

Technical Appendices and Supplementary Material

A Preliminaries

Let $\mathbb{N} = \{1, 2, 3, \dots\}$. For $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, 2, \dots, n\}$ and assume throughout that the set of actions of players both in MABs and games is $[n]$. For a discrete set Ω , we denote by $\Delta(\Omega)$ the set of probability distributions over Ω . We often identify a distribution $p \in \Delta([n])$ with the vector $p = (p_1, \dots, p_n)$ such that $p_i = p(i) = \mathbb{P}_{x \sim p}[x = i]$. Finally, we define the notion of a smooth distribution:

Definition A.1. Let $n \in \mathbb{N}$ and $\sigma \in [1/n, 1]$. A probability distribution $p \in \Delta([n])$ is called σ -smooth if for every $i \in [n]$, $p_i \leq \sigma$.

The degree of smoothness of a distribution is governed by the parameter σ . For $\sigma = 1$, smoothness is vacuous as every probability distribution is 1-smooth. On the other extreme when $\sigma = 1/n$ the distribution is the smoothest possible: the uniform distribution.

A.1 Cryptographic preliminaries

Let $\lambda \in \mathbb{N}$ denote the security parameter. We write p.p.t. to mean probabilistic polynomial time. We let $\text{negl}(\lambda)$ denote a function that is $O(1/\lambda^c)$ for all $c > 0$.

A.1.1 Vector commitments

A *vector commitment* [Catalano and Fiore, 2013] is a tuple of p.p.t. algorithms:

- $\text{KeyGen}(1^\lambda, n) \rightarrow \text{pp}$: takes as input the security parameter and size n of the vectors to be committed, and outputs public parameters pp .
- $\text{Commit}_{\text{pp}}(v) \rightarrow c_v, \text{aux}$: takes as input a length- n vector v , and outputs a commitment c_v and auxiliary information aux . aux often contains the entire committed vector v .
- $\text{Open}_{\text{pp}}(v_i, i, \text{aux}) \rightarrow \text{pf}$: takes as input a value v_i , an index i , and auxiliary information aux . It outputs a proof pf that v_i is the i^{th} component of v corresponding to aux .
- $\text{Verify}_{\text{pp}}(c_v, v_i, i, \text{pf}) \rightarrow \{\text{accept}, \text{reject}\}$: takes as input a commitment c_v , a value v_i , an index i , and a proof pf . It accepts if and only if c_v commits to a vector whose i^{th} component is v_i (except for events with negligible probability).

Vector commitments must satisfy *correctness* and *position binding*. Correctness requires that with overwhelming probability, any honestly generated public parameters and honestly committed vectors yield valid opening proofs for all of their components. Position binding requires that it is infeasible for any non-uniform p.p.t. adversary to produce a commitment and two valid proofs for *different* openings of that commitment. More precisely, for all $n \in \mathbb{Z}^+$ and all p.p.t. adversaries \mathcal{A} ,

$$\mathbb{P}_{\text{pp} \leftarrow \text{KeyGen}(1^\lambda, n)} \left[\begin{array}{l} \text{accept} \leftarrow \text{Verify}_{\text{pp}}(c, v_i, i, \text{pf}) \\ \wedge \text{accept} \leftarrow \text{Verify}_{\text{pp}}(c, v'_i, i, \text{pf}') \end{array} \middle| (c, i, v_i, v'_i, \text{pf}, \text{pf}') \leftarrow \mathcal{A}(1^\lambda, n) \right] \leq \text{negl}(\lambda).$$

We refer the reader to Catalano and Fiore [2013] for further details.

A.1.2 Succinct non-interactive arguments of knowledge

We present a simplified description of *succinct non-interactive arguments of knowledge* (SNARKs) and refer the reader to Groth [2016] for full details.

A succinct non-interactive argument of knowledge for a relation generator \mathcal{R} is a tuple of p.p.t. algorithms:

- $\text{Setup}(1^\lambda, R) \rightarrow \text{pp}, \tau$: takes as input the security parameter and a relation $R \in \mathcal{R}$, and outputs public parameters pp and a simulation trapdoor τ .
- $\text{Prove}(R, \text{pp}, \phi, w) \rightarrow \text{pf}$: takes as input a relation R , public parameters pp , and a statement-witness pair $(\phi, w) \in R$. It outputs a proof pf of this pair's membership in the relation.

- $\text{Verify}(R, \text{pp}, \phi, \text{pf}) \rightarrow \{\text{accept}, \text{reject}\}$: takes as input a relation R , public parameters pp , a statement ϕ , and a proof pf . It should accept if and only if ϕ has a witness w such that $(\phi, w) \in R$.

We consider SNARKs that satisfy *perfect completeness* and *computational knowledge soundness*.

