GWIDEXPLORE: COMPUTATIONALLY-EFFICIENT EXPLORATION IN CONTEXTUAL LINEAR BANDITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Achieving nearly optimal $\widetilde{O}(d\sqrt{T})$ regret with efficient optimization over large action sets has remained an open problem in contextual linear bandits. Dani, Hayes, and Kakade (Dani et al., 2008) gave an efficient algorithm, but with $\widetilde{O}(d^{3/2}\sqrt{T})$ regret. The problem was posed in (Agrawal and Goyal, 2014) and revisited by (Kim and Tewari, 2020). The oracle-efficient linear Thompson-sampling and spanner-based methods of (Agrawal and Goyal, 2014; Abeille and Lazaric, 2017; Zhu et al., 2022) have guarantees that incur the additional $\sqrt{d}$ factor. We close this gap for compact convex action sets. Under standard boundedness and sub-Gaussian noise assumptions, our algorithm achieves $\widetilde{O}(d\sqrt{T})$ regret with high probability using polylogarithmically many linear optimizations over convex sets per round. The algorithm progressively removes actions that are suboptimal by adding linear constraints to the current action set. It measures the remaining uncertainty using the Gaussian width of a rescaled feasible set, where the rescaling reflects information from previously sampled actions. When the width is small, the algorithm shrinks the set while preserving an optimal action with high probability. When the width is large, it samples a random direction and chooses uniformly between the two feasible actions that are farthest apart in that direction.

1 INTRODUCTION

Contextual linear bandits model sequential decisions where feedback from one action can help predict the rewards of others. At each round t, a learner chooses an action X_t from a set $A_t \subseteq \mathbb{R}^d$ and observes a noisy reward with mean $\langle \theta, X_t \rangle$, where θ is unknown. For example, in personalized recommendation, actions encode candidate items through user and item features, so feedback on one item helps the learner evaluate other items with similar features (Li et al., 2010). In many applications, however, the action set is too large to enumerate, including recommendation from large catalogs (Zhu et al., 2022), routing with exponentially many paths (Awerbuch and Kleinberg, 2008; Hazan et al., 2013), and continuous control or structured resource allocation (Ayala-Romero et al., 2024; Harsha et al., 2025). These settings require methods that exploit the action set structure without evaluating every possibility.

If θ were known, selecting the best action would require only linear optimization over A_t. The central computational question is whether a learner can retain this advantage while exploring enough to achieve nearly optimal regret. Optimism algorithms attain $\widetilde{O}(d\sqrt{T})$ regret, but their action-selection step solves a non-convex optimization problem (Dani et al., 2008; Abbasi-Yadkori et al., 2011). (Dani et al., 2008) gave an efficient algorithm with $\widetilde{O}(d^{3/2}\sqrt{T})$ regret, losing a factor of $\sqrt{d}$. (Hazan et al., 2013) removed this factor for a fixed convex action set: their volumetric-spanner method achieves $\widetilde{O}(d\sqrt{T})$ expected regret for bounded adversarial linear losses, using a membership oracle and log-concave sampling. The contextual counterpart, in which the action set may change from round to round, remained open. This question was posed explicitly by (Agrawal and Goyal, 2014, pp. 7-8); and (Kim and Tewari, 2020) revisited it. Existing oracle-efficient Thompson-sampling and spanner-based methods (Agrawal and Goyal, 2014; Abeille and Lazaric, 2017; Ito et al., 2019; Zhu et al., 2022) retain $\widetilde{O}(d^{3/2}\sqrt{T})$ guarantees in their respective settings.

The difficulty is that sampling allocations designed for a fixed action set cannot be used directly when actions may disappear before receiving their allocated samples. We resolve this statistical-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

computational gap for compact convex action sets. Under standard boundedness and sub-Gaussian noise assumptions, our algorithm achieves $\widetilde{O}(d\sqrt{T})$ regret with high probability for any action set sequence fixed before interaction. It uses only polylogarithmically many linear optimizations per round, together with a polynomial additional computation. Each optimization is over the current action set intersected with logarithmically many linear inequalities. Section 3 states the precise guarantees and oracle model.

The algorithm uses Gaussian width to decide both which actions to eliminate and when to explore. It draws a Gaussian coefficient vector (random objective) with larger variance in directions where previous samples provide less information. It then finds the surviving actions that maximize and minimize the corresponding linear objective. The expected difference between these two values is the Gaussian width, which measures the uncertainty of the reward estimates. When the width is small, confidence bounds certify that actions with low estimated rewards can be safely eliminated, while preserving an optimal action with high probability. When it is large, the learner explores by choosing one of the two extreme actions uniformly at random. We show that exploration regret is bounded by the Gaussian width, while the exploration rule ensures that a large width cannot occur too often. Combining these results yields the bound $\widetilde{O}(d\sqrt{T})$. Section 5 gives the key inequalities.

Our proof must control the dependence between sampled actions, surviving action set, and reward noise. To do so, the algorithm maintains separate estimators: each estimator may eliminate actions for later estimators, but its rewards do not influence its own exploration decisions. For history-dependent noise, this separation alone is not sufficient. We additionally use randomized reward rounding and a coupling argument to reduce to a response model in which conditioned on the actions sampled, the corresponding reward errors are independent. Section 4 describes the full algorithm.

2 Related work

The statistical-computational gap. Dani, Hayes, and Kakade (Dani et al., 2008) established the $\widetilde{O}(d\sqrt{T})$ statistical benchmark for contextual linear bandits. Self-normalized confidence bounds (Abbasi-Yadkori et al., 2011) improve the logarithmic factors, with further refinements in (Li et al., 2019; 2021). Dani et al.'s computationally efficient algorithm has $\widetilde{O}(d^{3/2}\sqrt{T})$ regret (Dani et al., 2008, Theorem 2 and Section 3.4). Agrawal and Goyal explicitly posed the question of achieving $\widetilde{O}(d\sqrt{T})$ regret efficiently in contextual linear bandits (Agrawal and Goyal, 2014, Section 2.4), and (Kim and Tewari, 2020, Section 5) revisited it in 2020.

