the smoothed $\mathcal{P}(\mathcal{A})$, i.e. a smoothed region around the leaf node.

Deterministic Policy Gradients (DPG):

RL for continuous control has its origins in the Deep Q-Network (DQN) [29] algorithm that combines advances in deep learning with reinforcement learning (RL). This seminal work adapted the standard Q-learning algorithm in order to make effective use of a neural network as a flexible function approximator. However, while DQN tackles problems with high-dimensional observation spaces, it can only handle discrete and low-dimensional action spaces. For learning in high-dimensional and continuous action spaces,

[2] A function $f : X \rightarrow \mathbb{R}$ which satisfies $|f(x) - f(y)| \leq L.|x - y|$ for any two arms $x, y \in X$, is called Lipschitz-continuous on X, with Lipschitz constant L.

previous work combines an off-policy actor-critic approach with a deterministic policy gradient (DPG) [33]. DPG maintains an actor function $\mu(s|\theta^\mu)$, represented as a neural network and parameterised by θ^μ, which specifies the current policy by deterministically mapping states to specific actions. The critic function $Q(s,a)$ is also learned as a neural network using the Bellman equation. However, since the action space is continuous, the critic is presumed to be differentiable with respect to the action argument. This allows for a policy gradient-based learning rule, and we can approximate the best action-value to take $\max_a Q(s,a)$ with $Q(s,\mu(s))$.

The standard REINFORCE, or vanilla actor-critic algorithms are restricted to on-policy updates [35, 39]. This means that training samples are collected according to the policy currently being optimised for. These samples are then discarded and cannot be reused. However, DPG is an off-policy actor-critic approach [11] that does not require full trajectories to learn and can reuse past interactions by storing them in an experience replay buffer. This facilitates better sample efficiency by allowing a behaviour policy to collect new training samples to be decoupled from the deterministic policy being learned, providing better exploration. This allows the behaviour policy to be stochastic for exploration purposes, but the target policy remains deterministic.

Finally, combining DPG with insights from the success of DQN, the deep deterministic policy gradients (DDPG) [27] algorithm demonstrates solving continuous simulated physics tasks directly from pixels. To achieve stability, the authors borrow ideas from DQN, such as a replay buffer, soft update target networks, and added a novel noise Ornstein-Uhlenbeck process to generate temporally-correlated exploration noise.

We propose a deep contextual bandit agent for continuous actions based upon the DDPG algorithm. We describe our notation and the modifications to the standard DDPG algorithm in the next section.

4 Methodology

We define the contextual bandit setting as a single-step reinforcement learning problem that can be specified by a Markov decision process (MDP) [21, 35] with state space $\mathcal{S}$ and reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. We consider the state space equivalent to the contextual bandit context space $\mathcal{X}$ defined in Section 3.

In the RL setting, the value of a state can be defined as the expected total discounted future reward from that state onwards. The return is defined to be the total discounted reward from time-step τ onwards, $R^\gamma_\tau = \sum_{k=\tau}^{\infty} \gamma^{k-\tau} r(s_k, a_k)$, for discount factor $0 < \gamma < 1$. The agent's goal is to maximise the immediate reward plus the estimated return. Notice that τ refers to the timestep within a trajectory evolving under a policy and transition function up to some infinite horizon. Whereas, t specifies the trial from T total trials in a contextual bandit setting, i.e. the agent is provided with i.i.d context vectors from the environment, i.e. $x_t \sim E$. Given the absence of a transition function in the bandit setting one major insight is that we do not require an estimate for the return R^γ_τ, as the value of any future states collapses to the instant reward achieved at the first time point. We can therefore consider this as a single-step RL problem. Instead of maximising rewards, in the contextual bandit setting it is common to minimise the equivalent loss $l_t(x_t, a_t)$.

We define the optimal single-step action-value function based on the Bellman equation as: $Q^*(x_t, a_t) = \mathbb{E}\left[r(x_t, a_t)\right]$, and the mean-squared Bellman error (MSBE) function collapses to: $L(\phi, E) = \mathop{\mathbb{E}}_{(x_t, a_t, r_t)\sim E}\left[\left(Q_\phi(x_t, a_t) - r_t\right)^2\right]$, for network parameters ϕ and environment E [35]. Note we can interchange the rewards r with losses depending on the environment.