Perfect completeness requires that for all $\lambda \in \mathbb{N}$, $R \in \mathcal{R}$, and $(\phi, w) \in R$:

$$\mathbb{P}_{(\text{pp}, \tau) \leftarrow \text{Setup}(1^\lambda, R)}[\text{pf} \leftarrow \text{Prove}(R, \text{pp}, \phi, w) : \text{accept} \leftarrow \text{Verify}(R, \text{pp}, \phi, \text{pf})] = 1.$$

Computational knowledge soundness requires that there exists a non-uniform p.p.t. extractor that can extract a witness whenever an adversary can compute an accepting proof. That is, for all non-uniform adversaries \mathcal{A} , there exists a non-uniform p.p.t. extractor $\mathcal{X}_{\mathcal{A}}$ such that

$$\mathbb{P} \left[\begin{array}{l} (\phi, w) \notin R \text{ and} \\ \text{Verify}(R, \text{pp}, \phi, \text{pf}) \rightarrow \text{accept} \end{array} \middle| \begin{array}{l} (R, z) \leftarrow \mathcal{R}(1^\lambda) \\ (\phi, \tau) \leftarrow \text{Setup}(1^\lambda) \\ ((\phi, w), \text{pf}) \leftarrow (\mathcal{A} \parallel \mathcal{X}_{\mathcal{A}})(R, z, \text{pp}) \end{array} \right] \leq \text{negl}(\lambda),$$

where $(\mathcal{A} \parallel \mathcal{X}_{\mathcal{A}})$ denotes that the extractor has access to the adversary's internal state and randomness.

B Bandits

B.1 Definitions

Definition B.1 (Bandit). Let $n \in \mathbb{N}$. An *n -arm bandit* is a vector of n distributions $q = (q_1, \dots, q_n) \in (\Delta([0, 1]))^n$.

A bandit defines a bandit oracle such that, given a query $i \in [n]$ (corresponding to “pulling the i -th arm of the bandit”), the oracle returns a utility $x \sim q_i$ sampled independently of all previous oracle queries and responses.

The *expected utilities vector* of q is a vector $u = \text{utility}(q) \in [0, 1]^n$ such that $u_i = \mathbb{E}_{x \sim q_i}[x]$ for all $i \in [n]$. A *strategy for an n -arm bandit* is a distribution $\pi = (\pi_1, \dots, \pi_n) \in \Delta([n])$. The *expected utility of π with respect to u* is $\mathbb{E}_{i \sim \pi, x \sim q_i}[x] = \sum_{i=1}^n \pi_i u_i = \pi \cdot u$.

Definition B.2 (Smooth bandit strategy). Let $n \in \mathbb{N}$ and $\sigma \in [1/n, 1]$. A strategy $\pi \in \Delta([n])$ for an n -arm bandit is σ -smooth if $\pi_i \leq \sigma$ for all $i \in [n]$.

Definition B.3 (Optimal smooth bandit strategy). Let $n \in \mathbb{N}$, $\varepsilon \geq 0$, $\sigma \in [1/n, 1]$, let $u \in [0, 1]^n$ be the expected utilities vector of an n -arm bandit, and let $\pi \in \Delta([n])$ be a strategy. We say that π is ε -competitive with respect to σ -smooth policies for u , if for every σ -smooth strategy $\pi' \in \Delta([n])$,

$$\pi' \cdot u - \pi \cdot u \leq \varepsilon.$$

If in addition π is σ -smooth, then we say that π is an ε -optimal σ -smooth strategy for u .¹²

Definition B.4 (Verification of optimality for smooth bandit strategies). An *interactive proof system for verification of ε -optimal σ -smooth policies for n -arm bandits* is a pair of algorithms (V, P) such that for all $n \in \mathbb{N}$, and for every n -arm bandit q with expected utilities vector $u = \text{utility}(q) \in [0, 1]^n$ and bandit oracle \mathcal{O}_q , and for all $\sigma \in [1/n, 1]$ and $\varepsilon \in (0, 1)$, the following two conditions hold:

- **Completeness.** Let the random variable

$$\pi_V = [V^{\mathcal{O}_q}(n, \varepsilon, \sigma), P^{\mathcal{O}_q}(n, \varepsilon, \sigma)] \in \Delta([n]) \cup \{\text{reject}\}$$

denote the output of V after interacting with P , when each of them receives (n, ε, σ) as input and has oracle access to \mathcal{O}_q . Then

$$\mathbb{P}[(\pi_V \neq \text{reject}) \wedge (\forall \sigma\text{-smooth } \pi' \in \Delta([n]) : \pi' \cdot u - \pi_V \cdot u \leq \varepsilon)] \geq \frac{2}{3}.$$

¹ These are special cases of definitions in Daskalakis et al. [2024] (see Definition C.3 below). The first definition corresponds to a weak ε -approximate σ' -smooth Nash equilibrium for u , and the second definition corresponds to a strong ε -approximate σ' -smooth Nash equilibrium for u , where $\sigma' = 1/(n\sigma)$.

