Infinite Action Contextual Bandits with Reusable Data Exhaust

Mark Rucker [1] **Yinglun Zhu** [2] **Paul Mineiro** [3]

Abstract

For infinite action contextual bandits, smoothed regret and reduction to regression results in state-of-the-art online performance with computational cost independent of the action set: unfortunately, the resulting data exhaust does not have well-defined importance-weights. This frustrates the execution of downstream data science processes such as offline model selection. In this paper we describe an online algorithm with an equivalent smoothed regret guarantee, but which generates well-defined importance weights: in exchange, the online computational cost increases, but only to order smoothness (i.e., still independent of the action set). This removes a key obstacle to adoption of smoothed regret in production scenarios.

1. Introduction

Those who ignore history are doomed to repeat it. A modern variant of this truth arises in controlled experimentation platforms, where offline procedures are a critical complement to online tests, e.g., supporting counterfactual evaluation strategies (Agarwal et al., 2016), offline model selection (Li et al., 2015), and prioritization of scarce online experimental resources (Gomez-Uribe & Hunt, 2015). Consequently, the utility of a learning algorithm is not solely determined by online performance, but also by the post-hoc utility of the data exhaust.

The recent contribution of Zhu & Mineiro (2022) exemplifies this: an online contextual bandit algorithm for infinite action spaces with $O(1)$ space and time complexity with respect to the action set. Unfortunately, this performance is achieved by sampling from a distribution which is not absolutely continuous with the reference measure. Therefore, a variety of post-hoc evaluation procedures that rely on importance-weighting cannot be applied, limiting adoption. In this paper, we describe an alternative approach to infinite action spaces which not only enjoys similar smooth regret guarantee (and empirical performance), but also utilizes sampling distributions with well defined importance-weights. In exchange, we pay an increased computational cost. However, the computational cost only scales with the smoothness of the regret guarantee, rather than the cardinality or dimensionality of the action space per se. Furthermore the new approach does not require an $\arg\min$ oracle, which plays a critical role in the work of Zhu & Mineiro (2022).

Contributions. We highlight our main contributions:

1. In Section 3.2, we present CappedIGW, an algorithm that achieves near-optimal smooth regret guarantees with (i) a sampling distribution that generates reusable data exhaust, and (ii) no dependence on an expansive $\arg\min$ oracle (which is used by previous algorithms).
2. In Section 3.3, we develop algorithms to efficiently implement the algorithm CappedIGW. Our computational complexity only scales with the smoothness parameter, but otherwise has no explicit dependence on the cardinality or dimensionality of the action space. Our implementation leverages techniques from betting martingales (Waudby-Smith & Ramdas, 2020) and is of independent interest for Monte-Carlo integration.

In Section 4, we provide experimental demonstrations exhibiting a combination of equivalent online performance to Zhu & Mineiro (2022) and superior offline utility.

2. Problem Setting

Unfortunately several unusual aspects of our approach demand a tedious exposition: we operate via reduction to regression; we use a nonstandard (smoothed) regret criterion; and our computational complexity claims require careful specification of oracles in the infinite action setting. The impatient reader can skip directly to Section 3 and use this section as reference.

Notation. For functions $f, g : \mathcal{Z} \to \mathbb{R}_+$, we write $f = O(g)$ (resp. $f = \Omega(g)$) if there exists a constant $C > 0$ such that $f(z) \leq Cg(z)$ (resp. $f(z) \geq Cg(z)$) for all $z \in \mathcal{Z}$.

[1]University of Virginia [2]University of Wisconsin-Madison [3]Microsoft Research NYC. Correspondence to: Mark Rucker <mr2an@virginia.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

We write $f = \widetilde{O}(g)$ if $f = O(g \cdot \mathrm{polylog}(T))$, $f = \widetilde{\Omega}(g)$ if $f = \Omega(g/\mathrm{polylog}(T))$. For a set $\mathcal{Z}$, we let $\Delta(\mathcal{Z})$ denote the set of all Radon probability measures over $\mathcal{Z}$. We let $\mathbb{I}_z \in \Delta(\mathcal{Z})$ denote the delta distribution on z. For $x \in \mathbb{R}$ we define $(x)_+ := \max(x, 0)$.

2.1. Contextual Bandits: Reduction to regression

We consider the following standard contextual bandit problems. At any time step $t \in [T]$, nature selects a context $x_t \in \mathcal{X}$ and a distribution over loss functions $\ell_t : \mathcal{A} \rightarrow [0,1]$ mapping from the (compact) action set $\mathcal{A}$ to a loss value in $[0,1]$. Conditioned on the context x_t, the loss function is stochastically generated, i.e., $\ell_t \sim \mathbb{P}_{\ell_t}(\cdot \mid x_t)$. The learner selects an action $a_t \in \mathcal{A}$ based on the revealed context x_t, and obtains (only) the loss $\ell_t(a_t)$ of the selected action. The learner has access to a set of measurable regression functions $\mathcal{F} \subseteq (\mathcal{X} \times \mathcal{A} \rightarrow [0,1])$ to predict the loss of any context-action pair. We make the following standard realizability assumption studied in the contextual bandit literature (Agarwal et al., 2012; Foster et al., 2018; Foster & Rakhlin, 2020; Simchi-Levi & Xu, 2021).

Assumption 1 (Realizability). *There exists a regression function $f^\star \in \mathcal{F}$ such that $\mathbb{E}[\ell_t(a) \mid x_t] = f^\star(x_t, a)$ for any $a \in \mathcal{A}$ and across all $t \in [T]$.*

2.2. Smoothed Regret

Let $(\mathcal{A}, \Omega)$ be a measurable space of the action set and μ be a base probability measure over the actions. Let $\mathcal{Q}_\tau$ denote the set of probability measures such that, for any measure $Q \in \mathcal{Q}_\tau$, the following holds true: (i) Q is absolutely continuous with respect to the base measure μ, i.e., $Q \ll \mu$; and (ii) The Radon-Nikodym derivative of Q with respect to μ is no larger than τ, i.e., $\frac{dQ}{d\mu} \leq \tau$. We call $\mathcal{Q}_\tau$ the set of smoothing kernels at smoothness level τ, or simply put the set of τ-smoothed kernels. For any context $x \in \mathcal{X}$, we denote by $\mathsf{Smooth}_\tau(x)$ the smallest loss incurred by any τ-smoothed kernel, i.e.,

$$\mathsf{Smooth}_\tau(x) := \inf_{Q \in \mathcal{Q}_\tau} \mathbb{E}_{a \sim Q}[f^\star(x, a)].$$

Rather than competing with $\arg\min_{a \in \mathcal{A}} f^\star(x, a)$—which is minimax prohibitive in infinite action spaces— we take $\mathsf{Smooth}_\tau(x)$ as the benchmark and define the *smooth regret* as follows:

$$\mathbf{Reg}_{\mathsf{CB},\tau}(T) := \mathbb{E}\left[\sum_{t=1}^{T} f^\star(x_t, a_t) - \mathsf{Smooth}_\tau(x_t)\right]. \quad (1)$$

One important feature about the above definition is that the benchmark, i.e., $\mathsf{Smooth}_\tau(x_t)$, automatically adapts to the context x_t: this gives the benchmark more power and makes it harder to compete against, compared to previously studied baselines (Chaudhuri & Kalyanakrishnan, 2018; Krishnamurthy et al., 2020).

2.3. Computational Oracles

The first step towards designing computationally efficient algorithms is to identify reasonable oracle models to access the sets of regression functions or actions. Otherwise, enumeration over regression functions or actions (both can be exponentially large) immediately invalidate the computational efficiency. We consider two common oracle models: a regression oracle and a sampling oracle.

The regression oracles. A fruitful approach to designing efficient contextual bandit algorithms is through reduction to supervised regression with the class $\mathcal{F}$ (Foster & Rakhlin, 2020; Simchi-Levi & Xu, 2021; Foster et al., 2020, 2021a). We provide a brief introduction to the reduction technique employed in this paper in Appendix A. Following Foster & Rakhlin (2020), we assume that we have access to an *online* regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$, which is an algorithm for sequential prediction under square loss. More specifically, the oracle operates in the following protocol: At each round $t \in [T]$, the oracle makes a prediction $\widehat{f}_t$, then receives context-action-loss tuple $(x_t, a_t, \ell_t(a_t))$. The goal of the oracle is to accurately predict the loss as a function of the context and action, and we evaluate its performance via the square loss $(\widehat{f}_t(x_t, a_t) - \ell_t(a_t))^2$. We measure the oracle's cumulative performance through the square-loss regret to $\mathcal{F}$, which is formalized below.

Assumption 2. *The regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$ guarantees that, with probability at least $1 - \delta$, for any (potentially adaptively chosen) sequence $\{(x_t, a_t, \ell_t(a_t))\}_{t=1}^{T}$,*

$$\mathbb{E}\Bigg[\sum_{t=1}^{T}\Big(\widehat{f}_t(x_t, a_t) - \ell_t(a_t)\Big)^2 - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T}\big(f(x_t, a_t) - \ell_t(a_t)\big)^2\Bigg] \leq \mathbf{Reg}_{\mathsf{Sq}}(T, \delta),$$

for some (non-data-dependent) function $\mathbf{Reg}_{\mathsf{Sq}}(T, \delta)$.