One can note that we no longer require an estimate of the action-value for the future states along our trajectory. A key innovation in DQN [29] was to introduce a target Q network to decouple the Q-value being optimised with the Q-value in the temporal difference update for stability. However, since the contextual bandit setting has no transition function, there is no need to estimate future action-values. This means that our computation requirements are reduced as we do not need to maintain a duplicate Q network, or interleave learning with target network updates.

It is particularly useful to estimate the policy gradient off-policy from actions sampled using a distinct behaviour (or exploration) policy, that is $\beta(a_t \mid x_t) \neq \pi(a_t \mid x_t)$. We use an off-policy actor-critic algorithm [11], coupled with a deterministic policy network and deterministic policy gradients [33]. We estimate the action-value function using a differentiable function approximator, and then update a deterministic parametric policy μ_θ in the direction of the approximate action-value gradient.

The fundamental result underlying our approach is the policy gradient theorem [36]. We are specifically interested in the deterministic policy gradient [33], and we adapt it here for the contextual bandit setting:

$$\begin{aligned}\nabla_\theta J_\beta(\mu_\theta) &= \int_{\mathcal{X}} \rho^\beta(x) \nabla_\theta \mu_\theta(a \mid x) Q^\mu(x, a) dx, \\ &= \mathbb{E}_{x_t \sim E}[\nabla_\theta \mu_\theta(x_t) \nabla_a Q^\mu(x_t, a_t) \mid_{a_t = \mu_\theta(x_t)}],\end{aligned} \tag{1}$$

where, $\rho^\beta(x)$ defines the state distribution visited under our behaviour policy β, which in the bandit setting is equivalent to sampling contexts x_t from our environment E at each timestep t.

There is a crucial difference between the stochastic policy gradient and the deterministic policy gradient: In the stochastic case, the policy gradient integrates over both state and action spaces, whereas in the deterministic case, it only integrates over the state space. This has additional benefits for continuous or high-dimensional action spaces.

For exploration, we take inspiration from the literature [27]. We construct a behaviour policy β by adding noise sampled from a noise process $\mathcal{N}$ to our deterministic actor policy:

$$\beta = \mu(x_t \mid \theta^\mu_t) + \mathcal{N}, \tag{2}$$

where $\mathcal{N}$ is chosen simply as a single-step Ornstein-Uhlenbeck process to aid exploration.

5 Experiments

5.1 Stage 1: Continuous actions and vector contexts

Following the experimental protocol provided in [28] and [6], we first evaluate the performance of our RL agent on four benchmark OpenML datasets [37]. We use four widely-used benchmark datasets from OpenML. They are open-source, publicly available, and version controlled. We do not manipulate the datasets, except for downloading, normalising, and randomly splitting into train and test sets.

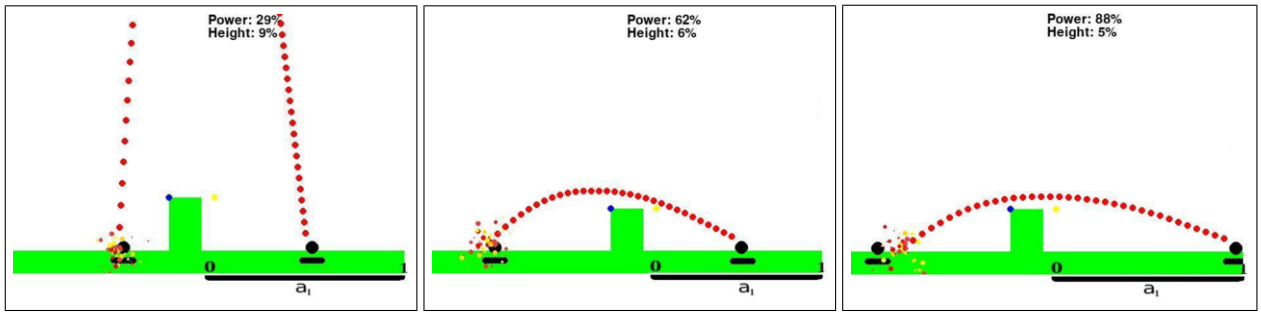

Figure 2. Three example images from our new Tanks Bandit domain. Demonstrated here are three successful actions. Tanks in black (enemy is left, agent is right of barrier), trajectory of action taken shown in red. 3-dimensional continuous actions are interpreted as a_1 : x-location, a_2 : shot power, a_3 : turret height.