² We neglect specifying the approximation parameter ε when it is 0, and speak simply of strategies that are optimal σ -smooth, or competitive with σ -smooth strategies.

- **Soundness.** For any (possibly malicious and computationally unbounded) prover P' (which in particular may depend on n, ε, σ and q), the verifier's output $\pi_V = [V^{\mathcal{O}_q}(n, \varepsilon, \sigma), P'] \in \Delta([n]) \cup \{\text{reject}\}$ satisfies

$$\mathbb{P}[(\pi_V = \text{reject}) \vee (\forall \sigma\text{-smooth } \pi' \in \Delta([n]) : \pi' \cdot u - \pi_V \cdot u \leq \varepsilon)] \geq \frac{2}{3}.$$

In both conditions, the probability is over the randomness of \mathcal{O}_q and V , as well as P or P' .

We also consider a related notion of verification, where the prover convinces the verifier that the optimal σ -smooth policy has expected reward in some interval $[t - \varepsilon, t + \varepsilon]$. In this protocol, the verifier *does not necessarily learn* this policy. This allows for very low communication between the prover and verifier.

B.2 Protocols for verifying smooth bandit strategies

Theorem B.5 (Verification for bandits). *Let $n \in \mathbb{N}$, let $\sigma \in [1/n, 1]$, let $\varepsilon \in (0, 1)$. Protocol 1 defines an interactive proof system (V, P) for verification of ε -optimal σ -smooth policies for n -arm bandits such that:*

- The protocol consists of a single message of $O(n \log(1/\varepsilon))$ bits sent from P to V .
- P performs $m_P = O(n \log(n/\varepsilon)/\varepsilon^2)$ nonadaptive queries to the bandit oracle and runs in time $\text{poly}(n, 1/\varepsilon)$.
- V performs

$$m_V = O\left(\frac{n\sigma}{\varepsilon^2} \cdot \log\left(\frac{n\sigma}{\varepsilon}\right) \log\left(\frac{1}{\varepsilon}\right)\right)$$

nonadaptive queries to the bandit oracle, and runs in time $\text{poly}(n, 1/\varepsilon)$.

In particular, if $\sigma = \Theta(1/\sqrt{n})$ then $m_V = \tilde{O}(\sqrt{n})$, and if $\sigma = \Theta(1/n)$ then m_V is independent of n .

The proof of Theorem B.5 appears in Section D.1.

B.2.1 A protocol variant for verifying optimality of a given strategy

Lemma B.6. *Let $n \in \mathbb{N}$, let $\eta, \varepsilon \in (0, 1)$, and let q be an n -armed bandit. There exists a protocol (Protocol 3) consisting of a prover P and a verifier V , both of whom are allowed oracle access to q and given $n, \eta, \varepsilon, \pi$ as input. The protocol satisfies the following:*

- **Completeness:** If π is an ε -optimal σ -smooth policy for q , the verifier V outputs 1 with probability at least $1 - \delta$.
- **Soundness:** If π is not an $(\varepsilon + \eta)$ -optimal σ -smooth policy for q , the verifier V outputs 0 with probability at least $1 - \delta$.

In either case:

- The protocol consists of $O(n \log(1/\eta) \log(1/\delta))$ bits sent from P to V .
- P performs $O(n(\log(n/\eta)/\eta^2) \log(1/\delta))$ nonadaptive queries to the bandit oracle and runs in time $\text{poly}(n, 1/\eta, 1/\delta)$.
- V performs

$$O\left(\frac{n\sigma}{\eta^2} \cdot \log\left(\frac{n\sigma}{\eta}\right) \log\left(\frac{1}{\eta}\right) \log\left(\frac{1}{\delta}\right)\right)$$

nonadaptive queries to the bandit oracle, and runs in time $\text{poly}(n, 1/\eta, 1/\delta)$.

Protocol 3 and the proof for Lemma B.6 appear in Section D.2.

Assumptions:

- $n \in \mathbb{N}; \sigma \in [1/n, 1]; \varepsilon > 0$.
- $q = (q_1, \dots, q_n) \in (\Delta([0, 1]))^n$ is an n -arm bandit.
- $p_P = \lceil 128 \ln(12n/\varepsilon)/\varepsilon^2 \rceil$.
- $p_V(b) = \lceil 128 \ln(12 \cdot (\log_4(1/\varepsilon) + 2) \cdot a(b))/(4^b \varepsilon)^2 \rceil$.
- $a(b) = \lceil 4^b \varepsilon \cdot 4n\sigma \cdot (\log_4(1/\varepsilon) + 2) \cdot \ln(6)/\varepsilon \rceil$.