Exploration over large action spaces. Geometric exploration methods include barycentric spanners (Awerbuch and Kleinberg, 2008), exponential weights (Bubeck et al., 2012), and volumetric spanners (Hazan et al., 2013). For fixed convex action sets, volumetric spanners achieve $\widetilde{O}(d\sqrt{T})$ expected regret under bounded adversarial linear losses using membership access and log-concave sampling (Hazan et al., 2013). With i.i.d. losses and fixed action sets, Ito et al. (Ito et al., 2019) obtain $\widetilde{O}(d^{3/2}\sqrt{T})$ regret efficiently. For changing action sets, SpannerIGW (Zhu et al., 2022) supports nonlinear context dependence through a regression oracle; under its spanner-initialization assumptions, the efficient implementation gives $\widetilde{O}(d^{3/2}\sqrt{T})$ regret for d-dimensional linear predictors.

Thompson sampling and related randomized methods. (Agrawal and Goyal, 2014, Theorem 1) prove $\widetilde{O}(d^{3/2}\sqrt{T})$ regret for oracle-efficient contextual Thompson sampling; (Abeille and Lazaric, 2017) give an alternative proof for fixed action sets. (Abeille et al., 2025) identify conditions yielding $\widetilde{O}(d\sqrt{T})$ frequentist regret for unmodified Thompson sampling on fixed action sets. Several works modify Thompson sampling to attain $\widetilde{O}(d\sqrt{T})$ frequentist regret without a provably efficient implementation for large action spaces (Zhang, 2022; Luo and Bayati, 2025; Kim, 2025). Absolute Thompson Sampling (Kitamura et al., 2026) retains $\widetilde{O}(d^{3/2}\sqrt{T})$ regret on fixed compact action sets.

Stochastic contexts and infrequent updates. For i.i.d. action sets, (Hanna et al., 2023a) reduce contextual to fixed-set linear bandits, obtaining $O(d\sqrt{T\log T})$ regret; their efficient batched method (Hanna et al., 2023b) achieves $\widetilde{O}(d^{3/2}\sqrt{T})$ regret with $O(\log\log T)$ batches. Extensions cover Markovian contexts (Buyukkalayci et al., 2026) and adversarial losses with i.i.d. action sets (van Erven et al., 2025); the latter gives $\widetilde{O}(d^2\sqrt{T})$ regret in polynomial time without a context simulator and improves the rate with simulator access. Under stochastic context assumptions,

Yu and Oh (Yu and Oh, 2026) achieve nearly minimax regret with $O(\log \log T)$ reward-dependent parameter updates, but require computation linear in the number of actions.

Gaussian width and experimental design. (Katz-Samuels et al., 2020) and RegretMED (Wagenmaker et al., 2021) use Gaussian width to design sampling allocations over fixed action sets for pure exploration and regret minimization, respectively. These methods optimize over sampling distributions, for which no efficient implementation is given. RegretMED provides an efficient implementation in the combinatorial semi-bandit setting: for example, when choosing a route, the learner observes each traversed edge's cost (semi-bandit) rather than the total route cost (bandit). Our method instead explores an action set filtered by logarithmically many halfspaces, using two linear optimizations along a random Gaussian direction scaled by the current design matrix. We show that this exploration provides expected information gain controlled by the same Gaussian width that bounds reward-comparison error, yielding nearly optimal regret for changing action sets, with polylogarithmically many linear optimizations per round.

Layering and quantization. We utilize a layering approach introduced in SupLinRel (Auer, 2002) and SupLinUCB (Chu et al., 2011), later refined in (Li et al., 2019; 2021). We also use one-bit reward quantization which appears in communication-efficient contextual linear bandits (Hanna et al., 2022); here, however, we use it to control the dependence of reward noise on past observations. Since a distribution on two fixed values is determined by its mean, one-bit reward quantization allows us to analyze an approximately equivalent ideal reward model whose conditional reward distribution depends only on the current action.

3 Problem Setup and Main Result

Notation. For $n \in \mathbb{N}$, let $[n] := \{1, \dots, n\}$. We write $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ for the Euclidean inner product and norm on $\mathbb{R}^d$, I_d for the identity matrix, and $\succeq$ for the Loewner order. For a symmetric positive definite $M \in \mathbb{R}^{d \times d}$, let $M^{-1/2} := (M^{1/2})^{-1}$ and $\|x\|_M := \sqrt{x^\top M x}$. All logarithms are natural unless subscripted, and $\mathcal{N}(0, I_d)$ denotes the standard Gaussian distribution on $\mathbb{R}^d$. Letters c and C denote positive universal constants whose values may change across occurrences.

Protocol. Fix a dimension $d \geq 1$ and a horizon $T \geq 2$, and let $\mathbb{B}^d := \{x \in \mathbb{R}^d : \|x\| \leq 1\}$. An unknown parameter $\theta \in \mathbb{B}^d$ and a sequence $A_1, \dots, A_T$ of subsets of $\mathbb{B}^d$ are fixed before the interaction begins. At each round $t = 1, \dots, T$, the learner observes A_t, selects an action $X_t \in A_t$, and observes a noisy reward $Y_t = \langle \theta, X_t \rangle + \eta_t$, where η_t is the noise, specified in Assumption 3.3.

Regret. Performance is measured by the pseudo-regret $R_T := \sum_{t=1}^{T} \Big(\max_{x \in A_t} \langle \theta, x \rangle - \langle \theta, X_t \rangle \Big)$, which is random through the actions $X_1, \dots, X_T$. Each maximum is attained by compactness. The learner knows d, T, a confidence parameter $\delta \in (0, 1/2)$, and the noise scale σ of Assumption 3.3.

Assumption 3.1 (Bounded parameter)**.** $\|\theta\| \leq 1$.

Assumption 3.2 (Oblivious bounded action sets)**.** Each A_t is a nonempty, compact, convex subset of $\mathbb{B}^d$. The sequence $(A_t)_{t \leq T}$ and parameter θ are deterministic and fixed before the interaction. This is the oblivious-adversary model: the instance may be chosen with knowledge of the algorithm's description, but not of the algorithm's realized randomness.