We will consider the following operations $O(1)$ cost: (i) query the oracle's estimator $\widehat{f}_t$ with context-action pair (x_t, a) and receive its predicted value $\widehat{f}_t(x_t, a) \in [0,1]$; and (ii) update the oracle with example $(x_t, a_t, \ell_t(a_t))$.

Online regression is a well-studied problem, with known algorithms for many model classes (Foster & Rakhlin, 2020; Foster et al., 2020): including linear models (Hazan et al., 2007), generalized linear models (Kakade et al., 2011), nonparametric models (Gaillard & Gerchinovitz, 2015), and beyond. Using Vovk's aggregation algorithm (Vovk, 1998), one can show that $\mathbf{Reg}_{\mathsf{Sq}}(T, \delta) = O(\log(|\mathcal{F}|/\delta))$ for any finite set of regression functions $\mathcal{F}$, which is the canonical setting studied in contextual bandits (Langford & Zhang, 2007; Agarwal et al., 2012). In the following of this paper, we use abbreviation $\mathbf{Reg}_{\mathsf{Sq}}(T) := \mathbf{Reg}_{\mathsf{Sq}}(T, T^{-1})$, and will keep the $\mathbf{Reg}_{\mathsf{Sq}}(T)$ term in our regret bounds to accommodate for general set of regression functions.

The sampling oracle. In order to design algorithms that work with large/continuous action spaces, we assume access to a sampling oracle $\mathbf{Alg}_{\mathsf{Sample}}$ to get access to the action space. In particular, the oracle $\mathbf{Alg}_{\mathsf{Sample}}$ returns an action $a \sim \mu$ randomly drawn according to the base probability measure μ over the action space $\mathcal{A}$. We consider this operation $O(1)$ cost.

Representing the actions. In practice the number of bits required to represent any action $a \in \mathcal{A}$ scales with $O(\log|\mathcal{A}|)$ with a finite set of actions and $\widetilde{O}(d)$ for actions represented as vectors in $\mathbb{R}^d$. Nonetheless we consider this $O(1)$, i.e., we elide the representational overhead in big-O notation for our computational analysis.

3. Algorithms

3.1. Background: SmoothIGW

Zhu & Mineiro (2022) designed an oracle-efficient SmoothIGW that achieves a $\sqrt{T}$-type regret under the smooth regret defined in Eq. (1). Algorithm 1 contains the pseudo code of the SmoothIGW algorithm. At each round $t \in [T]$, the learner observes the context x_t from the environment, obtains the estimator $\widehat{f}_t$ from the regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$, and computes the greedy action $\widehat{a}_t$. It then constructs a sampling distribution P_t by mixing a smoothed inverse gap weighted (IGW) distribution (Abe & Long, 1999; Foster & Rakhlin, 2020) and a delta mass at the greedy action. The algorithm samples an action $a_t \sim P_t$ and updates the regression oracle.

Algorithm 1 SmoothIGW (Zhu & Mineiro, 2022)

```
Input: Exploration parameter $\gamma > 0$; online regression
    oracle $\mathbf{Alg}_{\mathsf{Sq}}$.
1: for $t = 1, 2, \dots, T$ do
2:     Observe context $x_t$.
3:     Receive $\widehat{f}_t$ from regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$.
4:     Get $\widehat{a}_t := \arg\min_{a \in \mathcal{A}} \widehat{f}_t(x_t, a)$.
5:     Set
           $P_t := M_t + (1 - M_t(\mathcal{A})) \cdot \mathbb{1}_{\widehat{a}_t}$
       where $M_t$ is the measure defined in Eq. (2)
6:     Sample $a_t \sim P_t$ and observe loss $\ell_t(a_t)$.
7:     Update $\mathbf{Alg}_{\mathsf{Sq}}$ with $(x_t, a_t, \ell_t(a_t))$
```

The measure M_t on line 5 of Algorithm 1 is defined by the following density with respect to the reference measure,

$$\frac{dM_t}{d\mu}(a) := \frac{\tau}{\tau + \gamma \cdot \left(\widehat{f}_t(x_t, a) - \widehat{f}_t(x_t, \widehat{a}_t)\right)}. \tag{2}$$

Note that M_t is only a sub-probability measure since $dM_t/d\mu(a) \le 1$, hence an additional $(1 - M_t(\mathcal{A})) \cdot \mathbb{1}_{\widehat{a}_t}$ term is needed (to make sure that P_t is a probability measure).

The Problems. While SmoothIGW is the first oracle-efficient contextual bandit algorithm that works with smooth regret, it is not without problems. We highlight two problems associated with SmoothIGW below.

- **The** $\arg\min$ **oracle.** Note that Algorithm 1 requires an exact $\arg\min$ oracle to compute the greedy action $\widehat{a}_t$ (on line 4), which is later on used to construct the sampling distribution P_t (on line 5). However, when working with large, and potentially continuous, action spaces, it can be computationally expensive to obtain such an exact $\arg\min$ oracle. For their experiments, Zhu & Mineiro (2022) construct a regressor class with an $O(1)$ $\arg\min$ oracle, but their construction induces a unimodal $\widehat{f}_t$, which may not always be appropriate.
- **Insufficient data reuse.** While the M_t term is always absolutely continuous with respect to the base measure μ, the delta distribution $\mathbb{1}_{\widehat{a}_t}$ is not absolutely continuous with respect to μ in many common cases, e.g., when μ is the Lebesgue measure in $\mathbb{R}_d$. As a result, a variety of post-hoc procedures that rely on importance-weighting cannot be applied. Unfortunately, to achieve $O(\sqrt{T})$ regret, SmoothIGW uses $\gamma \propto \sqrt{T}$, which implies that the fraction of actions sampled from the $\mathbb{1}_{\widehat{a}_t}$ component increases with horizon length, e.g., Fig. 3.

These two drawbacks frustrate the deployment of SmoothIGW in real-world applications.

3.2. New Approach: CappedIGW

Resolution of the above issues requires eliminating the use of the greedy action $\widehat{a}_t$, which occurs in two places:

- **Inverse-gap weighting.** In sub-probability measure M_t, its density (with respect to μ) on any action a is defined to be inversely proportional to the empirical loss gap $\left(\widehat{f}_t(x_t, a) - \widehat{f}_t(x_t, \widehat{a}_t)\right)$: here, we use $\widehat{f}_t(x_t, \widehat{a}_t)$ as a *benchmark* to compute the loss gap.
- **Pseudo normalization.** Since M_t is only a sub-probability measure, to actually sample from a probability measure, SmoothIGW shifts the remaining probability mass to the delta distribution at the greedy action, i.e., $\mathbb{1}_{\widehat{a}_t}$: here, we use $\mathbb{1}_{\widehat{a}_t}$ to pseudo normalize the sub-probability measure M_t.

In the sequel we eliminate use of the greedy action.

Sampling Density. Let $\beta_t, \kappa_t \in \mathbb{R}$ be two parameters (whose values will be computed later). We consider a probability measure P_t whose density with respect to the base measure μ is defined as follows:

$$\frac{dP_t}{d\mu}(a) = \kappa_t \frac{\tau}{1 + \gamma \left(\widehat{f}_t(x_t, a_t) - \beta_t \right)_+}, \tag{3}$$

where $(x)_+ := \max(x, 0)$. Relative to SmoothIGW:

- We replace the old loss gap benchmark $\widehat{f}_t(x_t, \widehat{a}_t)$ by the new parameter β_t; we also take another $\max$ operation over $\widehat{f}_t(x_t, a) - \beta_t$ and 0 to ensure the positivity of the loss gap. This was inspired by observing the optimal τ-smooth policy plays uniformly over the τ^{-1}-th quantile of the true f^*, but is ultimately justified by the regret decomposition in the proof of Theorem 1.
- We use κ_t as a normalization factor instead of shifting mass to $\mathbb{1}_{\widehat{a}_t}$; the normalization factor κ_t is determined by the choice of β_t via

$$\kappa_t = 1 \Bigg/ \mathbb{E}_{a_t \sim \mu} \left[\frac{\tau}{1 + \gamma \left(\widehat{f}_t(x_t, a_t) - \beta_t \right)_+} \right].$$

With this new sampling distribution in Eq. (3) at hand, we develop a new algorithm for smooth regret, shown next in Algorithm 2.

Algorithm 2 CappedIGW

```
Input: Exploration parameter $\gamma > 0$; online regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$.
for $t = 1, 2, \ldots, T$ do
    Observe context $x_t$.
    Receive $\widehat{f}_t$ from regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$.
    Compute $\beta_t$.                                  // Algorithm 3
    Sample $a_t \sim P_t$                               // Eq. (3), Algorithm 4
    Observe loss $\ell_t(a_t)$.
    Update $\mathbf{Alg}_{\mathsf{Sq}}$ with $(x_t, a_t, \ell_t(a_t))$
```

We will show in next section that β_t can be computed efficiently in $\widetilde{O}(\tau \log \tau)$ calls to the sampling oracle. First we state a regret guarantee.