PROVER(n, ε):

```

for  $i \in [n]$ 
  for  $j \in [p_P]$ 
    sample  $r_{i,j} \sim q_i$ 
   $\tilde{u}_i \leftarrow \frac{1}{k} \sum_{j=1}^k r_{i,j}$ 
 $\tilde{u} \leftarrow (\tilde{u}_1, \dots, \tilde{u}_n)$ 
send  $\tilde{u}$  to verifier

```

VERIFIER(n, ε, σ):

```

receive  $\tilde{u}$  from prover
for  $b \in \{0, 1, 2, \dots, \lceil \log_4(1/\varepsilon) \rceil\}$ :
   $\varepsilon_b \leftarrow \varepsilon \cdot 4^b$ 
   $a_b \leftarrow a(b)$ 
   $p_b \leftarrow p_V(b)$ 
  for  $t \in [a_b]$ :
    sample  $i_{b,t} \sim U([n])$ 
    for  $j \in [p_b]$ :
      sample  $r_{b,t,j} \sim q_{i_{b,t}}$ 
       $\hat{u}_{i_{b,t}} \leftarrow \frac{1}{p_b} \sum_{j=1}^{p_b} r_{b,t,j}$ 
    if  $|\tilde{u}_{i_{b,t}} - \hat{u}_{i_{b,t}}| > \varepsilon_b/8$ :
      reject and terminate execution
   $\pi_V \leftarrow \text{COMPUTEOPTIMALSMOOTHBANDITSTRATEGY}(n, \sigma, \tilde{u})$ 
output  $\pi_V$ 

```

- ▷ Iterate over all bins.
- ▷ Bin b contains lies of magnitude $|\tilde{u}_i - u_i| \in (\varepsilon \cdot 4^{b-1}, \varepsilon \cdot 4^b]$.
- ▷ Number of bandit arms to pull.
- ▷ Number of pulls to each bandit arm.
- ▷ Select a bandit arm at random.
- ▷ Pull (query) the bandit arm.
- ▷ Estimate the bandit arm's utility.
- ▷ Reject if prover's purported utility is far from estimated utility.

▷ Compute optimal strategy for prover's purported utilities using Algorithm 1.

Protocol 1: A verification protocol for bandits, satisfying the requirements of Theorem B.5.

B.2.2 A low-communication protocol variant

We observe that using cryptographic tools, specifically succinct non-interactive arguments of knowledge (SNARKs) and vector commitments (VCs), one can significantly reduce the communication between the prover and the verifier in our bandit verification protocol. A VC allows one to commit to a vector, and reveal individual components whose consistency with the commitment can be proven. The commitment and opening proofs require space independent of the length of the vector. A SNARK allows a prover to succinctly prove that a given instance belongs to a polynomial-time computable relation.

We apply these tools in our protocol by having the prover send a commitment to the vector \tilde{u} of purported rewards, rather than sending \tilde{u} in full. It uses a SNARK to prove that the optimal smooth policy with respect to \tilde{u} has a claimed value. The verifier then proceeds exactly as in Protocol 1; but instead of examining \tilde{u} directly at each index, it asks the prover for the value and opening proof.

Lemma B.7. *Let $\lambda \in \mathbb{N}$ be a security parameter, let $n \in \mathbb{N}$, let $\varepsilon \in (0, 1)$, let q be an n -armed bandit, and let u denote be the vector of expected utilities of q . There exists a protocol (Protocol 4) as follows. The protocol consists of a trusted setup phase, in which shared parameters are generated by a trusted entity; and an interactive phase between a prover and a verifier. Assuming the security of the underlying SNARK Π and vector commitment VC, our protocol satisfies:*

- **Completeness:** *If the prover behaves honestly, the verifier outputs a value t that is within ε of the value of the optimal σ -smooth policy with probability at least $\frac{2}{3} - \text{negl}(\lambda)$.*
- **Soundness:** *Even if the p.p.t. prover behaves arbitrarily, the probability that the verifier outputs a value t that is not within ε of the optimal value is at most $\frac{1}{3} + \text{negl}(\lambda)$.*

The efficiency of the protocol is as follows:

- *If Π has $O(\lambda)$ -sized proofs, and VC has $O(\lambda)$ -sized commitments and opening proofs, the protocol consists of $O(\lambda \cdot (\sigma n \log n \log^2(1/\varepsilon)/\varepsilon))$ bits sent between P and V .*
- *P performs $m_P = O(n \log(n/\varepsilon)/\varepsilon^2)$ nonadaptive queries to the bandit oracle and runs in time $\text{poly}(n, 1/\varepsilon)$.*
- *V performs*

$$m_V = O\left(\frac{n\sigma}{\varepsilon^2} \cdot \log\left(\frac{n\sigma}{\varepsilon}\right) \log\left(\frac{1}{\varepsilon}\right)\right)$$

nonadaptive queries to the bandit oracle, and runs in time $\text{poly}(n, 1/\varepsilon)$.