Noise model. Let $\mathcal{F}_t^{\text{full}}$ be the σ-algebra generated by everything realized before η_t, including the current action, past rewards, consumed learner randomness, and the environment's current private state. The environment may choose its next noise law using all this information, but cannot observe future learner draws. Appendix E gives a canonical construction used in the proofs.

Assumption 3.3 (Conditionally sub-Gaussian noise)**.** There is a known $\sigma \geq 0$ such that for every $t \in [T]$, almost surely, the conditional distribution of η_t given $\mathcal{F}_t^{\text{full}}$ is centered and σ-sub-Gaussian: $\mathbb{E}[\eta_t \mid \mathcal{F}_t^{\text{full}}] = 0$ and, for every $\lambda \in \mathbb{R}$, $\mathbb{E}[e^{\lambda \eta_t} \mid \mathcal{F}_t^{\text{full}}] \leq e^{\sigma^2 \lambda^2 / 2}$.

The noises need not be independent across rounds. Any known upper bound on their conditional sub-Gaussian scale may be used as σ. The boundedness and conditional sub-Gaussian noise requirements in Assumptions 3.1-3.3 are standard in linear bandits (Abbasi-Yadkori et al., 2011; Agrawal and Goyal, 2014; Abeille and Lazaric, 2017).

Oracle Access and Computational Model.

Assumption 3.4 (Linear-optimization oracle)**.** The learner accesses the available action set at time t only through the following oracle. A *query* consists of an objective $c \in \mathbb{R}^d$, and finitely many pairs

$(v_j, r_j) \in \mathbb{R}^d \times \mathbb{R}$, $j \in [m]$; letting

$$K \;:=\; A_t \cap \bigcap_{j=1}^{m} \{x \in \mathbb{R}^d : \langle v_j, x\rangle \geq r_j\},$$

the oracle returns a maximizer of $\langle c, x\rangle$ over K whenever $K \neq \emptyset$. The returned point is a measurable deterministic function of the query (with deterministic tie-breaking). The algorithm below issues only queries with nonempty feasible set. Each query counts as one *oracle call*.

We use an idealized real-arithmetic oracle model. Scalar arithmetic, comparisons, square roots, and logarithms have unit cost; exact Gaussian and uniform draws and exact matrix factorizations are available primitives, and dimension-d linear algebra has cost polynomial in d. Appendix D implements the constrained queries using ordinary linear optimization on A_t (without the additional linear constraints) and shows that sufficiently accurate feasible approximate maximizers preserve the regret guarantee up to a constant factor.

Main result. Define the confidence logarithm

$$\Lambda \;:=\; \log \frac{64\,T\left(2 + \lceil \log_2 T \rceil\right)}{\delta}\,. \tag{3.1}$$

Theorem 3.5 (Main result)**.** *Let $d \geq 1$, $T \geq 2$, $\sigma \geq 0$, $\delta \in (0, 1/2)$, and let Assumptions 3.1-3.4 hold. The algorithm of Section 4, run with inputs (d, T, σ, δ), has the following guarantees.*

(i) (Regret) *With probability at least $1 - \delta$ over the learner's internal randomness and the noise,*

$$R_T \;\leq\; C\,(1+\sigma)\,\Lambda\,\sqrt{\log(1 + T/d) + \Lambda/d}\;\cdot d\sqrt{T},$$

where C is a universal constant. In particular, $R_T \leq (1+\sigma)\, d\sqrt{T}\,\operatorname{polylog}(T/\delta)$.

(ii) (Oracle and computational complexity) *Deterministically, the algorithm uses at most $C \log T \cdot \log \frac{T \log T}{\delta}$ oracle calls per round, for a universal constant C; every query maximizes a linear function over A_t intersected with at most $\lceil \log_2 T \rceil$ halfspaces. Beyond oracle calls, the per-round computation consists of elementary linear-algebra operations of cost polynomial in d and $\log(T/\delta)$.*

Remark 3.6 (Optimality and oracle efficiency). For unit-variance Gaussian noise and $T \geq d^2$, the minimax expected regret is $\Omega(d\sqrt{T})$ already on the fixed Euclidean unit ball (Lattimore and Szepesvári, 2020, Theorem 24.2); see also (Dani et al., 2008). Taking $\delta = 1/T$ for $T \geq 3$ in Theorem 3.5 and using $R_T \leq 2T$ on the failure event gives matching dependence on d and T up to logarithms. The theorem obtains this rate with polylogarithmically many constrained linear optimizations per round.

4 Algorithm

GWidExplore operates in T rounds. In each round t, it examines the available action set A_t by processing it through a sequence of stages, called layers, indexed by $\ell = 0, \ldots, L$. Each layer has its own accuracy target ε_ℓ and maintains a separate estimator computed from the observations previously assigned to that layer in past rounds. The algorithm starts at layer 0 and processes the action set layer by layer. At each layer, it uses Gaussian width as a set-level measure of the uncertainty in comparing the surviving actions. If the width is small relative to the layer's accuracy target, the algorithm removes actions whose estimated rewards are too low by adding one linear constraint, and passes the remaining actions to the next layer within the same round. The width is recomputed at each layer and therefore need not decrease monotonically. If the width is large at some layer ℓ, the algorithm chooses an exploratory action, observes its reward, and records the observation for use in updating layer ℓ's estimator in subsequent rounds. The round then ends. If the algorithm reaches the final layer L and the width is still small, it selects the surviving action with the highest estimated reward. The round then ends without an exploratory observation or an update to any layer. Although the reward of this action is observed, it is not used in future updates. The next round starts again at layer 0, using the updated estimators. We next introduce the notation and parameters used by the algorithm, followed by its pseudocode and a more detailed description.

4.1 Notation and Parameters for the Algorithm

Definition 4.1 (Whitened width)**.** For a nonempty compact $K \subseteq \mathbb{R}^d$ and a symmetric positive definite $V \in \mathbb{R}^{d\times d}$, define

$$w_V(K) \;:=\; \mathbb{E}\Big[\max_{x\in K}\langle G, V^{-1/2}x\rangle - \min_{x\in K}\langle G, V^{-1/2}x\rangle\Big], \qquad G \sim N(0, I_d).$$

The maximum and minimum are attained by compactness. Equivalently, for the difference set $D := K - K = \{x - y : x, y \in K\}$,

$$w_V(K) = \mathbb{E}\Big[\max_{z\in D}\langle G, V^{-1/2}z\rangle\Big].$$

Thus $w_V(K)$ is the expected range of the random linear objective $x \mapsto \langle V^{-1/2}G, x\rangle$ over K. In the algorithm, V is the regularized design matrix, so the rescaling $V^{-1/2}$ emphasizes directions less informed by previous samples.