Theorem 1. *Fix any smoothness level $\tau \geq 1$. Suppose $\forall t$: $\kappa_t \geq 1$ and let κ_∞ be an upper bound on κ_t for $\forall t$. By setting the exploration parameter $\gamma = \sqrt{8T\kappa_\infty \tau / \mathbf{Reg}_{\mathsf{Sq}}(T)}$, Algorithm 2 ensures that*

$$\mathbf{Reg}_{\mathsf{CB},\tau}(T) \leq \sqrt{4T\,\tau\kappa_\infty \mathbf{Reg}_{\mathsf{Sq}}(T)}.$$

Proof. See Appendix B. □

The guarantee in Theorem 1 is the same as the guarantee for SmoothIGW (which is near-optimal) up to a $\sqrt{\kappa_\infty}$ factor. Since we can always find appropriate β_t, κ_t to ensure $\kappa_\infty = O(1)$, we can efficiently achieve the near-optimal smooth regret guarantees without (i) an $\arg\min$ oracle, and (ii) with full data exhaust reuse.

Adapting to an unknown smoothness level τ. We can simply replace SmoothIGW with CappedIGW in Zhu & Mineiro (2022, Thm. 2) to build (i) Pareto optimal algorithms with unknown smoothness level τ, and (ii) develop nearly minimax optimal algorithms under the standard regret for bandits with multiple best arms Zhu & Nowak (2020) and Lipschitz/Hölder bandits Kleinberg (2004); Hadiji (2019): see Section 4 and Section 5 in Zhu & Mineiro (2022) for details.

3.3. Efficient Implementation

In this section, we discuss how to efficiently (i) compute parameter β_t and (ii) sample actions from the distribution P_t. We first notice that the condition $\kappa_t \geq 1$ is critical to Theorem 1. Intuitively, β_t must be chosen so that Algorithm 2 plays a policy which is at most τ-smooth. Because we are competing with τ-smooth policies, it makes sense to be less smooth than the competitor but not to be more smooth than the competitor (further, as described at the end of Section 3.2, the appropriate level for τ can be adaptively chosen).

Consistent with Theorem 1, our task is to find a β_t such that

$$\mathbb{E}_{a_t \sim \mu} \left[\frac{\tau}{1 + \gamma \left(\widehat{f}_t(x_t, a_t) - \beta_t \right)_+} \right] \in \left[\frac{1}{\kappa_\infty}, 1 \right]. \tag{4}$$

First, we establish that it is provably possible to satisfy Eq. (4) with high probability using $O\left(\tau \log(^{(\tau+\gamma)}/_{\delta})\right)$ samples from the reference measure.

Theorem 2. *With the choice $\kappa_\infty = 24$, with probability at least $(1 - \delta)$, it is possible to estimate β satisfying Eq. (4) using $O\left(\tau \log(^{(\tau+\gamma)}/_{\delta})\right)$ samples from μ.*

Proof. See Appendix C □

Theorem 2 uses a fixed sampling strategy which is amenable to analysis and provably terminates after $\widetilde{O}(\tau \log \tau)$ samples. However, this fixed sampling strategy is unnecessarily conservative in practice. To obtain a better empirical performance, instead, we use Algorithm 3—an anytime-valid technique–to ensure early-termination whenever possible. In lieu of proving termination, we backstop Algorithm 3 with Theorem 2, which leads to at most doubling the number of samples required.

Algorithm 3 Normalization CS to compute β_t. The subroutine `BettingMartingale.Update` is defined in Appendix D.

```
Input: $\widehat{f}_t$ (from regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$); exploration parameter $\gamma > 0$; failure probability $\delta$; and $\kappa_\infty \geq 1$.
                                              // It suffices to take $\kappa_\infty = 24$
1: Let $n_{\max} = O(\tau \log(\gamma/\delta))$             // from Theorem 2
2: $l, u \leftarrow \frac{1-\tau}{\gamma}, 1$                // Because $\widehat{f}_t(x_t, \cdot) \in [0,1]$
3: for $n = 1, 2, \ldots, n_{\max}$ do
4:     Sample $a_n \sim \mu$.
5:     Let $g_n(\cdot) = \frac{\tau}{1+\gamma\left(\widehat{f}_t(x_t,a_n)-(\cdot)\right)_+}$.
6:     $l_n, u_n \leftarrow$ BettingMartingale.Update$(g_n; \kappa_\infty; \delta)$
7:     if $l_n > u_n$ then
8:         return $l_n$     // Satisfies Eq. (4) w.p. $(1-\delta)$
9: tail call Theorem 2     // Never happens in practice
```

Theorem 3. *If Algorithm 3 returns a value on line 8, that value satisfies Eq. (4) with probability at least $(1-\delta)$ with respect to the realizations from line 4.*

Proof. See Appendix E □

In practice, Algorithm 3 is vastly more sample efficient than the procedure from Theorem 2: see Table 1 for an empirical comparison. Algorithm 3 operates by maintaining two betting martingales, one of which tries to refine a lower bound on β and the other an upper bound. We defer complete details to Appendix D.

As a motivation, note the combination of betting martingales and no-regret algorithms yields a test with asymptotic optimal power (Casgrain et al., 2022), but which can be safely composed with any stopping rule (e.g., line 8 of Algorithm 3). Early stopping is advantageous to the extent $\widehat{f}_t$ is closer to a constant function, because evidence regarding the normalization constant accumulates more rapidly than accounted for by Theorem 2.

Algorithm 4 Sampling routine

```
Input: $\widehat{f}_t$ (from regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$); $\beta_t$ (from Algorithm 3); exploration parameter $\gamma > 0$
1: while true do
2:     Sample $a_t \sim \mu$.
3:     Compute $p_{\text{accept}} := \frac{1}{1+\gamma\left(\widehat{f}_t(x_t,a_t)-\beta_t\right)_+}$.
4:     With probability $p_{\text{accept}}$, return $a_t$.
```

Efficiently sampling $a_t \sim P_t$. Algorithm 4 is an efficient rejection sampling on the density from Eq. (3). Note that p_{accept} on line 3 is proportional to the desired sampling density P_t defined in Eq. (3), but at most 1. Hence we have the following two established properties of rejection sampling:

1. If line 4 returns an action a_t, then the action a_t is distributed according to Eq. (3);
2. The number of samples required before Algorithm 4 terminates is geometrically distributed with mean $\kappa_t \tau$. In particular, with high probability the number of samples is $O(\kappa_t \tau)$ due to exponential tail bounds.

Computing κ_t. The astute reader will notice that κ_t need not be computed explicitly for Algorithm 2, i.e., for online inference. However an estimate of κ_t might be useful for having more accurate importance-weights for offline reuse. For our experiments we use the naive constant estimate $\widehat{\kappa}_t = 1$, and leave this an area for future work.

4. Experiments

We conduct multiple experiments in this section. In Section 4.1, we empirically compare the performance of Theorem 2 and Algorithm 3. We compare our algorithm CappedIGW with the previous state-of-the-art algorithm SmoothIGW (Zhu & Mineiro, 2022) in terms of both the online performance (Section 4.2) and the offline utility (Section 4.3). We also demonstrate why SmoothIGW lacks offline utility in Section 4.4. Code to reproduce all experiments available at `https://github.com/mrucker/onoff_experiments`.

4.1. Normalization CS

This experiment establishes the empirical validity and efficacy of Algorithm 3. For these simulations we use the unit interval as the action space; Lebesgue reference measure; $\widehat{f}_t(x_t, a_t) = 1_{2a_t\tau > 1}$, corresponding to loss function which is a narrow "needle in the haystack"; failure probability $\delta = 2.5\%$; and $\kappa_\infty = 24$. As indicated in Table 1, Algorithm 3 is a vast improvement over the procedure from Theorem 2. Note in Table 1, κ_t is the true value computed analytically from the β_t produced by Algorithm 3.

Table 1. Algorithm 3 is vastly more sample efficient than the procedure from Theorem 2. The n and κ_t from Algorithm 3 are random variables: shown are 95% bootstrap CI of the realization (*not* the population mean) over different sampler seeds.

τ	γ	n (THM 2)	n (ALG 3)	κ_t (ALG 3)
2	16	942	[18, 24]	[1.3, 3.0]
20	304	13496	[123, 227]	[10.2, 11.8]
200	6368	177141	[2254, 2788]	[1.8, 23.6]

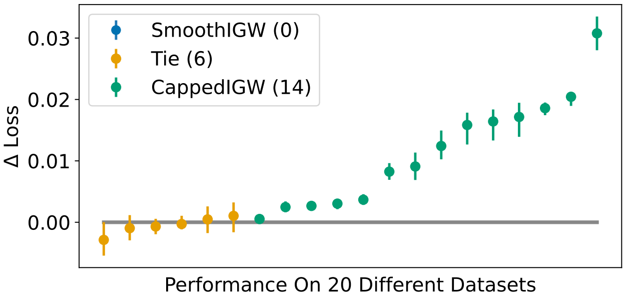

Figure 1. Consistent with their similar theoretical guarantees, online performance of SmoothIGW and CappedIGW is similar, although CappedIGW enjoys a slight advantage. Each datapoint represents a single dataset. Plotted here is (Loss(SmoothIGW) − Loss(CappedIGW)) with 90% bootstrap CIs, i.e., larger values in the plot indicate CappedIGW is outperforming SmoothIGW. Win/tie/loss is determined by if the CI contains 0.