We remark that there exist SNARKs with $O(\lambda)$ -sized proofs for arithmetic circuit satisfiability, which have knowledge soundness in the generic group model [Groth, 2016, Theorem 2]. There also exist vector commitments with $O(\lambda)$ -sized commitments and opening proofs; for example, the CDH-based scheme of [Catalano and Fiore, 2013, Theorem 5].

When $n\sigma$ is a constant, the number of bits sent between P and V is only $\tilde{O}(\lambda)$, hiding $\log(1/\varepsilon)$ factors.

The protocol and proof for Lemma B.7 appear in Section D.3.

B.3 Lower bound for bandit verification

Theorem B.8. *There exist constants $C, c > 0$ as follows. Let $n \in \mathbb{N}$, let $\sigma \in [24/n, 1]$, and let $\varepsilon \geq 0$. Assume that (V, P) is an interactive proof system for verification of ε -optimal σ -smooth policies for n -arm bandits, as in Definition B.4. Then there exists an n -armed bandit q such that if (V, P) are executed with access to the bandit oracle \mathcal{O}_q , then with probability at least $9/10$, V uses at least*

$$m_V \geq C \cdot \frac{\sigma n}{\varepsilon^2} - c = \Omega\left(\frac{\sigma n}{\varepsilon^2}\right)$$

queries to \mathcal{O}_q .

Remark B.9. The linear lower bound in Theorem B.8 is stated only for $\sigma \in [24/n, 1]$, and not for all $\sigma \in [1/n, 1]$. Note that for $\sigma = 1/n$ there exists only a single σ -smooth policy, and so indeed a linear lower bound cannot hold for the entire interval $[1/n, 1]$. However, the constants 24 and $9/10$ appearing in Theorem B.8 are somewhat arbitrary, and can be replaced by constants closer to 1. \square

The proof for Theorem B.8 appears in Section D.5.

B.4 Lower bound for learning smooth bandit strategies

Claim 1 (Lower bound for learning smooth bandits). *Let $n, m \in \mathbb{N}$, $n \geq 3$, let $\sigma \in [5/n, 1]$, and let $\varepsilon \in [0, 1/4]$. Let A be a (possibly randomized) algorithm such that for any n -armed bandit q , A performs m (possibly adaptive) oracle queries to the bandit oracle \mathcal{O}_q , and outputs a σ -smooth*

strategy π such that with probability at least $2/3$, π is an ε -optimal σ -smooth strategy for q . Then $m \geq n/6$.

The proof for Claim 1 appears in Section D.6.

Remark B.10. As in Remark B.9, the linear lower bound cannot hold for the entire interval $[1/n, 1]$. However, the specific constants of $5/n$ and $1/4$ in the statement are somewhat arbitrary. \square

C Games

C.1 Definitions

Definition C.1 (Game). Let $k, n \in \mathbb{N}$. A normal-form game with k players and n actions is a vector $u = (u_1, \dots, u_k)$ such that for each $i \in [k]$, $u_i : [n]^k \rightarrow [0, 1]$ is a utility function for player i . A strategy profile for such a game is a vector $\pi = (\pi_1, \dots, \pi_k)$ where for each $i \in [k]$, $\pi_i = (\pi_{i,1}, \dots, \pi_{i,n}) \in \Delta([n])$ is a strategy for player i .

A game defines a game oracle \mathcal{O}_u (corresponding to “playing the game”) such that given a strategy profile π , the oracle samples an action vector a by independently sampling $a_1 \sim \pi_1, \dots, a_k \sim \pi_k$, and returns a vector of realized utilities $u(a) = (u_1(a), \dots, u_k(a))$. Thus, $\mathcal{O}_u(\pi)$ is the distribution of realized utilities under u obtained by sampling a from π .

Given a game u and strategy profile π , the expected utility for a player $i \in [k]$ is

$$\mathbb{E}_{a_1 \sim \pi_1, \dots, a_k \sim \pi_k} [u_i(a_1, \dots, a_k)].$$

Definition C.2 (Smooth game strategy). Let $k, n \in \mathbb{N}$ and $\sigma \in [1/n, 1]$. In a game with k players and n actions, a strategy $\pi_i = (\pi_{i,1}, \dots, \pi_{i,n})$ for a player $i \in [k]$, is σ -smooth if $\pi_{i,j} \leq \sigma$ for each action $j \in [n]$. A strategy profile is σ -smooth if each strategy in the profile is σ -smooth.³

Definition C.3 (Smooth Nash equilibrium; Definition 4 in Daskalakis et al. [2024]). Let $k, n \in \mathbb{N}$, $\sigma \in [1/n, 1]$, $\varepsilon \geq 0$. Let u be a game with k players and n actions. A strategy profile π for u is a weak ε -approximate σ -smooth Nash equilibrium if for every player $i \in [k]$ and every σ -smooth strategy $\pi'_i \in \Delta([n])$,

$$\mathbb{E}_{a_i \sim \pi'_i, a_{-i} \sim \pi_{-i}} [u_i(a)] - \mathbb{E}_{a \sim \pi} [u_i(a)] \leq \varepsilon.$$