Parameters. Given the inputs (d, T, σ, δ) and Λ of (3.1), define

$$B := 1 + \sigma\sqrt{2\Lambda}, \qquad H := (1+B)\sqrt{\Lambda}, \qquad \mathcal{C}_T := d\log(1 + T/d) + \Lambda. \tag{4.1}$$

Here B is the clipping level, H converts width into reward-comparison error, and $\mathcal{C}_T$ controls design-matrix growth and confidence. Define

$$Q := H\sqrt{d\mathcal{C}_T}, \tag{4.2}$$

which determines the terminal accuracy and regret scale; in particular $Q = \widetilde{O}\big((1+\sigma)d\big)$. Next, we define the dyadic accuracy scales, the filter thresholds, and the number of layers as

$$\varepsilon_\ell := 2^{-\ell} \quad (\ell \geq 0), \qquad \tau_\ell := \frac{\varepsilon_\ell}{64\, C_\star H}, \qquad L := \left\lfloor \log_2 \frac{\sqrt{T}}{Q} \right\rfloor \quad \text{when } Q \leq \sqrt{T}\,. \tag{4.3}$$

Here ε_ℓ is the accuracy scale of layer ℓ, and τ_ℓ is a normalized version. When $Q > \sqrt{T}$ the bound of Theorem 3.5 trivially holds. When $Q \leq \sqrt{T}$, each width estimate is assigned failure probability

$$\alpha := \frac{\delta}{16\, T(L+1)}\,. \tag{4.4}$$

Lemma 4.2 (Parameter facts)**.** *$Q \geq 2\Lambda \geq 2$. If $Q \leq \sqrt{T}$, then L is well defined, $\frac{Q}{\sqrt{T}} \leq \varepsilon_L < \frac{2Q}{\sqrt{T}}$, $L + 1 \leq \frac{1}{2}\log_2 T$, and $\Lambda \geq \log\frac{64\,T(L+1)}{\delta}$.*

The proof is given in Appendix C.1.

Remark 4.3 (Constants). Every numerical constant in the algorithm is explicit except one: the threshold constant $C_\star$, fixed in Definition F.6 as an explicit function of two named universal constants (from the majorizing measure theorem and a generic chaining tail bound). Only an upper bound on $C_\star$ is ever needed; see Remark F.7. The algorithm can be implemented without knowing $C_\star$ by replacing it with $\log_2(T)$. This preserves the $\widetilde{O}(d\sqrt{T})$ regret guarantee: when $\log_2(T) \geq C_\star$, the existing analysis applies with an additional factor of $\log_2^2(T)$; otherwise, $T < 2^{C_\star}$ and the deterministic bound $R_T \leq 2T$ is absorbed into the universal constant.

Algorithm Description

We first handle a degenerate parameter regime. If $Q > \sqrt{T}$, the algorithm plays an arbitrary action in every round and ignores the observed reward. This already satisfies the regret guarantee because every instantaneous regret is at most 2, and hence $R_T \leq 2T < 2Q\sqrt{T}$. We therefore focus below on the main case $Q \leq \sqrt{T}$. The complete pseudocode is provided in Algorithm 1.

Layer states. Layer $\ell \in \{0, \ldots, L\}$ has accuracy target $\varepsilon_\ell = 2^{-\ell}$ and receives observations only from rounds in which it explores. Let $\mathcal{T}_\ell$ denote those rounds. Its regression estimate is

$$V_\ell = I_d + \sum_{s\in\mathcal{T}_\ell} X_s X_s^\top, \qquad b_\ell = \sum_{s\in\mathcal{T}_\ell} X_s Z_s, \qquad \hat{\theta}_\ell = V_\ell^{-1} b_\ell, \tag{4.5}$$

Algorithm 1 GWidExplore: Gaussian Width Exploration with layered elimination

Require: d, T, σ, δ; the constant $C_\star$ of Definition F.6

compute $\Lambda, B, H, \mathcal{C}_T, Q$ by (3.1), (4.1) and (4.2)
if $Q > \sqrt{T}$ **then**
    at every round t, play an arbitrary action of A_t (one oracle call with objective $c = 0$), observe and discard the reward Y_t; **stop**
end if
compute L, $(\varepsilon_\ell, \tau_\ell)_{\ell=0}^{L}$, α by (4.3)-(4.4); initialize $\mathcal{T}_\ell \leftarrow \emptyset$, $V_\ell \leftarrow I_d$, $b_\ell \leftarrow 0$, $\hat{\theta}_\ell \leftarrow 0$ for $0 \le \ell \le L$
for $t = 1, \dots, T$ **do**
    $K_{t,0} \leftarrow A_t$
    for $\ell = 0, 1, \dots, L$ **do**
        $\widehat{w}_{t,\ell} \leftarrow \textsc{WidthEstimate}(K_{t,\ell}, V_\ell, \alpha)$
        if $\widehat{w}_{t,\ell} > \tau_\ell$ **then** ▷ *explore* at layer ℓ
            draw a fresh $G_{t,\ell}^{\mathrm{end}} \sim N(0, I_d)$ and a fresh coin $\varsigma_{t,\ell} \sim \mathrm{Unif}\{+,-\}$
            $X^+ \leftarrow \arg\max_{x \in K_{t,\ell}} \langle V_\ell^{-1/2} G_{t,\ell}^{\mathrm{end}}, x\rangle$; $X^- \leftarrow \arg\max_{x \in K_{t,\ell}} \langle -V_\ell^{-1/2} G_{t,\ell}^{\mathrm{end}}, x\rangle$
            play $X_t \leftarrow X^{\varsigma_{t,\ell}}$; observe Y_t
            $Z_t \leftarrow \textsc{Quantize}_B(Y_t)$ using the fresh uniform $U_{t,\ell}$; erase Y_t
            $\mathcal{T}_\ell \leftarrow \mathcal{T}_\ell \cup \{t\}$; $V_\ell \leftarrow V_\ell + X_t X_t^\top$; $b_\ell \leftarrow b_\ell + X_t Z_t$; $\hat{\theta}_\ell \leftarrow V_\ell^{-1} b_\ell$
            end round (continue with round $t+1$)
        else ▷ *filter* at layer ℓ
            $a_{t,\ell} \leftarrow \arg\max_{x \in K_{t,\ell}} \langle \hat{\theta}_\ell, x\rangle$ ▷ one oracle call
            if $\ell < L$ **then**
                $K_{t,\ell+1} \leftarrow \{x \in K_{t,\ell} : \langle \hat{\theta}_\ell, x\rangle \ge \langle \hat{\theta}_\ell, a_{t,\ell}\rangle - 2\varepsilon_\ell\}$ ▷ append one halfspace
            else
                play $X_t \leftarrow a_{t,L}$; observe the reward Y_t and erase it (no update)
            end if
        end if
    end for
end for