4.2. Online Regret

Here we demonstrate that SmoothIGW has similar online regret to CappedIGW. We use twenty regression datasets converted to contextual bandit datasets with action space $\mathcal{A} = [0, 1]$ via a supervised-to-bandit transformation (Bietti et al., 2021). Each dataset is individually shifted and scaled so that target value $y_t \in [0, 1]$. When an algorithm plays action $a_t \in [0, 1]$, it receives bandit feedback $\ell_t(a_t) := |y_t - a_t|$.

We assess each algorithm (SmoothIGW, CappedIGW) on progressive validation loss. (Blum et al., 1999). For each dataset we run both algorithms using the same set of 30 different seeds, where a seed controls all non-determinism (including data set shuffling, parameter initialization, and action sampling). For each dataset we compute the average of the paired (by seed) differences between each algorithm, and then compute a 90% bootstrap confidence interval.

The two algorithms are declared to have tied on a dataset when the 90% CI for their difference contains a 0. Otherwise one of the algorithms is declared to win. In total we observe five ties, one small SmoothIGW win, and fourteen small CappedIGW wins. The complete result can be seen in Figure 1.

This experiment also demonstrates the effectiveness of Algorithm 3 within CappedIGW. In this experiment CappedIGW determines β_t each iteration using Algorithm 3 with $\kappa_\infty = 4$.

For further details (e.g., model class for $\widehat{f}_t$) see Appendix F.

4.3. Offline Utility

This experiment provides an example of the increased utility of CappedIGW's data exhaust for offline learning relative to SmoothIGW's exhaust. Here we mimic a typical production goal of evaluating a more complicated model class than was used online qua Gomez-Uribe & Hunt (2015, §4.6). As shown in Fig. 2, offline learners trained on CappedIGW exhaust exhibit statistically significant smaller average loss on twelve of twenty datasets.

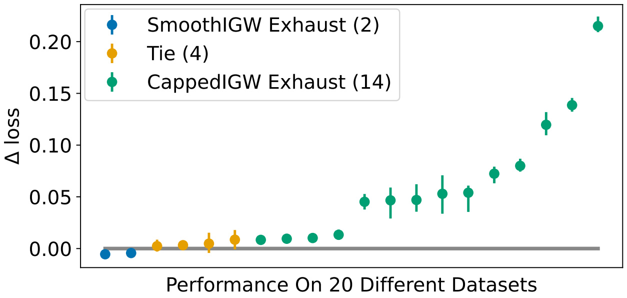

Figure 2. Policies trained offline using CappedIGW data exhaust exhibit less average loss online compared to policies trained offline using SmoothIGW data exhaust. Plotted here is (Loss(SmoothIGW) − Loss(CappedIGW)) with 90% bootstrap CIs, i.e., larger values in the plot indicate training on CappedIGW exhaust is superior to training on SmoothIGW exhaust. Win/tie/loss is determined by if the CI contains 0.

To generate data exhaust all $(x_t, a_t, \widehat{P}_t(a_t), \ell_t(a_t))$ were logged during the online experiments described in Section 4.2, where

$$\widehat{P}_t(a_t) := \frac{\tau}{1 + \gamma\left(\widehat{f}_t(x_t, a_t) - \beta_t\right)_+},$$

i.e., we (naively) estimate $\widehat{\kappa}_t = 1$. We use the inverse of $\widehat{P}_t(a_t)$ as the importance weight.

For each resulting dataset (SmoothIGW exhaust or CappedIGW exhaust), the best of two off-policy learning methods was selected: the direct method (Dudík et al., 2011), which does not use importance-weights; and clipped IPS (Strehl et al., 2010), where for SmoothIGW exhaust we assign the greedy action the maximum importance weight of 5.

To train the offline models data exhaust is split 80%-10%-10% for training, validation and testing respectively. Training epochs are performed on the training set until a decrease in model performance is observed on the validation set. After training learners are assessed using the average loss on the test set. Note validation and test evaluation are independent of what data exhaust was used to train, as the source datasets contain the true label and therefore admit on-policy evaluation.

For each dataset we run the offline learners 30 times using the exhaust files generated from the 30 online seeds. For each dataset we compute the average of the paired (by exhaust) differences between offline learners and then compute a 90% bootstrap confidence interval.

For further details see Appendix G.

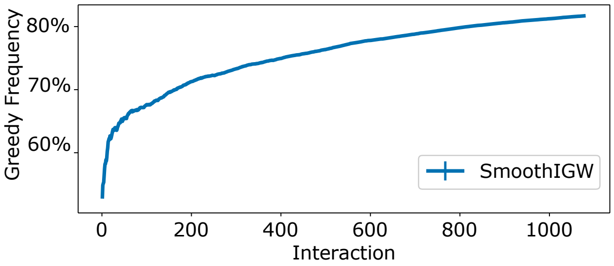

Figure 3. With SmoothIGW, over time an increasing fraction of data exhaust has no importance-weight. (Not shown) With CappedIGW the data exhaust always has an importance-weight.

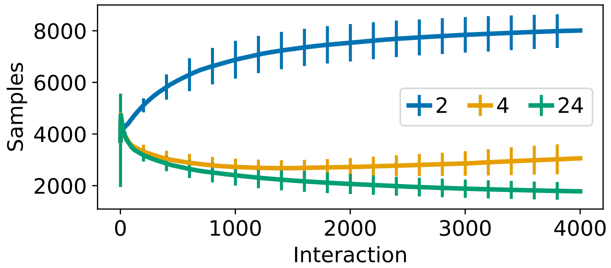

Figure 4. We see an increase in the number of samples required to estimate β_t on each learning update of CappedIGW as κ_∞ shrinks.

4.4. SmoothIGW Increasingly Plays Greedy

Here we show the frequency that SmoothIGW plays its greedy action during the online experiment described in Section 4.2. This is not a problem for online performance. Rather, as described in Section 3.1, this only becomes a problem when attempting to conduct post-hoc analysis with importance-weighting techniques. We can see in Fig. 3 that by the $1,000^{th}$ learning iteration in the online experiment over 80% of played actions no longer have usable importance weights for post-hoc analysis.

4.5. CappedIGW Sensitivity to κ_∞

Here we look at the effect of varying levels of κ_∞ on online and offline performance using our 20 Datasets. For these experiments we set κ_∞ equal to 2, 4, and 24 (note, κ_∞ was 4 for experiments in Section 4.2 and 4.3).

In our experiments the value of κ_∞ strongly impacted the number of samples required to estimate β_t with smaller values of κ_t requiring more samples (Fig. 4). This is expected given that smaller values of κ_∞ indicate tighter confidence bounds on β_t.

We observe a negligible impact in online performance for the three levels of κ_∞ (Fig. 5). In most datasets the average reward seen was nearly identical at all levels. At the same time we observe a slight increase in offline utility when training on the online exhaust generated with κ_∞ equal to 4 or 24 (Fig. 6).

For further details see Appendix H.

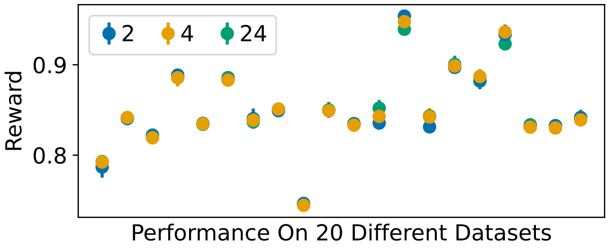

Figure 5. We see very little difference in online performance across our 20 datasets for varying levels of κ_∞ in CappedIGW.

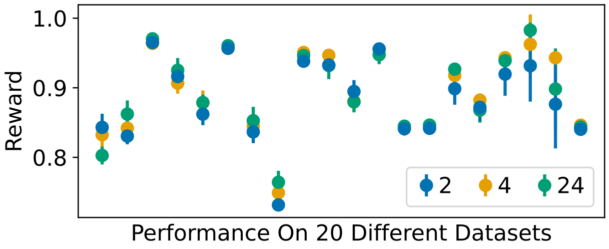

Figure 6. We tend to see increased offline performance across our 20 datasets with data exhaust from CappedIGW when κ_∞ is 4 or 24.

5. Additional Related Work

In this section, we briefly highlight related work that we have not already mentioned in previous sections.

Large action spaces with additional assumptions. Unlike contextual bandits with finite action sets, infinite (or very large) action space contextual bandits are minimax intractable—as observed from the lower bound in (Agarwal et al., 2012). Nonetheless, the setting with infinite action spaces are highly practical in many real-world scenarios, e.g., in large-scale recommender systems. To make progress in this setting, researchers have develop algorithms that work with additional modeling assumptions, such as contextual bandits with linear functions (Auer, 2002; Li et al., 2010; Abbasi-Yadkori et al., 2011), with linearly-structured actions and general (context) function approximation (Foster et al., 2020; Xu & Zeevi, 2020; Zhu et al., 2022), with Lipschitz/Hölder regression functions (Kleinberg, 2004; Hadiji, 2019), and with convex functions (Lattimore, 2020). While these modeling assumptions have lead to fruitful theoretical guarantees, they might be violated in practice.