If in addition π is σ -smooth, we say that π is a strong ε -approximate σ -smooth Nash equilibrium.⁴

Definition C.4 (Verification of smooth Nash equilibrium). An interactive proof system for verification of ε -approximate σ -smooth Nash equilibria for k -player n -action games with error η is a pair of algorithms (V, P) such that for all $k, n \in \mathbb{N}$, for every k -player n -action game u , and for all $\sigma \in [1/n, 1]$ and $\varepsilon, \eta \in (0, 1)$, the following two conditions hold:

- **Completeness.** Let π be any ε -approximate σ -smooth equilibrium. Let the random variable

$$X_\pi = [V^{\mathcal{O}_u}(k, n, \varepsilon, \sigma, \eta, \pi), P^{\mathcal{O}_u}(k, n, \varepsilon, \sigma, \eta, \pi)] \in \{\text{accept}, \text{reject}\}$$

denote the output of V after interacting with P , when each of them receives $(k, n, \varepsilon, \sigma, \eta, \pi)$ as input and has oracle access to \mathcal{O}_u . Then

$$\mathbb{P}[X_\pi = \text{accept}] \geq \frac{2}{3}.$$

- **Soundness.** For any π' that is not an $(\varepsilon + \eta)$ -approximate σ -smooth equilibrium, and any (possibly malicious and computationally unbounded) prover P' (which in particular may depend on $k, n, \varepsilon, \sigma, \eta, u$, and π'), the verifier’s output $X_{\pi'} = [V^{\mathcal{O}_u}(k, n, \varepsilon, \sigma, \eta, \pi'), P'] \in \{\text{accept}, \text{reject}\}$ satisfies

$$\mathbb{P}[X_{\pi'} = \text{reject}] \geq \frac{2}{3}.$$

³ This and the following definition correspond to σ' -smoothness in the terminology of Daskalakis et al. [2024], where $\sigma' = 1/(\sigma n)$.

⁴ We neglect specifying the approximation parameter ε when it is 0, and simply speak of (weak or strong) σ -smooth Nash equilibria. In this paper we focus on strong smooth equilibria. Hence, all smooth NE in this paper are strong smooth NE, unless explicitly mentioned otherwise.

C.2 Protocol for verifying smooth Nash equilibria

Assumptions:

- $k, n \in \mathbb{N}; \sigma \in [1/n, 1]; \varepsilon, \eta \in (0, 1)$.
- $u = (u_1, \dots, u_k)$ is a normal-form game with k players and n actions, where for each $i \in [k]$, $u_i : [n]^k \rightarrow [0, 1]$.
- $\pi = (\pi_1, \dots, \pi_k)$ is a strategy profile, where for each $i \in [k]$, $\pi_i \in \Delta([n])$.
- For each $i \in [k]$, $\mathcal{B}(i, u, \pi)$ denotes the n -arm bandit $q = (q_1, \dots, q_n)$ where for each $j \in [n]$, q_j is the distribution of player i 's utility given that player i plays action j , and the remaining players play according to π . Namely, $q_j = \mathcal{O}_u(\pi^{i,j})_i$ where $\pi^{i,j} \in (\Delta([n]))^k$ and for all $i' \in [k]$ and all $j' \in [n]$,

$$\pi_{i'}^{i,j}(j') = \begin{cases} \pi_{i'}(j') & i' \neq i \\ \mathbb{1}(j' = j) & i' = i \end{cases}.$$

INTERACTIVE PHASE:

Both the prover and verifier are given as input $k, n, \sigma, \varepsilon, \eta$

$\delta \leftarrow 1/(3k)$

for $i \in [k]$:

$b_i \leftarrow \text{VERIFYSMOOTHBANDITSTRATEGY}(n, \mathcal{B}(i, u, \pi), \sigma, \pi_i, \varepsilon, \delta, \eta)$

VERIFIER:

if $b_i = 0$ for any $i \in [k]$:

reject and terminate execution

accept

Protocol 2: A verification protocol for strong smooth Nash equilibria of k -player n -action games, satisfying the requirements of Theorem C.5.

Theorem C.5 (Verification for smooth Nash equilibrium). *Let $k, n \in \mathbb{N}$, let $\sigma \in [1/n, 1]$, let $\varepsilon, \eta \in (0, 1)$. Protocol 2 defines an interactive proof system (V, P) for verification of ε -approximate σ -smooth Nash equilibria for k -player n -action games with slackness η such that:*

- *The protocol consists of $\text{poly}(k, n, 1/\eta)$ rounds between P and V , with a total of $\text{poly}(k, n, 1/\eta)$ bits sent.*

- *P performs*

$$m_P = O\left(\frac{n \log(n/\eta)}{\eta^2} \cdot k \log k\right)$$

nonadaptive queries to the bandit oracle and runs in time $\text{poly}(k, n, 1/\eta)$.