where $Z_s \in \{-B, +B\}$ is the quantized reward defined below. Initially $\mathcal{T}_\ell = \emptyset$, $V_\ell = I_d$, and $b_\ell = \hat{\theta}_\ell = 0$. The matrix V_ℓ records the actions sampled by the layer, while b_ℓ also incorporates their quantized rewards. A second subscript denotes the state at the beginning of round t: $\mathcal{T}_{\ell,t} = \mathcal{T}_\ell \cap [t-1]$, with $V_{\ell,t}$, $b_{\ell,t}$, and $\hat{\theta}_{\ell,t}$ defined by (4.5) using $\mathcal{T}_{\ell,t}$. Because filtering changes no regression and an exploration update ends the round, these are also the values used whenever layer ℓ is examined during that round. All internal random draws used by the algorithm are mutually independent; Appendix D gives the formal randomness conventions.

A round of GWidExplore. Round t starts with $K_{t,0} = A_t$ and visits layers in order, from $\ell = 0$ to L. The set $K_{t,\ell}$ contains the actions retained by the preceding layers in this round. At a reached layer, the algorithm calls WIDTHESTIMATE to estimate $w_{V_\ell}(K_{t,\ell})$, then proceeds as follows.

1. *Explore if* $\widehat{w}_{t,\ell} > \tau_\ell$. Draw a fresh $G \sim N(0, I_d)$. Find the actions maximizing and minimizing the linear objective $x \longmapsto \langle V_\ell^{-1/2} G, x\rangle$ over $K_{t,\ell}$. An independent fair coin selects one of these two actions. Play it, quantize its observed reward, and add the resulting (action, response) pair to layer ℓ's regression. This update ends the round.

2. *Filter if* $\widehat{w}_{t,\ell} \le \tau_\ell$. Compute an empirical maximizer $a_{t,\ell} \in \arg\max_{x \in K_{t,\ell}} \langle \hat{\theta}_\ell, x\rangle$. If $\ell < L$, update
$$K_{t,\ell+1} = \{x \in K_{t,\ell} : \langle \hat{\theta}_\ell, x\rangle \ge \langle \hat{\theta}_\ell, a_{t,\ell}\rangle - 2\varepsilon_\ell\}$$
and continue to layer $\ell + 1$. This appends one linear constraint without playing an action or changing a regression. If $\ell = L$, play $a_{t,L}$ and discard its reward; no layer is updated.

Thus each round plays exactly one action and updates at most one layer. The surviving set is rebuilt from the newly available A_t each round, whereas the regression states persist across rounds.

Estimating the width. For fixed K and V, each independent Gaussian realization $G_{ij} \sim N(0, I_d)$ produces the range

$$F_{ij} = \max_{x \in K} \langle V^{-1/2} G_{ij}, x \rangle - \min_{x \in K} \langle V^{-1/2} G_{ij}, x \rangle.$$

This nonnegative quantity has expectation $w_V(K)$ and requires two linear-optimization calls. Algorithm 2 in Appendix A averages $m_0 = 128$ realizations in each of $g = 2\lceil \log_2(2/\alpha) \rceil + 1$ independent blocks and returns the median of the block averages. Appendix B.3 proves that its output satisfies

$$\tfrac{1}{2} w_V(K) \le \widehat{w} \le \tfrac{3}{2} w_V(K)$$

with probability at least $1 - \alpha$. One invocation uses exactly $2m_0 g = 256(2\lceil \log_2(2/\alpha) \rceil + 1)$ oracle calls. These realizations are used only for the width estimation; exploration uses a separate independent Gaussian realization.

Recording an exploratory reward. Algorithm 3 in Appendix A clips the observed reward Y to $C_B(Y) = \max\{-B, \min\{B, Y\}\}$ and then rounds it randomly to $-B$ or $+B$ so that $\mathbb{E}[Z \mid Y] = C_B(Y)$. Only Z enters the regression. The learner discards the raw reward and never uses it in later computations, although the environment may still use it when choosing future noise laws.

Remark 4.4 (Layering). Our confidence analysis uses a regression bound that assumes the errors in the quantized rewards are independent conditioned on the sampled actions. Layering is designed to satisfy this requirement. A layer's quantized rewards help choose actions for later layers, but are not used to choose the actions added to its own regression. In the ideal noise model described in Appendix B.1, this lets us condition on the sampled actions and surviving sets while the errors in the quantized rewards remain independent (Appendix B.2). We can then apply the regression bound to obtain confidence bounds in terms of Gaussian width.

Remark 4.5 (Quantization). Layering restricts how the learner uses rewards, but not how the environment generates them. Under our noise assumption, the environment may use past rewards from any layer to choose future noise distributions. Thus, layering alone does not give the conditional independence needed in the preceding remark. Quantization addresses this remaining dependence. A distribution on $\{-B, +B\}$ is determined entirely by its mean. Since clipping changes the conditional mean reward only slightly, the conditional distribution of each quantized reward is close to one that depends only on the current action only, rather than the full history. Appendix B.1 uses this fact to construct an ideal process that agrees with the learner's actions and retained quantized rewards throughout the interaction with high probability. We apply the layerwise independence argument in that process and transfer the resulting guarantee to the original noise model.