Large action spaces with smooth regret. An alternative line of research to tackling the large action spaces problems, in which this paper sits, is to weaker the competing benchmark to avoid the otherwise minimax negative result. This idea was first proposed in non-contextual bandits by Chaudhuri & Kalyanakrishnan (2018), where they compete against the $1-\alpha$th quantile (of reward) instead of highest reward. In the case with contextual bandits, Krishnamurthy et al. (2020) proposed a variant of the smooth regret (defined in Section 2.2) for agnostic policy-based analysis. Krishnamurthy et al. (2020) develops algorithms that are statistically optimal, but computationally intractable; a computationally tractable instantiation was later developed in Majzoubi et al. (2020) (but with slightly weaker statistical performance). We remark here that, even though our definition of smoothed regret (in Section 2.2) dominates the one appearing in Krishnamurthy et al. (2020), our approach requires an additional realizability assumption to reduce to regression (instead of classification); Foster et al. (2020) shows how to manage misspecification within a reduction to regression framework.

Offline learning in contextual bandits. Offline learning, or off-policy evaluation, considers the problem of learning a new policy/model only using historic logging data collected from other online policies. Because offline learning permits learning/testing without costly online exploration, it has been used in many real-world applications, such as recommender systems (Thomas et al., 2017) and healthcare industry (Nie et al., 2021). Focusing on contextual bandits, the method of inverse propensity scoring (IPS) (Horvitz & Thompson, 1952) has been extensively used to correct the mismatch between action distributions under the offline and online policies. Besides the IPS method, the direct method (DM) (Dudík et al., 2011; Rothe, 2016) has also been used in offline learning where the learner first learns a reward estimator based on the offline data and then evaluates the new policies.

6. Discussion

This work exhibits a statistical free lunch: the online regret guarantee of an algorithm is essentially unchanged, while the subsequent offline utility of the data exhaust is increased.[1] We speculate this is not typical but rather an artifact of the sub-optimality of the prior technique. In other words, we anticipate that online regret and offline utility are conflicting objectives that must be traded off, suggesting a currently unknown Pareto frontier remains to be discovered. The empirical study of Williams et al. (2021) provides evidence in this direction.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. In *NIPS*, volume 11, pp. 2312–2320, 2011.

Abe, N. and Long, P. M. Associative reinforcement learning using linear probabilistic concepts. In *ICML*, pp. 3–11. Citeseer, 1999.

Agarwal, A., Dudík, M., Kale, S., Langford, J., and Schapire, R. Contextual bandit learning with predictable rewards. In *Artificial Intelligence and Statistics*, pp. 19–26. PMLR, 2012.

Agarwal, A., Bird, S., Cozowicz, M., Hoang, L., Langford, J., Lee, S., Li, J., Melamed, D., Oshri, G., Ribas, O., et al. Making contextual decisions with low technical debt. *arXiv preprint arXiv:1606.03966*, 2016.

Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Bietti, A., Agarwal, A., and Langford, J. A contextual bandit bake-off. *Journal of Machine Learning Research*, 22(133):1–49, 2021.

Blum, A., Kalai, A., and Langford, J. 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.

Casgrain, P., Larsson, M., and Ziegel, J. Anytime-valid sequential testing for elicitable functionals via supermartingales. *arXiv preprint arXiv:2204.05680*, 2022.

Chaudhuri, A. R. and Kalyanakrishnan, S. Quantile-regret minimisation in infinitely many-armed bandits. In *UAI*, pp. 425–434, 2018.

Dudík, M., Langford, J., and Li, L. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*, 2011.

Foster, D. and Rakhlin, A. Beyond UCB: Optimal and efficient contextual bandits with regression oracles. In *International Conference on Machine Learning*, pp. 3199–3210. PMLR, 2020.

Foster, D., Agarwal, A., Dudik, M., Luo, H., and Schapire, R. Practical contextual bandits with regression oracles. In *International Conference on Machine Learning*, pp. 1539–1548. PMLR, 2018.

Foster, D., Rakhlin, A., Simchi-Levi, D., and Xu, Y. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective. In *Conference on Learning Theory*, pp. 2059–2059. PMLR, 2021a.

Foster, D. J., Gentile, C., Mohri, M., and Zimmert, J. Adapting to misspecification in contextual bandits. *Advances in Neural Information Processing Systems*, 33, 2020.

Foster, D. J., Kakade, S. M., Qian, J., and Rakhlin, A. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021b.

[1] Although there is a computational cost, this is arguably mitigated by eliminating the $\arg\min$ oracle.

Gaillard, P. and Gerchinovitz, S. A chaining algorithm for online nonparametric regression. In *Conference on Learning Theory*, pp. 764–796. PMLR, 2015.

Gomez-Uribe, C. A. and Hunt, N. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):1–19, 2015.

Hadiji, H. Polynomial cost of adaptation for X-armed bandits. *Advances in Neural Information Processing Systems*, 32, 2019.

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Horvitz, D. G. and Thompson, D. J. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

Kakade, S. M., Kanade, V., Shamir, O., and Kalai, A. Efficient learning of generalized linear and single index models with isotonic regression. *Advances in Neural Information Processing Systems*, 24, 2011.

Karampatziakis, N. and Langford, J. Online importance weight aware updates. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 392–399, 2011.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kleinberg, R. Nearly tight bounds for the continuum-armed bandit problem. *Advances in Neural Information Processing Systems*, 17:697–704, 2004.

Krishnamurthy, A., Langford, J., Slivkins, A., and Zhang, C. Contextual bandits with continuous actions: Smoothing, zooming, and adapting. *Journal of Machine Learning Research*, 21(137):1–45, 2020.

Langford, J. and Zhang, T. The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in neural information processing systems*, 20(1):96–1, 2007.

Lattimore, T. Improved regret for zeroth-order adversarial bandit convex optimisation. *Mathematical Statistics and Learning*, 2(3):311–334, 2020.

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

Li, L., Chen, S., Kleban, J., and Gupta, A. Counterfactual estimation and optimization of click metrics in search engines: A case study. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 929–934, 2015.

Majzoubi, M., Zhang, C., Chari, R., Krishnamurthy, A., Langford, J., and Slivkins, A. Efficient contextual bandits with continuous actions. *Advances in Neural Information Processing Systems*, 33:349–360, 2020.

Nie, X., Brunskill, E., and Wager, S. Learning when-to-treat policies. *Journal of the American Statistical Association*, 116(533):392–409, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Rothe, C. The value of knowing the propensity score for estimating average treatment effects. *Available at SSRN 2797560*, 2016.

Simchi-Levi, D. and Xu, Y. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Mathematics of Operations Research*, 2021.

Strehl, A., Langford, J., Li, L., and Kakade, S. M. Learning from logged implicit exploration data. *Advances in neural information processing systems*, 23, 2010.

Thomas, P., Theocharous, G., Ghavamzadeh, M., Durugkar, I., and Brunskill, E. Predictive off-policy policy evaluation for nonstationary decision problems, with applications to digital marketing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(2):4740–4745, Feb. 2017. doi: 10.1609/aaai.v31i2.19104.

Vanschoren, J., Van Rijn, J. N., Bischl, B., and Torgo, L. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

Vovk, V. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56(2):153–173, 1998.

Waudby-Smith, I. and Ramdas, A. Estimating means of bounded random variables by betting. *arXiv preprint arXiv:2010.09686*, 2020.

Williams, J. J., Nogas, J., Deliu, N., Shaikh, H., Villar, S. S., Durand, A., and Rafferty, A. Challenges in statistical analysis of data collected by a bandit algorithm: An empirical exploration in applications to adaptively randomized experiments. *arXiv preprint arXiv:2103.12198*, 2021.

Xu, Y. and Zeevi, A. Upper counterfactual confidence bounds: a new optimism principle for contextual bandits. *arXiv preprint arXiv:2007.07876*, 2020.

Zhu, Y. and Mineiro, P. Contextual bandits with smooth regret: Efficient learning in continuous action spaces. In *International Conference on Machine Learning*, pp. 27574–27590. PMLR, 2022.

Zhu, Y. and Nowak, R. On regret with multiple best arms. *Advances in Neural Information Processing Systems*, 33: 9050–9060, 2020.

Zhu, Y., Foster, D. J., Langford, J., and Mineiro, P. Contextual bandits with large action spaces: Made practical. In *International Conference on Machine Learning*, pp. 27428–27453. PMLR, 2022.

A. Background: Minmax Reduction Design

Our approach is based on the work of Foster et al. (2021b), which we review here. From this work, we define the Decision-Estimation Coefficient for any smoothness level $\tau \geq 1$, function class $\mathcal{F}$, context $x \in \mathcal{X}$, and $\mathbf{Alg}_{\mathsf{Sq}}$ estimate $\widehat{f}$:

$$\mathsf{dec}_\gamma(\mathcal{F}; \widehat{f}, x) := \inf_{P \in \Delta(\mathcal{A})} \sup_{Q \in \mathcal{Q}_\tau} \sup_{f \in \mathcal{F}} \mathbb{E}_{a \sim P, a^\star \sim Q}\left[f(x,a) - f(x,a^\star) - \frac{\gamma}{4} \cdot \left(\widehat{f}(x,a) - f(x,a) \right)^2 \right], \tag{5}$$

where f is the true loss function, Q is the optimal smoothed policy with respect to f, P is a policy of our choosing, and γ is a tunable learning rate. Note that with this formulation $\mathsf{Smooth}_\tau(x) = -\sup_{Q \in \mathcal{Q}_\tau} \mathbb{E}_{a \sim Q}[-f^\star(x,a)]$ with respect to results in the paper.