- *V performs*

$$m_V = O\left(\frac{n\sigma}{\eta^2} \log\left(\frac{n\sigma}{\eta}\right) \log\left(\frac{1}{\eta}\right) k \log k\right)$$

nonadaptive queries to the game oracle, and runs in time $\text{poly}(k, n, 1/\eta)$.

In particular, in terms of the dependence on n , if $\sigma = \Theta(1/\sqrt{n})$ then $m_V = \tilde{O}(k\sqrt{n})$, and if $\sigma = \Theta(1/n)$ then m_V is independent of n .

The proof of Theorem C.5 appears in Section E.1.

C.3 Lower bound for smooth Nash verification

Theorem C.6 (Lower bound for verification of smooth Nash equilibrium). *Let $k, n \in \mathbb{N}$, $k, n \geq 2$, $\varepsilon \geq 0$, $\eta \in [0, 1]$, and let $\sigma \in [2/n, 1]$. Assume that (V, P) is an interactive proof system for verification of ε -approximate σ -smooth Nash equilibria for k -player n -action games with slackness η , as in Definition C.4. Then the verifier must use at least*

$$m_V = \Omega(kn\sigma)$$

queries to the game oracle.

The proof of Theorem C.6 appears in Section E.2.

D Proofs for bandits

D.1 Proof of upper bound for verification of smooth bandit strategies

In this appendix we prove Theorem B.5.

Claim 2. *Let $q \in (\Delta([0, 1]))^n$ be an n -arm bandit with expected utilities vector $u \in [0, 1]^n$. In the context of Protocol 1, let $\tilde{u} \in [0, 1]^n$ be the purported expected utilities vector provided by the prover. Assume that there exists a σ -smooth policy $\pi = (\pi_1, \dots, \pi_n) \in \Delta([n])$ such that*

$$|\pi \cdot u - \pi \cdot \tilde{u}| \geq \varepsilon/2.$$

Then the verifier in Protocol 1 rejects with probability at least $2/3$.

Proof of Claim 2. For each $i \in [n]$, let $\Delta_i = |u_i - \tilde{u}_i|$. Let $B = \{-1, 0, 1, 2, 3, \dots, \lceil \log_4(1/\varepsilon) \rceil\}$, and let

$$\{I_b : b \in B\}$$

be a partition of the indices $[n]$ into ‘buckets’, such that for $b \geq 0$,

$$I_b = \{i \in [n] : \Delta_i \in (\varepsilon \cdot 4^{b-1}, \varepsilon \cdot 4^b]\},$$

and

$$I_{-1} = \{i \in [n] : \Delta_i \leq \varepsilon/4\}.$$

By the assumption,

$$\begin{aligned} \varepsilon/2 &\leq |\pi \cdot u - \pi \cdot \tilde{u}| \\ &\leq \sum_{i \in [n]} \pi_i \cdot |u_i - \tilde{u}_i| \\ &= \sum_{i \in [n]} \pi_i \cdot \Delta_i \\ &= \sum_{b \in B} \sum_{i \in I_b} \pi_i \cdot \Delta_i \\ &\leq \sum_{b \in B} \sum_{i \in I_b} \pi_i \cdot \varepsilon \cdot 4^b && (i \in I_b \implies \Delta_i \in (\varepsilon \cdot 4^{b-1}, \varepsilon \cdot 4^b]) \\ &\leq \varepsilon/4 + \sum_{b \in B \setminus \{-1\}} \sum_{i \in I_b} \pi_i \cdot \varepsilon \cdot 4^b && (\sum_{i \in [n]} \pi_i \leq 1) \\ &\leq \varepsilon/4 + \sum_{b \in B \setminus \{-1\}} |I_b| \cdot \sigma \cdot \varepsilon \cdot 4^b. && (\pi \text{ is } \sigma\text{-smooth}) \end{aligned}$$

Rearranging and dividing by $(|B| - 1) \cdot \sigma \cdot \varepsilon$ gives

$$\frac{1}{|B| - 1} \sum_{b \in B \setminus \{-1\}} |I_b| \cdot 2^b \geq \frac{1}{4\sigma(|B| - 1)}.$$

Because the maximum is greater than the average, there exists $b^* \in B$, $b^* \geq 0$ such that

$$|I_{b^*}| \geq \frac{1}{2^{b^*} \cdot 4\sigma \cdot (|B| - 1)} \geq \frac{1}{2^{b^*} \cdot 4\sigma \cdot (\log_4(1/\varepsilon) + 2)} \geq \frac{\ln(6)}{a_{b^*}} \cdot n, \quad (1)$$

where a_{b^*} is defined as in Protocol 1.