Significant efforts have been made to ensure the reproducibility of this work, and that the community can build upon our research. We provide all our hyperparameters in the Appendix.

Each dataset has a single continuous action dimension and vector context space. The four benchmark datasets are:

1. Wisconsin dataset: 32 dim state vector, 1 action dim, 194 samples.
2. CPU_act dataset: 21 dim state vector, 1 action dim, 8,192 samples.
3. Zurich Delays 5% dataset: 17 dim state vector, 1 action dim, 26,670 samples.
4. Black friday dataset : 9 dim state vector, 1 action dim, 166,821 samples.

We demonstrate the performance of our RL agent compared to state-of-the-art tree-based policy method CATS [28], and the shared parameter multi-class classification re-implementation CATX [5] as described in Section 3.

We used an implementation of CATS released with publication [28]. It is developed in the open-source Vowpal Wabbit framework [23]. The code performs a parameter sweep over discretization and smoothing parameters: $\mathcal{J} = \{ (h, K) : h \in \{2^{-13}, \dots 2^{-1}\}, K \in \{2^2, \dots 2^{13}\}\}$, and we report the best results. The hyperparamer settings for all experiments presented can be found in Appendix **??**. Similarly, we used the open-source JAX implementation of CATX code [5], and also swept for optimal hyperparameters per dataset.

Following the literature, we also compare to two baseline approaches: Firstly, $dLinear$, a discretized $\epsilon-$greedy algorithm which by default uses a doubly robust approach for policy evaluation and optimisation [32]. Secondly, $dTree$, is a discretized tree-based algorithm which is equivalent to CATS without smoothing, i.e. zero bandwidth. For all experiments we used $\epsilon = 0.05$.

5.2 Stage 2: Continuous actions and image contexts

In this section, we significantly increase the challenge by providing image based contexts. That is, we move away from small vector context spaces, to high-dimensional contexts directly from pixels. We evaluate performance on two image datasets and describe each below:

1. a single-action dimension task based on the widely used MNIST dataset [24].
2. a novel Tanks game with multi-dimensional continuous action parameters.

MNIST contains 60K training and 10K held-out samples, where each is a 28×28 image. We define a continuous loss function $l(x_t, a_t) = |a_t - y_t|$, where y_t is the label provided as supervision i.e. the digit value.

Tanks Domain:
We introduce a novel image benchmark domain for multi-dimensional actions based on a 2D game of Tanks. See Figure 2 for three sample game images. For this new domain, we provide a convenient OpenAI Gym [8] python interface to an existing popular source game [12].

The Tanks domain is a new benchmark that we will release with the paper. It has two challenging properties over existing contextual bandit benchmark datasets:

1. large image context space (800×600 pixels)
2. three-dimensional continuous actions, $\mathcal{A} = [0, 1]^N$, where $N = 3$.

The domain is flexible, such that the context x_t can be provided as either a low-dimensional vector describing the locations of key objects in the scene, or directly as an image. In Figure 2 we show three example game images that contain the ground, a randomly placed barrier of random height, and an enemy tank with random location to the left hand side of the barrier. A multi-dimensional continuous action vector is required from the agent in order to shoot: a_1 : the x-location of its own tank (restricted to the right hand side of the barrier); a_2 : the power of its shot; and a_3 : the height of its turret.[3] The resulting trajectories from three example actions can be see overlaid onto Figure 2. Note that the agent does not get to observe the trajectory as shown here. In Appendix Figure **??**, we show the only the context image x_t available to the agent.

We intend for the Tanks domain to become a useful benchmark in this challenging domain. As such, we provide two different loss functions, one a smooth loss function and the second is sparse around the enemy tank.

The smooth loss function is defined across the enemy's potential location, i.e. the left hand side of the barrier. We first define x_{i}, as the x-location where the fired trajectory intersects with the ground plane; x_{e}, as the x-location of the enemy tank; and x_{b} as the x-location of the random barrier. Then the loss function for context x_t and action

[3] In Figure 2 the Power action a_2 is scaled $[0, 1] \rightarrow [0, 100]$ and the turret height a_3 $[0, 1] \rightarrow [0, 9]$ for visual purposes only.

a_t is defined as:

$$l_t(x,a) = \begin{cases} |x_{\mathtt{i}} - x_{\mathtt{e}}|/x_{\mathtt{b}} & \text{if } (0-\delta) \leq x_{\mathtt{i}} \leq (x_{\mathtt{b}}) \\ 1 & \text{otherwise,} \end{cases} \quad (3)$$

where δ is a free parameter and set to 50 pixels by default.