Feasibility and computation. Each filter retains its empirical maximizer, so the surviving sets, and hence all oracle queries, remain feasible. By Lemma 4.2, each query has at most $\lceil \log_2 T \rceil$ additional constraints. Appendix B.7 gives the full oracle count; the remaining computation is polynomial-time linear algebra in d.

5 Analysis Overview

We focus on the main case $Q \le \sqrt{T}$. We first analyze an ideal quantized-reward model in which, given action X_t, the learner receives $Z_t \in \{-B, +B\}$ with mean $\langle \theta, X_t \rangle$, where Z_t is independent on the history given X_t. Its error is centered and B-sub-Gaussian, where $B = \widetilde{O}(1)$. The proof connects two properties of Gaussian width: it controls the error in reward comparisons and lower-bounds the progress made by exploration. These facts lead to the regret bound and support efficient width estimation. We suppress universal constants, logarithmic factors in T/δ, and dependence on σ, keeping only the dependence on d and T explicit. The notation $\lesssim$ indicates inequality up to constant factor. We return at the end of the section to the original history-dependent noise model.

Gaussian width controls reward-comparison error. Adapting the Gaussian-width argument of (Katz-Samuels et al., 2020) to ridge regression, we show that the width controls both the noise and regularization-bias terms. Fix a layer and round, and write K, V, and $\hat{\theta}$ for the surviving action set, design matrix, and regression estimate. Recall that

$$w = w_V(K) = \mathbb{E}_G \left[\max_{x \in K} \langle V^{-1/2} G, x \rangle - \min_{x \in K} \langle V^{-1/2} G, x \rangle \right], \qquad G \sim N(0, I_d).$$

In the ideal model, we may condition on this layer's sampled actions and surviving set while its quantized-reward errors $\xi_s := Z_s - \langle \theta, X_s \rangle$ remain independent, centered, and B-sub-Gaussian. We work under this conditioning below.

Writing $S_\xi = \sum_{s \in \mathcal{T}_{\ell,t}} X_s \xi_s$, the ridge regression error decomposes as

$$\hat{\theta} - \theta = \underbrace{V^{-1} S_\xi}_{\text{rewards noise}} - \underbrace{V^{-1}\theta}_{\text{regularization bias}}, \qquad V = I_d + \sum_{s \in \mathcal{T}_{\ell,t}} X_s X_s^\top .$$

We bound both terms over action differences $z \in D := K - K$. Let $r = \max_{z \in D} \|z\|_{V^{-1}}$. Since D is symmetric, a maximizer z_* and its negative give

$$w = \mathbb{E}_G \sup_{z \in D} \langle G, V^{-1/2} z \rangle \geq \mathbb{E}_G |\langle G, V^{-1/2} z_* \rangle| = \sqrt{2/\pi}\, r.$$

Thus, using $V \succeq I_d$ and $\|\theta\| \leq 1$, the bias satisfies

$$\sup_{z \in D} |\langle z, V^{-1}\theta \rangle| \leq r \|\theta\|_{V^{-1}} \leq r \lesssim w.$$

For the random term, define $A_z = \langle z, V^{-1} S_\xi \rangle$. Independence and sub-Gaussianity imply, for every $z, z' \in D$, using $\sum_s X_s X_s^\top = V - I_d \preceq V$

$$\mathbb{E} e^{\lambda (A_z - A_{z'})} \leq \exp\left(\frac{\lambda^2 B^2}{2} \sum_{s \in \mathcal{T}_{\ell,t}} \left((z - z')^\top V^{-1} X_s \right)^2 \right) \leq \exp\left(\frac{\lambda^2 B^2}{2} \|z - z'\|_{V^{-1}}^2 \right).$$

The Gaussian process $z \mapsto B\langle G, V^{-1/2} z \rangle$ has exactly this increment metric and expected supremum Bw. Following the Gaussian-width concentration argument of Katz-Samuels et al. (Katz-Samuels et al., 2020, Appendix L), generic chaining and the majorizing measure theorem (Dirksen, 2015, Theorem 3.2 and Remark 3.3(i)) therefore give, with probability at least $1 - \alpha$,

$$\sup_{z \in D} |A_z| \lesssim B\big(w + r\sqrt{\log(2/\alpha)}\big) = \widetilde{O}(w),$$

where the last step uses $r \lesssim w$. Combining the random and bias terms, then taking a union bound over the at most $T(L+1)$ layer–round pairs, yields with high probability

$$\sup_{x,y \in K} |\langle \hat{\theta} - \theta, x - y \rangle| \leq \widetilde{O}\big(w_V(K)\big) \tag{5.1}$$

simultaneously at every reached layer and round. Appendix B.4 gives the detailed argument.

Accurate comparisons make exploration inexpensive. Layer ℓ targets accuracy $\varepsilon_\ell = 2^{-\ell}$. With probability at least $1 - \delta/16$, all width estimates satisfy $\frac{1}{2} w \leq \widehat{w} \leq \frac{3}{2} w$ (Lemma B.16). Thus, whenever layer ℓ filters, $\widehat{w} \leq \tau_\ell = \varepsilon_\ell / (64 C_\star H)$ and hence $w \leq 2\tau_\ell$. The confidence bound gives

$$\sup_{x,y \in K} |\langle \hat{\theta} - \theta, x - y \rangle| \leq C_\star H w \leq \varepsilon_\ell / 32.$$

Consequently, an optimal action in K survives the $2\varepsilon_\ell$ filter, and every retained action has true reward gap less than $3\varepsilon_\ell$. Since $K_{t,0} = A_t$, an optimal action therefore survives every filter.

Exploration at layer $\ell \geq 1$ incurs regret at most $3\varepsilon_{\ell-1} = 6\varepsilon_\ell$, while at layer 0 the regret is at most $2 = 2\varepsilon_0$. If all layers filter, the final selected action has regret at most $\varepsilon_L/32$. Hence, writing N_ℓ for the number of explorations at layer ℓ,

$$R_T \lesssim \sum_{\ell=0}^{L} \varepsilon_\ell N_\ell + T\varepsilon_L. \tag{5.2}$$

It remains to bound N_ℓ.