Our goal is to construct P such that we can derive an upper bound on dec. Because dec is the difference between the expectation of $\mathbf{Reg}_{\mathsf{CB},\tau}$ and $\mathbf{Reg}_{\mathsf{Sq}}$ an upper bound on dec implies that $\mathbf{Reg}_{\mathsf{CB},\tau}(T)$ has an upper bound in terms of $\mathbf{Reg}_{\mathsf{Sq}}(T)$. This allows us to reduce the CB problem to simply minimizing the $\mathbf{Reg}_{\mathsf{Sq}}(T)$ via any regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$ of our choosing.

For our work we prove a bound on the Decision-Estimation Coefficient in Appendix B.1 and from there derive a regret bound in Appendix B.2.

B. Proof of Theorem 1

Theorem 1. *Fix any smoothness level $\tau \geq 1$. Suppose $\forall t : \kappa_t \geq 1$ and let κ_∞ be an upper bound on κ_t for $\forall t$. By setting the exploration parameter $\gamma = \sqrt{8T\kappa_\infty \tau / \mathbf{Reg}_{\mathsf{Sq}}(T)}$, Algorithm 2 ensures that*

$$\mathbf{Reg}_{\mathsf{CB},\tau}(T) \leq \sqrt{4T\,\tau\kappa_\infty \mathbf{Reg}_{\mathsf{Sq}}(T)}.$$

The proof proceeds by first bounding the Decision-Estimation Coefficient (Foster et al., 2021b), after which the regret bound follows almost directly.

B.1. Bounding the Decision-Estimation Coefficient

With respect to any context $x \in \mathcal{X}$ and estimator $\widehat{f}$ obtained from $\mathbf{Alg}_{\mathsf{Sq}}$, we consider Eq. (5).

Lemma 1 (Zhu & Mineiro (2022))**.** *Fix constant $\gamma > 0$ and context $x \in \mathcal{X}$. For any measures P and Q such that $Q \ll P$, we have*

$$\begin{aligned} &\sup_{f \in \mathcal{F}} \mathbb{E}_{a \sim P, a^\star \sim Q}\left[f(x,a) - f(x,a^\star) - \frac{\gamma}{4} \cdot \left(\widehat{f}(x,a) - f(x,a) \right)^2 \right] \\ &\leq \mathbb{E}_{a \sim P}\left[\widehat{f}(x,a) \right] - \mathbb{E}_{a \sim Q}\left[\widehat{f}(x,a) \right] + \frac{1}{\gamma} \cdot \mathbb{E}_{a \sim P}\left[\left(\frac{dQ}{dP}(a) - 1 \right)^2 \right]. \end{aligned}$$

Subsequently we omit the dependence on the context $x \in \mathcal{X}$, and use abbreviations $f(a) := f(x,a)$ and $\widehat{f}(a) := \widehat{f}(x,a)$.

We first notice that for any $Q \in \mathcal{Q}_\tau$ we have $Q \ll P_t$ for P_t defined in Eq. (3): since (i) $Q \ll \mu$ by definition, and (ii) $\mu \ll P_t$. Therefore, applying Lemma 1 we have

$$\begin{aligned} &\mathbb{E}_{a \sim P_t, a^\star \sim Q}\left[f(a) - f(a^\star) - \frac{\gamma}{4} \cdot \left(\widehat{f}(a) - f(a) \right)^2 \right] \\ &\leq \mathbb{E}_{a \sim P_t}\left[\widehat{f}(a) \right] - \mathbb{E}_{a \sim Q}\left[\widehat{f}(a) \right] + \frac{1}{\gamma} \cdot \mathbb{E}_{a \sim P_t}\left[\left(\frac{dQ}{dP_t}(a) - 1 \right)^2 \right]. \end{aligned}$$

Denote $p(a) = \frac{dP_t}{d\mu}(a)$ and $q(a) = \frac{dQ}{d\mu}(a)$. Continuing

$$
\begin{aligned}
&\mathbb{E}_{a\sim P_t}\left[\widehat{f}(a)\right] - \mathbb{E}_{a\sim Q}\left[\widehat{f}(a)\right] + \frac{1}{\gamma}\cdot \mathbb{E}_{a\sim P_t}\left[\left(\frac{dQ}{dP_t}(a) - 1\right)^2\right] \\
&= \mathbb{E}_{a\sim\mu}\left[p(a)\cdot\left(\widehat{f}(a)-\beta\right)\right] - \mathbb{E}_{a\sim\mu}\left[q(a)\cdot\left(\widehat{f}(a)-\beta\right)\right] + \frac{1}{\gamma}\cdot\mathbb{E}_{a\sim\mu}\left[p(a)\cdot\left(\frac{q(a)}{p(a)}-1\right)^2\right] \\
&= \mathbb{E}_{a\sim\mu}\left[p(a)\cdot\left(\widehat{f}(a)-\beta\right)\right] - \mathbb{E}_{a\sim\mu}\left[q(a)\cdot\left(\widehat{f}(a)-\beta\right)\right] + \frac{1}{\gamma}\cdot\mathbb{E}_{a\sim\mu}\left[q(a)\cdot\frac{q(a)}{p(a)} - 2q(a) + p(a)\right] \\
&= \mathbb{E}_{a\sim\mu}\left[p(a)\cdot\left(\widehat{f}(a)-\beta\right)\right] + \frac{1}{\gamma}\cdot\mathbb{E}_{a\sim Q}\left[\frac{q(a)}{p(a)} - \gamma\cdot\left(\widehat{f}(a)-\beta\right)\right] - \frac{1}{\gamma} \\
&= \mathbb{E}_{a\sim\mu}\left[p(a)\cdot\max\left\{0,\widehat{f}(a)-\beta\right\}\right] + \frac{1}{\gamma}\cdot\mathbb{E}_{a\sim Q}\left[\frac{q(a)}{p(a)} - \gamma\cdot\max\left\{0,\widehat{f}(a)-\beta\right\}\right] \\
&\quad + \mathbb{E}_{a\sim\mu}\left[(p(a)-q(a))\cdot\min\left\{0,\widehat{f}(a)-\beta\right\}\right] - \frac{1}{\gamma}. \qquad (6)
\end{aligned}
$$

Now we note the definition of P_t implies

$$
\mathbb{E}_{a\sim\mu}\left[p(a)\cdot\max\left\{0,\widehat{f}(a)-\beta\right\}\right] \leq \frac{\kappa_t\tau}{\gamma}; \qquad (7)
$$

furthermore, the constraints $q(a) \leq \tau$ and $\kappa_t \geq 1$ imply

$$
\begin{aligned}
&\frac{1}{\gamma}\cdot\mathbb{E}_{a\sim Q}\left[\frac{q(a)}{p(a)} - \gamma\cdot\max\left\{0,\widehat{f}(a)-\beta\right\}\right] \\
&\leq \frac{1}{\gamma}\mathbb{E}_{a\sim Q}\left[\frac{1+\gamma\cdot\max\left\{0,\widehat{f}(a)-\beta\right\}}{\kappa_t} - \gamma\cdot\max\left\{0,\widehat{f}(a)-\beta\right\}\right] \\
&\leq \frac{1}{\gamma}; \qquad (8)
\end{aligned}
$$

and finally

$$
\begin{aligned}
&\mathbb{E}_{a\sim\mu}\left[(p(a)-q(a))\cdot\min\left\{0,\widehat{f}(a)-\beta\right\}\right] \\
&\leq (\kappa_t - 1)\tau\;\mathbb{E}_{a\sim\mu}\left[\min\left\{0,\widehat{f}(a)-\beta\right\}\right] \\
&\leq 0. \qquad (9)
\end{aligned}
$$

Substituting Eq. (7), Eq. (8), and Eq. (9) into Eq. (6) yields

$$
\mathbb{E}_{a\sim P_t, a^\star\sim Q}\left[f(a) - f(a^\star) - \frac{\gamma}{4}\cdot\left(\widehat{f}(a) - f(a)\right)^2\right] \leq \frac{\kappa_t\tau}{\gamma}. \qquad (10)
$$

B.2. Finishing the proof

This part is almost verbatim from Zhu & Mineiro (2022, Thm 1, Appendix A.2), but included for completeness.