The sparse loss is defined only around the enemy's actual location:

$$l_t(x,a) = \begin{cases} 0 & \text{if } |x_{\mathtt{i}} - x_{\mathtt{e}}| \leq \delta' \\ 1 & \text{otherwise,} \end{cases} \quad (4)$$

where δ' is a free parameter and set to 30 pixels by default.

As described in Section 3 we evaluate both online regret and offline held-out cost over an unseen validation set. Each agent was trained for 50 epochs and 50 batches per epoch (further experimental details are provided in Appendix **??**).

5.3 Results

To evaluate our agents we provide metrics based upon the two settings introduced in Section 3: online regret minimisation, and offline held-out costs. Online regret is popular in the Bandits literature, where the agent has access to each training sample only once. Offline held-out costs is popular in the RL literature, and facilitates multiple passes over the dataset of experience during a training phase. We simulate this setting by passing over the training dataset for multiple epochs, and report the average cost on a held-out validation set (10% of the dataset size unless otherwise stated).

Online Regret:

We report online regret on four benchmark OpenML datasets with vector contexts in Figure 3 (left) (where lower is better). We provide the dataset size reported along the x-axis (increasing from left to right). In the online setting this is also the number of fixed trials T the agent observes, i.e. each instance only once.

We can extract clear trends from the results: as the number of trials the agent observes increases, i.e. for larger datasets, our RL agent (yellow) begins to noticeably outperform the tree-based methods in average online regret. For example, Wisconsin is a very small dataset, and provides only 194 trials to the agent. In this case, the tree-based method CATS achieves the lowest online regret. However, as the number of trials increases, for example in the Zurich and Friday datasets, our RL agent achieves lowest regret.

This result is expected since our RL agent's policy and value neural networks require a certain number of policy gradient updates in order to converge, after initialising with random weights. This experiment demonstrates that our RL agent can outperform the baseline agents when provided with enough examples.

Offline Costs:

We report the average offline costs on the four OpenML datasets in Figure 3 (right) (lower is better). We hold out 10% of the available dataset and report average costs on the held-out samples.

In the offline setting the agent has multiple passes over the training samples. In Figure 3 (right) we shaded the additional improvement in performance by re-using the training data and iterating from epoch 0 (bold) and epoch 10 (shaded) (30 for Wisconsin dataset). We depicted the methods that re-used training data using a "+" in the legend.

The results demonstrate a clear trend, that on held-out samples our RL agent outperforms the tree-based methods on all datasets. The shared-parameter implementation CATX also outperforms CATS. This experiment demonstrates that our RL agent achieves the best performance on unseen samples, outperforming standard contextual bandit methods for continuous actions.

Image Context:

The substantially more challenging image context datasets are described in Section 5.2. We report the held-out costs on the MNIST dataset in Figure 3 (far right) (where lower is better).

We can clearly see our RL agent achieves the lowest average cost, and continues to improve with more epochs of training (the shaded agents had access to 10 epochs during the training phase). Our RL agent is able to learn the required representations of high-dimensional contexts in order to provide accurate continuous actions without discretisation unlike the baseline methods. Further, the shared-parameter CATX implementation outperforms the original CATS algorithm that is not able to adequately route image contexts to continuous actions using its learned binary tree policies.

Multi-dimension Actions and Image Context:

In this final section we provide our initial results applying our RL agent to the novel, multi-dimensional continuous action Tanks domain. Neither of our baseline methods based on tree-policies (CATS and CATX) can handle more than a single action dimension. We extend the RL formulation so that the policy neural-network outputs a 3-dimensional continuous value for an action. All images were rescaled to (32×32) greyscale images for this experiment.

In Figure 4 we present the training curves (lower is better) for the online regret setting (left) and held-out costs (right) for four different RL agents:

1. low-dimensional vector context (blue);
2. image context with smooth loss function in Equation 3 (orange);
3. image context with sparse loss function in Equation 4 (green);
4. random agent (red).