Large width forces progress from exploration. Fix the surviving set K, the design V, and the outcome of the width test before an exploration. For a fresh Gaussian G, let X^+ and X^- maximize and minimize $\langle V^{-1/2} G, x \rangle$ over K, and let X be the endpoint selected by the independent fair coin.

Adding X to the regression increases $\log\det V$ by $\log(1+\|X\|^2_{V^{-1}})$. Thus $\|X\|^2_{V^{-1}}$ measures the progress of this update. Cauchy-Schwarz gives

$$w^2 = \Big(\mathbb{E}\langle G, V^{-1/2}(X^+ - X^-)\rangle\Big)^2 \\ \le d\,\mathbb{E}\|X^+ - X^-\|^2_{V^{-1}} \le 2d\,\mathbb{E}\big[\|X^+\|^2_{V^{-1}} + \|X^-\|^2_{V^{-1}}\big] = 4d\,\mathbb{E}\|X\|^2_{V^{-1}}.$$

The expectations average over the Gaussian direction and endpoint coin. Since $u = \|X\|^2_{V^{-1}} \le 1$, we have $\log(1+u) \ge u/2$, so the expected log-determinant increase is at least ε_ℓ^2/d up to a constant factor. A standard trace-determinant bound limits each layer's total log-determinant increase to $d\log(1+T/d)$. Standard martingale concentration converts this conditional expected progress into the simultaneous high-probability bound

$$N_\ell \le \frac{d\log(1+T/d)}{\varepsilon_\ell^2/d} = \widetilde{O}\left(\frac{d^2}{\varepsilon_\ell^2}\right) \qquad (0 \le \ell \le L).$$

Appendix B.6 proves this bound, accounting for width-estimation failures.

Combining the regret. Substituting the exploration counts into (5.2), the geometric sum of $1/\varepsilon_\ell$ is dominated by its final term, giving

$$R_T \le \widetilde{O}\left(\frac{d^2}{\varepsilon_L}\right) + O(T\varepsilon_L) = \widetilde{O}\big(d\sqrt{T}\big).$$

The terminal accuracy specified in Section 4.1 balances these two contributions, at scale $d/\sqrt{T}$ up to logarithmic factors.

Estimating width with logarithmically many oracle calls. Each Gaussian projection range has mean w and requires two linear optimizations. Gaussian Poincaré and $r \lesssim w$ bound its variance by $O(w^2)$, so averaging a constant number of independent ranges estimates w within a constant factor with probability at least $3/4$. Taking the median of independent block averages reduces the failure probability exponentially in the number of blocks, achieving failure probability α with $O(\log(1/\alpha))$ oracle calls (Appendix B.3). Each round visits $O(\log T)$ layers, yielding the stated per-round oracle bound. Each filter adds one halfspace and retains its empirical maximizer, so every oracle query remains feasible. Median-of-means is a standard method for obtaining high-probability mean estimates under finite variance assumptions (Hsu and Sabato, 2014). Katz-Samuels et al. (Katz-Samuels et al., 2020) use it to estimate a gap-normalized Gaussian supremum for experimental design in pure exploration. Here, we use it to estimate the unnormalized width w.

Transferring the guarantee to history-dependent noise. The actual noise distribution may depend on the entire past. Quantization reduces the effect of this dependence because a distribution on $\{-B, +B\}$ is determined by its mean. Randomized rounding preserves the clipped reward's mean, and the sub-Gaussian tail bound makes clipping bias small. Hence, conditional on the past, the actual quantized-reward distribution is close to the ideal distribution. Appendix B.1 couples the actual and ideal runs so that all actions, and hence their regrets, agree with high probability, transferring the ideal regret guarantee to the original noise model.

The dimension saving. Standard optimism and elimination based algorithms, rely on maximizing $\|x-y\|_{V^{-1}}$ over x, y in a modified action set to compute the exploration distribution. This yields a nearly optimal regret, however, that optimization problem is non-convex; approximating it with linear oracle solutions introduces a factor of $\sqrt{d}$. Our exploration rule maximizes and minimizes a Gaussian linear objective over K, then samples uniformly from the two resulting actions. Together with the fact that Gaussian-width bounds the reward comparison error, this gives two properties: (i) sufficient exploration progress: $\mathbb{E}\|X\|^2_{V^{-1}} \ge w^2/(4d)$, so exploration at accuracy $\varepsilon_\ell = \widetilde{O}(w)$ guarantees expected progress $\widetilde{\Omega}(\varepsilon_\ell^2/d)$, matching the progress rate of standard elimination based algorithms; and (ii) exploration requires two linear optimizations, and w can be estimated efficiently within a constant factor.

AI USE STATEMENT

This problem originated in our work approximately two years ago. At that time, we invested substantial effort but did not obtain a complete proof. We later used ChatGPT 5.6 Sol through Codex as an interactive aid in developing the algorithm and completing the regret proof. This was not a one-shot process: over approximately 20-30 interactive turns, the authors identified errors in successive proof attempts and prompted the system to correct them; this error-identification and revision process allowed us to obtain the presented final proof. The authors independently reviewed and verified every step of the argument. We also used AI to identify related papers, improve the exposition, and assist with proofreading. The authors verified the literature claims and reviewed all AI-assisted revisions. We take responsibility for the final content of the paper.

REFERENCES

Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24*, 2011.

M. Abeille and A. Lazaric. Linear Thompson Sampling Revisited. In *International Conference on Artificial Intelligence and Statistics*, PMLR 54, pages 176–184, 2017.

M. Abeille, D. Janz, and C. Pike-Burke. When and why randomised exploration works (in linear bandits). In *International Conference on Algorithmic Learning Theory*, PMLR 272, pages 4–22, 2025.

S. Agrawal and N. Goyal. Thompson Sampling for Contextual Bandits with Linear Payoffs. In *International Conference on Machine Learning*, PMLR 28(3), pages 127–135, 2013.