We use abbreviation $f_t(a) := f(x_t, a)$ for any $f \in \mathcal{F}$. Let $a_t^\star$ denote the action sampled according to the best smoothing kernel within $\mathcal{Q}_\tau$ (which could change from round to round). We let $\mathcal{E}$ denote the good event where the regret guarantee stated in Assumption 2 (i.e., $\mathbf{Reg}_{\mathsf{Sq}}(T) := \mathbf{Reg}_{\mathsf{Sq}}(T, T^{-1})$) holds with probability at least $1 - T^{-1}$. Conditioned on this good event, following the analysis provided in Foster et al. (2020), we decompose the contextual bandit regret as follows.

$$
\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^T f_t^\star(a_t) - f_t^\star(a_t^\star)\right] &= \mathbb{E}\left[\sum_{t=1}^T f_t^\star(a_t) - f_t^\star(a_t^\star) - \frac{\gamma}{4}\cdot\left(\widehat{f}_t(a_t) - f_t^\star(a_t)\right)^2\right] + \frac{\gamma}{4}\cdot\mathbb{E}\left[\sum_{t=1}^T\left(\widehat{f}_t(a_t) - f_t^\star(a_t)\right)^2\right] \\
&\leq T\cdot\frac{\kappa_\infty\tau}{\gamma} + \frac{\gamma}{4}\cdot\mathbb{E}\left[\sum_{t=1}^T\left(\widehat{f}_t(a_t) - f_t^\star(a_t)\right)^2\right],
\end{aligned}
$$

where the bound on the first term follows from Eq. (10). We analyze the second term below.

$$
\begin{aligned}
&\frac{\gamma}{4} \cdot \mathbb{E}\left[\sum_{t=1}^{T}\left(\left(\widehat{f}_t(a_t) - \ell_t(a_t)\right)^2 - \left(f^\star(a_t) - \ell_t(a_t)\right)^2 + 2\left(\ell_t(a_t) - f_t^\star(a_t)\right) \cdot \left(\widehat{f}_t(a_t) - f_t^\star(a_t)\right)\right)\right] \\
&= \frac{\gamma}{4} \cdot \mathbb{E}\left[\sum_{t=1}^{T}\left(\left(\widehat{f}_t(a_t) - \ell_t(a_t)\right)^2 - \left(f_t^\star(a_t) - \ell_t(a_t)\right)^2\right)\right] \\
&\leq \frac{\gamma}{4} \cdot \mathbf{Reg}_{\mathsf{Sq}}(T),
\end{aligned}
$$

where on the second line follows from the fact that $\mathbb{E}[\ell_t(a) \mid x_t] = f^\star(x_t, a)$ and ℓ_t is conditionally independent of a_t, and the third line follows from the bound on regression oracle stated in Assumption 2. As a result, we have

$$
\mathbf{Reg}_{\mathsf{CB},\tau}(T) \leq \frac{T\kappa_\infty \tau}{\gamma} + \frac{\gamma}{4} \cdot \mathbf{Reg}_{\mathsf{Sq}}(T) + O(1),
$$

where the additional term $O(1)$ accounts for the expected regret suffered under event $\neg\mathcal{E}$. Taking $\gamma = \sqrt{8T\kappa_\infty\tau / \mathbf{Reg}_{\mathsf{Sq}}(T)}$ leads to the desired result.

C. Proof of Theorem 2

Theorem 2. *With the choice $\kappa_\infty = 24$, with probability at least $(1-\delta)$, it is possible to estimate β satisfying Eq. (4) using $O\left(\tau \log({}^{(\tau+\gamma)}/_{\delta})\right)$ samples from μ.*

We elide the contextual dependence here, as x_t is a constant for all of these operations.

Define

$$
\begin{aligned}
g(a;\beta) &:= \frac{\tau}{1 + \gamma \max\left(0, \widehat{f}_t(a) - \beta\right)}, \\
\beta_{\min} &:= \frac{1-\tau}{\gamma}, \\
\beta_{\max} &:= 1,
\end{aligned}
$$

where $\widehat{f}_t(a) \in [0,1]$. We note the following properties:

$$
\begin{aligned}
g(a;\beta) &\in [0,\tau], \\
\frac{d}{d\beta} g(a;\beta) &\in [0, \gamma g(a;\beta)], \\
\mathbb{E}_{a\sim\mu}\left[g(x,\beta_{\min})\right] &\leq 1, \\
\mathbb{E}_{a\sim\mu}\left[g(x,\beta_{\max})\right] &\geq 1, \\
\mathbb{E}_{a\sim\mu}\left[g^2(x,\beta)\right] &\leq \tau \mathbb{E}_{a\sim\mu}\left[g(x,\beta)\right].
\end{aligned}
$$

Fixed β bound. With n samples we can estimate the integral $z(\beta) \doteq \mathbb{E}_{a\sim\mu}\left[g(a,\beta)\right]$ at any fixed β from the empirical mean $\bar{z}(\beta)$ via

$$
\begin{aligned}
z(\beta) &\in \bar{z}(\beta) \pm \left(\sqrt{\frac{2\mathbb{E}_{a\sim\mu}\left[g^2(a,\beta)\right]\ln(2/\delta)}{n}} + \tau\frac{\ln(2/\delta)}{3n}\right) && \text{(Bernstein)} \\
&\in \bar{z}(\beta) \pm \left(\sqrt{\frac{2\tau z(\beta)\ln(2/\delta)}{n}} + \tau\frac{\ln(2/\delta)}{3n}\right) && \text{(self-bounding)} \\
&\in \bar{z}(\beta) \pm \left(\frac{1}{2}z(\beta) + \frac{4\tau\ln(2/\delta)}{n} + \tau\frac{\ln(2/\delta)}{3n}\right), && \text{(AM-GM)} \\
z(\beta) &\in \left[\frac{2}{3}\left(\bar{z}(\beta) - \frac{26\tau\ln(2/\delta)}{3n}\right), 2\left(\bar{z}(\beta) + \frac{26\tau\ln(2/\delta)}{3n}\right)\right].
\end{aligned}
$$

with probability at least $1-\delta$.

Picking the β grid. Suppose

$$\frac{26\tau \ln(2/\delta)}{3n} \leq \frac{1}{8},$$

then

$$z(\beta) \in \left[\frac{2}{3}\bar{z}(\beta) - \frac{1}{12}, 2\bar{z}(\beta) + \frac{1}{4}\right],$$

therefore

$$\bar{z}(\beta) \in \left[\frac{3}{16}, \frac{3}{8}\right] \implies z(\beta) \in \left[\frac{1}{24}, 1\right].$$

Thus if we can evaluate $\bar{z}(\beta)$ on a grid where it increases by at most a factor of 2, then we will obtain a β^* such that $z(\beta^*) \in \left[\frac{1}{24}, 1\right]$.

Using the assumptions,

$$\begin{aligned} &\bar{z}'(\beta) \leq \gamma \bar{z}(\beta) \\ \implies\ &\bar{z}(\beta) \leq \bar{z}(\beta_0) \exp\left(\gamma\left(\beta - \beta_0\right)\right) \end{aligned}$$

hence evaluation over a grid spaced as $\Delta\beta = \log(2)\gamma^{-1}$ will ensure $\bar{z}(\beta)$ does not increase by more than a factor of 2. Using a union bound over these points we need

$$\begin{aligned} \frac{\beta_{\max} - \beta_{\min}}{\Delta\beta} &\leq \frac{\tau + \gamma}{\log(2)}, \\ \frac{26\tau\left(\log(2\log(2)) + \log(\tau+\gamma) - \log(\delta)\right)}{3n} &\leq \frac{1}{8}, \\ 8\frac{26\tau\left(\log(2\log(2)) + \log(\tau+\gamma) - \log(\delta)\right)}{3} &\leq n, \end{aligned}$$

thus $n = O\left(\tau \log\left({}^{(\tau+\gamma)}/{}_{\delta}\right)\right)$.

D. Explanation of Algorithm 3

Note the following discussion is localized to a single invocation of Algorithm 3, and therefore we elide the contextual dependence.

Using the notation from the proof of Theorem 2, note that $z(\beta) := \mathbb{E}_{a\sim\mu}\left[g(x,\beta)\right]$ is continuous and non-decreasing in β. Fix $\kappa_\infty > 1$ and define

$$\begin{aligned} \beta_{\kappa_\infty} &:= \sup\left\{\beta \,\middle|\, z(\beta) \leq \kappa_\infty^{-1}\right\}, \\ \beta_1 &:= \inf\left\{\beta \,\middle|\, z(\beta) \geq 1\right\}. \end{aligned}$$

Given a failure probability δ, we will construct an lower confidence sequence L_n for β_1 and an upper confidence sequence U_n for β_{κ_∞}, each with failure probability $\delta/2$, i.e., a pair of adapted random processes L_n and U_n satisfying

$$\mathbb{P}\left(\forall n \in \mathbb{N} : \beta_{\kappa_\infty} \leq U_n\right) \geq 1 - {}^{\delta}/_{2}, \tag{11}$$

$$\mathbb{P}\left(\forall n \in \mathbb{N} : L_n \leq \beta_1\right) \geq 1 - {}^{\delta}/_{2}, \tag{12}$$

where our random processes are defined on the discrete-time filtered probability space $(\Omega, \mathcal{F}, \left\{\mathcal{F}_n\right\}_{n\in\mathbb{N}}, \mathbb{P})$ generated by the sampling oracle in line 4 of Algorithm 3. Standard techniques for achieving Eq. (11) and Eq. (12) are described further below: for now, assuming those properties, note that whenever $L_n \geq U_n$, we can conclude with probability at least $(1-\delta)$ that

$$\beta \in [U_n, L_n] \implies z(\beta) \in \left[\frac{1}{\kappa_\infty}, 1\right]$$

which is the desired property from Eq. (4). Because $z(\beta)$ is non-decreasing and we want the smallest κ_t possible, we use the largest β, and hence return L_n on line 8 of Algorithm 3.