Figure 4 shows that the using the smooth loss function, our RL agent is able to learn in both the online, and offline setting. We provide the sparse loss function as a challenge to the community.

6 Discussion

In this paper we have adapted an RL agent for contextual bandit problems with continuous action spaces based on the deterministic policy gradients algorithm. We demonstrated it outperforms existing tree-based policy methods specifically designed for this task. Our approach is effective from vector context, as is standard in the literature, and also from image contexts where existing methods fail to handle the high dimensionality.

Traditionally, the design of contextual bandit algorithms has been steered towards minimising cumulative (or average) regret over a fixed number of T trials, with no burn-in phase, or sample re-use. However in challenging real world scenarios, like those faced in personalised healthcare, we propose designing agents that minimise either simple regret or held-out costs *after* a training period. In this way, flexible RL agents can be trained using policy gradients as in our proposed method that re-use previous experience multiple times. We have successfully demonstrated that this is possible from 2D image contexts, and it is clear that as a community we must solve these problems at a larger scale to address challenges in real world

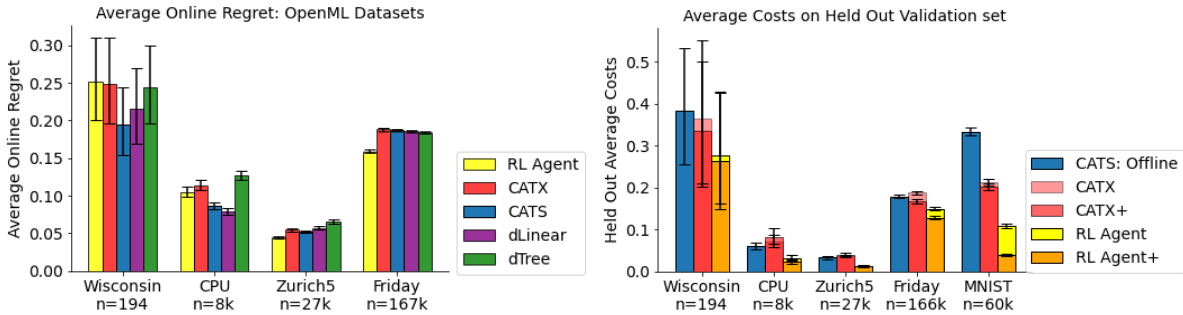

Figure 3. (Left:) Average regret in the online setting on four benchmark OpenML datasets each with a single continuous action parameter. Dataset size and number of fixed trials T is provided along the x-axis. (Right:) Average cost on a held-out validation set (10% unless stated). Agents using multiple epochs during training are depicted with "+" symbol. All confidence intervals are calculated with a single run using the Clopper-Pearson interval with 95% confidence level (as is standard in the literature). The confidence intervals are noticeably larger for the small dataset sizes: Wisconsin and CPU datasets.

personalised healthcare, for example from 3D image scans.

Finally, existing works based on tree policies are limited to a single action dimension. This is a major limitation to their approach. In our new Tanks domain, we successfully demonstrated control of a 3-dimensional continuous action value from image context. Our neural network policy can be trivially extended to output more than a single continuous action value and share parameters within the network architecture, unlike existing bandit methods.

Whilst our paper introduces a new, open-source challenging Tanks domain; Games have long provided a testbed for research into intelligent agents, from DeepBlue to AlphaGo. It is worth noting that although tanks are depicted in this game, the agents intentionally do not resemble real world tanks, and the game does not encourage violent actions. This simplified video game provides a contextual bandit Environment where an agent faces a continuous control problem: learn to predict a scoring trajectory by adjusting the parameters of the tank, i.e. the actions, based on a single image of the environment, i.e. the context.

This paper is under double-blind review. For this reason, we honour confidentiality and have provided no links to Github repositories or code.