S. Agrawal and N. Goyal. Thompson Sampling for Contextual Bandits with Linear Payoffs. Expanded version, arXiv:1209.3352v4, February 2014.

P. Auer. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

B. Awerbuch and R. D. Kleinberg. Online Linear Optimization and Adaptive Routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008.

J. A. Ayala-Romero, A. Garcia-Saavedra, and X. Costa-Perez. Risk-Aware Continuous Control with Neural Contextual Bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):20930–20938, 2024.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

S. Bubeck, N. Cesa-Bianchi, and S. M. Kakade. Towards Minimax Policies for Online Linear Optimization with Bandit Feedback. In *Conference on Learning Theory*, PMLR 23, pages 41.1–41.14, 2012.

K. Buyukkalayci, O. Hanna, and C. Fragouli. Mixing Makes Markovian Contexts Cheap for Linear Bandits. arXiv:2603.12530v2, June 2026.

W. Chu, L. Li, L. Reyzin, and R. E. Schapire. Contextual Bandits with Linear Payoff Functions. In *International Conference on Artificial Intelligence and Statistics*, PMLR 15, pages 208–214, 2011.

V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pages 355–366, 2008.

S. Dirksen. Tail bounds via generic chaining. *Electronic Journal of Probability*, 20:no. 53, 1–29, 2015.

M. Grötschel, L. Lovász, and A. Schrijver. Geometric methods in combinatorial optimization. In W. R. Pulleyblank, editor, *Progress in Combinatorial Optimization*, pages 167–183. Academic Press, 1984.

O. A. Hanna, L. F. Yang, and C. Fragouli. Learning from Distributed Users in Contextual Linear Bandits Without Sharing the Context. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

O. A. Hanna, L. F. Yang, and C. Fragouli. Contexts can be Cheap: Solving Stochastic Contextual Bandits with Linear Bandit Algorithms. In *Conference on Learning Theory*, PMLR 195, pages 1791–1821, 2023.

O. A. Hanna, L. F. Yang, and C. Fragouli. Efficient Batched Algorithm for Contextual Linear Bandits with Large Action Space via Soft Elimination. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

P. Harsha, C. Subramanian, N. Abe, S. Subramanian, A. Ba, K. A. Fernández Román, M. Longinos Garrido, and C. Narayanaswami. Practical Contextual Bandits for Large-Scale Structured Discrete Constrained Optimization Problems. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 850–859, 2025.

E. Hazan, Z. Karnin, and R. Meka. Volumetric Spanners and their Applications to Machine Learning. arXiv preprint arXiv:1312.6214v1, 2013.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

D. Hsu and S. Sabato. Heavy-tailed regression with a generalized median-of-means. In *Proceedings of the 31st International Conference on Machine Learning*, PMLR 32(2), pages 37–45, 2014.

S. Ito, D. Hatano, H. Sumita, K. Takemura, T. Fukunaga, N. Kakimura, and K.-I. Kawarabayashi. Oracle-Efficient Algorithms for Online Linear Optimization with Bandit Feedback. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

O. Kallenberg. *Foundations of Modern Probability*. 2nd edition, Springer, 2002.

J. Katz-Samuels, L. Jain, Z. Karnin, and K. Jamieson. An Empirical Process Approach to the Union Bound: Practical Algorithms for Combinatorial and Linear Bandits. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

B. Kim and A. Tewari. Randomized Exploration for Non-Stationary Stochastic Linear Bandits. In *Conference on Uncertainty in Artificial Intelligence*, PMLR 124, pages 71–80, 2020. Corrected version: arXiv:1912.05695v5, October 2021.

W. Kim. Adaptive Data Augmentation for Thompson Sampling. arXiv:2506.14479, 2025.

T. Kitamura, S. Liu, and C. Szepesvári. Randomized Exploration for Linear Bandits via Absolute Perturbations. arXiv:2606.28616, 2026.

T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

L. Li, W. Chu, J. Langford, and R. E. Schapire. A Contextual-Bandit Approach to Personalized News Article Recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670, 2010.

Y. Li, Y. Wang, and Y. Zhou. Nearly Minimax-Optimal Regret for Linearly Parameterized Bandits. In *Conference on Learning Theory*, PMLR 99, pages 2173–2174, 2019.

Y. Li, Y. Wang, X. Chen, and Y. Zhou. Tight Regret Bounds for Infinite-armed Linear Contextual Bandits. In *International Conference on Artificial Intelligence and Statistics*, PMLR 130, pages 370–378, 2021.

Y. Luo and M. Bayati. Geometry-Aware Approaches for Balancing Performance and Theoretical Guarantees in Linear Bandits. In *International Conference on Learning Representations*, 2025.

M. Talagrand. *Upper and Lower Bounds for Stochastic Processes: Modern Methods and Classical Problems*. Springer, 2014.

T. van Erven, J. Mayo, J. Olkhovskaya, and C.-Y. Wei. An Improved Algorithm for Adversarial Linear Contextual Bandits via Reduction. In *Advances in Neural Information Processing Systems*, volume 38, 2025. Revised version: arXiv:2508.11931v3, June 2026.

A. Wagenmaker, J. Katz-Samuels, and K. Jamieson. Experimental Design for Regret Minimization in Linear Bandits. In *International Conference on Artificial Intelligence and Statistics*, PMLR 130, pages 3088–3096, 2021.

S. Yu and M.-H. Oh. Practical and Optimal Algorithm for Linear Contextual Bandits with Rare Parameter Updates. arXiv:2606.00984v3, August 2026.

L. Zhang, T. Yang, R. Jin, Y. Xiao, and Z.-H. Zhou. Online Stochastic Linear Optimization under One-bit Feedback. In *International Conference on Machine Learning*, PMLR 48, pages 392–401, 2016.

T. Zhang. Feel-Good Thompson Sampling for Contextual Bandits and Reinforcement Learning. *SIAM Journal on Mathematics of Data Science*, 4(2):834–857, 2022.

Y. Zhu, D. J. Foster, J. Langford, and P. Mineiro. Contextual Bandits with Large Action Spaces: Made Practical. In *International Conference on Machine Learning*, PMLR 162, pages 27428–27453, 2022.