To achieve Eq. (11) and Eq. (12), we combine betting martingales with a no-regret algorithm, aka ONS-m. (Waudby-Smith & Ramdas, 2020) To ease exposition, we describe the lower bound only; the upper bound is analogous. For the lower bound we define the wealth process

$$W_n^{(-)}(\nu; \beta) = \prod_{m=1}^{n}\left(1 + \nu_m\left(1 - g(A_m; \beta)\right)\right),$$

where A_m is the sequence of actions generated by line 4 of Algorithm 3; and $\nu_n \in [0, (\tau - 1)^{-1})$ is a predictable betting sequence (to be specified below). This wealth process is a non-negative martingale with initial value of 1 when evaluated at $\beta = \beta_1$ and therefore due to Ville's inequality

$$\mathbb{P}\left(\forall n \in \mathbb{N} : W_n^{(-)}(\nu; \beta) \leq {}^2\!/\!_\delta\right) \geq 1 - {}^\delta\!/\!_2.$$

Because $g(\cdot, \beta)$ is non-decreasing in β, it follows $L_n = \sup\left\{\beta \,\middle|\, W_n^{(-)}(\nu; \beta) \leq {}^2\!/\!_\delta\right\}$ is a lower confidence sequence for β_1. It remains to specify the betting process ν_n: we use online Newton step to choose bets that maximize the (log) wealth, using loss $\left(-\log W_m^{(-)}(\cdot, L_{m-1})\right)$, and constraining the bet sequence $\nu \in [0, {}^1\!/\!_{2\tau}]$ to ensure a bounded gradient.

The upper bound is similar, but using a martingale of the form $W_n^{(+)}(v; \beta) = \prod_{m=1}^n \left(1 + v_m \left(g(A_m; \beta) - \kappa_\infty^{-1}\right)\right)$, and constraining the bet sequence $v \in [0, {}^{\kappa_\infty}\!/\!_2]$.

E. Proof of Theorem 3

Theorem 3. *If Algorithm 3 returns a value on line 8, that value satisfies Eq. (4) with probability at least* $(1 - \delta)$ *with respect to the realizations from line 4.*

This uses the notation from Appendix D.

From Waudby-Smith & Ramdas (2020, Corollary 1), L_n and U_n satisfy Eq. (11) and Eq. (12) respectively. Therefore, whenever $L_n \geq U_n$, given the monotonicity of $g(\cdot; \beta)$ wrt β, Eq. (4) holds with probability at least $(1 - \delta)$.

F. Online Regret Experiment: Additional Details

We perform the online regret experiment using twenty regression datasets hosted on OpenML (Vanschoren et al., 2014) and released under a CC-BY[2] license. The exact data ids for these datasets are: 150, 422, 1187, 41540, 41540, 42225, 42225, 44025, 44031, 44056, 44059, 44069, 44140, 44142, 44146, 44148, 44963, 44964, 44973, and 44977.

For large datasets a random subset of $80,000$ examples is selected. Features in every data set are transformed so that the i-th feature in sample x_t^i is shifted by $\min_t x_t^i$ and scaled by ${}^1\!/\!_{\max_t x_t^i - \min_t x_t^i}$. This transformation is applied to the labels y_t as well so that for every label $y_t \in [0, 1]$.

During evaluation of SmoothIGW and CappedIGW the contexts x_t is revealed to the learners in batches of eight. The learners then pick an action to play for each context in the batch. After picking their actions learners then receive the loss $\ell_t(a_t) = |a_t - y_t|_1$ for each of the selected actions. This process continues until all examples in a dataset are exhausted.

Both SmoothIGW and CappedIGW assume access to a $\widehat{f}_t$ and SmoothIGW also assumes access to an $\arg\min$ orcale to compute $\widehat{a}_t$. To satisfy these requirements we mirror the implementation pattern of Zhu & Mineiro (2022) where θ are learned parameters, $\widehat{f}_t(x, a; \theta) := g(\widehat{a}(x; \theta) - a; \theta)$, and g is defined so that its global minimizer is 0. With $\theta := (u; w; q; z; \zeta)$ our experiment defines $\widehat{a}(x; \theta) = \sigma\left(u + \langle x, w\rangle\right)$ where σ is the sigmoid function and, given $z = \widehat{a}(x; \theta) - a$,

$$\widehat{g}(x, a; \theta) = \begin{cases} q + \langle w, (z, z^{3/2}, z^2)\rangle & \text{if } z \geq 0 \\ q + \langle \zeta, (|z|, |z|^{3/2}, |z|^2)\rangle & \text{if } z < 0. \end{cases} \tag{13}$$

To optimize θ we use a mean squared error loss with Adam (Kingma & Ba, 2014) in PyTorch (Paszke et al., 2019).

We use the Corral meta-algorithm from Zhu & Mineiro (2022) to select the smoothness parameter τ for both SmoothIGW and CappedIGW. The hyperparameter settings for the meta-algorithm were optimized and fixed globally to give the best average performance across all experiment datasets. For SmoothIGW we use $\eta := 0.3$ and select τ from the set $\{2, 3.76, 7.05, 13.24, 24.87, 46.7, 87.7, 164.69, 309.27, 580.77, 1090.6, 2048\}$. For CappedIGW we use $\eta := 0.3$ and select τ from $\{6, 9.57, 15.28, 24.37, 38.89, 62.05, 99.01, 157.98, 252.08, 402.21, 641.77, 1024\}$.

[2] https://creativecommons.org/licenses/by/2.0/

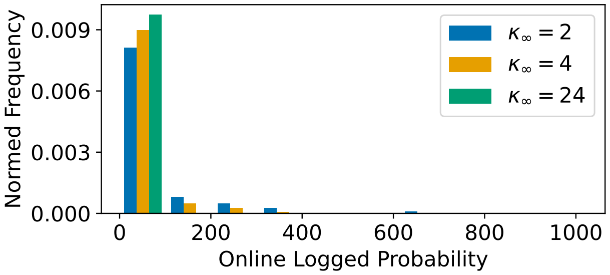

Figure 7. As κ_∞ grows online probability is increasingly near 0.

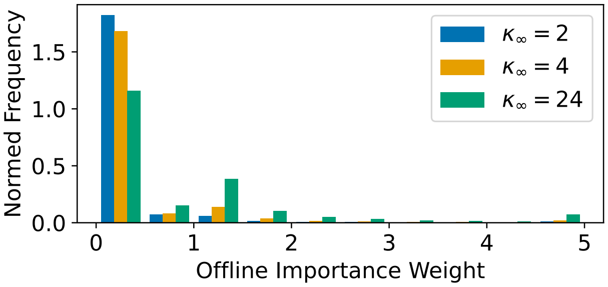

Figure 8. As κ_∞ grows offline weight has a greater spread.

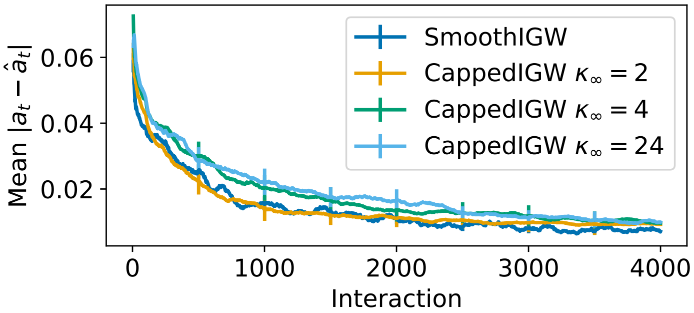

Figure 9. As κ_∞ grows actions have a greater spread around what the learners believes the argmax $\hat{a}_t$ is.

G. Offline Utility Experiment: Additional Details

For the offline experiment we use data exhaust from the online experiment which takes the form of $(x_t, a_t, \mathbb{P}_t(a_t), \ell_t(a_t))$. Because of this the datasets for the offline experiment are identical to those in the online experiment as are the dataset transformations, both of which are described in Appendix F.

The offline learners use the same functional form as the online learners; that is $f(x, a; \theta) := g(a^\star(x; \theta) - a; \theta)$ with the definition of g given in Eq. (13). The offline experiment uses a more complex form for $a^\star(x; \theta)$ than the online learners. Rather than one linear layer with a sigmoid output the offline learners use a three layer feedforward neural network with width equal to the number of features in a dataset, ReLU activation functions, and a sigmoid output.

We optimize offline learner parameters θ using mean squared error loss with Adam (Kingma & Ba, 2014) in PyTorch (Paszke et al., 2019). When using clipped IPS (Strehl et al., 2010) we multiply each mean squared error by its importance weight. It is known that this is not an optimal way to perform importance updates (Karampatziakis & Langford, 2011). Even so, the offline learners still benefit from the importance weighted updates when using CappedIGW exhaust. During testing our offline learners follow the policy $\pi^\star(x) := a^\star(x; \theta)$.

H. CappedIGW Sensitivity to κ_∞: Additional Details

To further understand how κ_∞ influences experimental outcomes we look here at the probabilities logged during online analysis along with the importance weights used during offline analysis. We see in Fig. 7 that for our 20 datasets as κ_∞ became larger logged CappedIGW probabilities tended toward 0. In turn, we see a greater spread in offline importance weights from this data Fig. 8.

Another perspective can be found by looking at the distribution of actions played by our learners. For this we recorded the distance a learner's played action was from what the learner believed the greedy action was. This perspective is only useful in these experiments due to the unimodal implementation of $\hat{f}_t$ (see Appendix F). We see that as κ_∞ increases so to does the spread of actions played around the believed argmax Fig. 9.