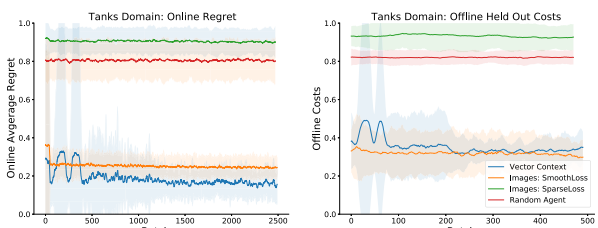

Figure 4. (Left:) Average regret in the online setting for Tanks domain. (Right:) Average cost on a held-out validation set. Four RL agents were evaluated: (blue:) using low-dimensional vector context; (orange:) image context with smooth loss function; (green:) image context with sparse loss function; (red:) random agent. Confidence intervals (shaded) are generated by 5 training runs and each agent was trained for 50 epochs, with 50 batches per epoch. Best viewed in colour.

7 Conclusion

In this paper we have demonstrated state-of-the-art performance on contextual bandit problems with continuous actions by modifying an RL algorithm for continuous control. We outperform hand-crafted contextual bandit algorithms for continuous actions in both online regret and held-out costs. Furthermore, we have successfully demonstrated generalising contextual bandits with multi-dimensional continuous actions to large context spaces, such as images. We provided state-of-the-art performance using RL, significantly outperforming tree-policy methods on multiple standard benchmark datasets.

Finally, we have introduced a new benchmark domain for contextual bandits with multi-dimensional continuous actions and image contexts. We provide this challenging domain to the wider research community in the hope of stimulating additional research towards solving these challenging domains, especially when using sparse loss signals.

Whilst the motivation for this work comes from personalised healthcare, we leave the application of our method to medical datasets for future work. Similarly, for even larger context spaces, for example 3D image scans, with few datapoints we expect that integrating unsupervised image representations with the RL agent will be a promising avenue for future work.

8 Acknowledgements

The authors gratefully acknowledge funding support from the EPSRC Programme Grant "From Sensing to Collaboration" (EP/V000748/1) and CRUK Radnet Oxford Centre (grant number A28736).

References

[1] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire, 'Taming the monster: A fast and simple algorithm for contextual bandits', in *International Conference on Machine Learning*, pp. 1638–1646. PMLR, (2014).

[2] Rajeev Agrawal, 'The continuum-armed bandit problem', *SIAM journal on control and optimization*, **33**(6), 1926–1951, (1995).

[3] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath, 'A brief survey of deep reinforcement learning', *arXiv preprint arXiv:1708.05866*, (2017).

[4] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer, 'Finite-time analysis of the multiarmed bandit problem', *Machine learning*, **47**(2), 235–256, (2002).

[5] Wissam Bejjani and Cyprien Courtot. Catx: contextual bandits library for continuous action trees with smoothing in jax, 2022.

[6] Alberto Bietti, Alekh Agarwal, and John Langford, 'A contextual bandit bake-off.', *J. Mach. Learn. Res.*, **22**, 133–1, (2021).

[7] Avrim Blum, Adam Kalai, and John Langford, 'Beating the hold-out: Bounds for k-fold and progressive cross-validation', in *Proceedings of the twelfth annual conference on Computational learning theory*, pp. 203–208, (1999).

[8] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba, 'Openai gym', *arXiv preprint arXiv:1606.01540*, (2016).

[9] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al., 'Regret analysis of stochastic and nonstochastic multi-armed bandit problems', *Foundations and Trends® in Machine Learning*, **5**(1), 1–122, (2012).

[10] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz, 'Pure exploration in finitely-armed and continuous-armed bandits', *Theoretical Computer Science*, **412**(19), 1832–1852, (2011).

[11] Thomas Degris, Martha White, and Richard S Sutton, 'Off-policy actor-critic', *In 29th International Conference on Machine Learning*, (2012).

[12] Data Flair. Blog post: Tank game in python with source code, 2022.

[13] Aurélien Garivier and Eric Moulines, 'On upper-confidence bound policies for switching bandit problems', in *International Conference on Algorithmic Learning Theory*, pp. 174–188. Springer, (2011).

[14] John Gittins, Kevin Glazebrook, and Richard Weber, *Multi-armed bandit allocation indices*, John Wiley & Sons, 1989.

[15] Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba, 'Mastering atari with discrete world models', *ArXiv*, **abs/2010.02193**, (2021).

[16] D. Jarrett, E. Stride, K. Vallis, and MJ. Gooding, 'Applications and limitations of machine learning in radiation oncology', *The British journal of radiology*, **92**(1100), 20190001, (2019).

[17] Nathan Kallus and Angela Zhou, 'Policy evaluation and optimization with continuous treatments', in *International conference on artificial intelligence and statistics*, pp. 1243–1251. PMLR, (2018).

[18] Robert Kleinberg, 'Nearly tight bounds for the continuum-armed bandit problem', *Advances in Neural Information Processing Systems*, **17**, (2004).

[19] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal, 'Multi-armed bandits in metric spaces', in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 681–690, (2008).

[20] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal, 'Bandits and experts in metric spaces', *J. ACM*, **66**(4), (2019).

[21] Andrey Kolobov, 'Planning with markov decision processes: An ai perspective', *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **6**(1), 1–210, (2012).

[22] Akshay Krishnamurthy, John Langford, Aleksandrs Slivkins, and Chicheng Zhang, 'Contextual bandits with continuous actions: Smoothing, zooming, and adapting', *The Journal of Machine Learning Research*, **21**(1), 5402–5446, (2020).

[23] John Langford, Alekh Agarwal, Miroslav Dudik, Daniel Hsu, Nikos Karampatziakis, Olivier Chapelle, Paul Mineiro, Matt Hoffman, Jake Hofman, Sudarshan Lamkhede, Shubham Chopra, Ariel Faigon, Lihong Li, Gordon Rios, and Alex Strehl. The Vowpal Wabbit (VW) project is a fast out-of-core learning system sponsored by Microsoft Research and (previously) Yahoo! Research, 2022.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, **86**(11), 2278–2324, (1998).

[25] Huitian Lei, Ambuj Tewari, and Susan A Murphy, 'An actor-critic contextual bandit algorithm for personalized mobile health interventions', *arXiv preprint arXiv:1706.09090*, (2017).

[26] Lihong Li, Wei Chu, John Langford, and Robert E Schapire, 'A contextual-bandit approach to personalized news article recommendation', in *Proceedings of the 19th international conference on World wide web*, pp. 661–670, (2010).

[27] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra, 'Continuous control with deep reinforcement learning', *ICLR*, (2016).

[28] Maryam Majzoubi, Chicheng Zhang, Rajan Chari, Akshay Krishnamurthy, John Langford, and Aleksandrs Slivkins, 'Efficient contextual bandits with continuous actions', *Advances in Neural Information Processing Systems (NeurIPS)*, **33**, 349–360, (2020).

[29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al., 'Human-level control through deep reinforcement learning', *nature*, **518**(7540), 529–533, (2015).

[30] Niklas T Rindtorff, MingYu Lu, Nisarg A Patel, Huahua Zheng, and Alexander D'Amour, 'A biologically plausible benchmark for contextual bandit algorithms in precision oncology using in vitro data', *arXiv preprint arXiv:1911.04389*, (2019).

[31] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al., 'Mastering atari, go, chess and shogi by planning with a learned model', *Nature*, **588**(7839), 604–609, (2020).

[32] Nian Si, Fan Zhang, Zhengyuan Zhou, and Jose Blanchet, 'Distributionally robust policy evaluation and learning in offline contextual bandits', in *International Conference on Machine Learning*, pp. 8884–8894. PMLR, (2020).

[33] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller, 'Deterministic policy gradient algorithms', in *International conference on machine learning*, pp. 387–395. PMLR, (2014).

[34] Aleksandrs Slivkins et al., 'Introduction to multi-armed bandits', *Foundations and Trends® in Machine Learning*, **12**(1-2), 1–286, (2019).

[35] Richard S Sutton and Andrew G Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[36] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour, 'Policy gradient methods for reinforcement learning with function approximation', *Advances in neural information processing systems*, **12**, (1999).

[37] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo, 'Openml: Networked science in machine learning', *SIGKDD Explorations*, **15**(2), 49–60, (2013).

[38] Olga Vrousgou, 'Cats - continuous action, contextual bandits'. Workshop at International Conference on Machine Learning ICML, 2021.

[39] Ronald J Williams, 'Simple statistical gradient-following algorithms for connectionist reinforcement learning', *Machine learning*, **8**(3), 229–256, (1992).

[40] Rong Zhu and Mattia Rigotti, 'Deep bandits show-off: Simple and efficient exploration with deep networks', *Advances in Neural Information Processing Systems*, **34**, 17592–17603, (2021).