In the verifier of Protocol 1, consider the iteration of the outer ‘for’ loop in which $b = b^*$. In that iteration, the verifier pulls a_{b^*} arms chosen independently and uniformly at random from $[n]$. The probability that none of these arms belongs to I_{b^*} is

$$\left(1 - \frac{|I_{b^*}|}{n}\right)^{a_{b^*}} \leq \left(1 - \frac{\ln(6)}{a_{b^*}}\right)^{a_{b^*}} \leq \frac{1}{6}, \quad (2)$$

where we used Eq. (1) and the inequality $1 + x \leq e^x$.

Assume that the verifier pulls some arm $i^* \in I_{b^*}$ in iteration b^* of the outer ‘for’ loop in Protocol 1. Then it pulls that arm p_{b^*} times, and obtains an average utility for those pulls of \hat{u}_{i^*} . Let u_{i^*} and \tilde{u}_{i^*} be the true and purported expected rewards for arm i^* .

$$\begin{aligned} |\hat{u}_{i^*} - \tilde{u}_{i^*}| &\geq |u_{i^*} - \tilde{u}_{i^*}| - |u_{i^*} - \hat{u}_{i^*}| \quad (\text{Triangle inequality}) \\ &> \varepsilon_{b^*}/4 - |u_{i^*} - \hat{u}_{i^*}|. \quad (\varepsilon_b := \varepsilon \cdot 4^b; i^* \in I_{b^*} \implies \Delta_{i^*} \in (\varepsilon_{b^*-1}, \varepsilon_{b^*}]) \end{aligned} \quad (3)$$

By Theorem F.1,

$$\begin{aligned} \mathbb{P}\left[|u_{i^*} - \hat{u}_{i^*}| \geq \frac{\varepsilon_{b^*}}{16}\right] &\leq 2 \exp\left(-2 \cdot \left(\frac{\varepsilon_{b^*}}{16}\right)^2 \cdot p_{b^*}\right) \\ &\leq 2 \exp\left(-2 \cdot \left(\frac{\varepsilon_{b^*}}{16}\right)^2 \cdot \frac{128 \cdot \ln(12 \cdot (\log_4(1/\varepsilon) + 2) \cdot a_{b^*})}{\varepsilon_{b^*}^2}\right) \\ &\leq \frac{1}{6 \cdot (\log_4(1/\varepsilon) + 2) \cdot a_{b^*}} < \frac{1}{6}. \end{aligned} \quad (4)$$

Combining Eqs. (3) and (4) implies that

$$\mathbb{P}\left[|\hat{u}_{i^*} - \tilde{u}_{i^*}| > \frac{\varepsilon_{b^*}}{8}\right] > \frac{1}{6}. \quad (5)$$

Finally, applying a union bound to Eqs. (2) and (5) implies that with probability at least $1 - 1/6 - 1/6 = 2/3$, the verifier pulls an arm $i^* \in I_{b^*}$, and obtains a measurement \hat{u}_{i^*} such that $|\hat{u}_{i^*} - \tilde{u}_{i^*}| > \varepsilon_{b^*}/8$. Therefore, the verifier rejects with probability at least $2/3$, as desired. \square

Claim 3. Let $n \in \mathbb{N}$, let $\varepsilon \geq 0$, and let $u, v \in [0, 1]^n$ such that $\max_{i \in [n]} |u_i - v_i| \leq \varepsilon/2$. Let π^* be an optimal σ -smooth strategy for u . Then π^* is an ε -optimal σ -smooth strategy for v .

Proof of Claim 3. For any σ -smooth strategy π' ,

$$\begin{aligned} \pi'v - \pi^*v &\leq (\pi'v - \pi'u) + (\pi'u - \pi^*u) + (\pi^*u - \pi^*v) \\ &\leq (\pi'v - \pi'u) + 0 + (\pi^*u - \pi^*v) \quad (\pi^* \text{ is } \sigma\text{-smooth optimal for } u) \\ &\leq \sum_{i \in [n]} |u_i - v_i| \cdot (\pi'_i + \pi^*_i) \\ &\leq \sum_{i \in [n]} (\varepsilon/2) \cdot (\pi'_i + \pi^*_i) \quad (\max_{i \in [n]} |u_i - v_i| \leq \varepsilon/2) \\ &\leq \varepsilon. \quad (\pi', \pi^* \text{ are distributions}) \quad \square \end{aligned}$$

Claim 4. Let $q \in (\Delta([0, 1]))^n$ be an n -arm bandit. If the prover and verifier of Protocol 1 interact with each other, and each of them has access to the bandit oracle for q , then with probability at least $2/3$, the verifier does not reject, and it outputs a policy π_v that is an ε -optimal σ -smooth policy with respect to q .

Proof of Claim 4. Let $u \in [0, 1]^n$ be the expected utilities vector of q , and let $\tilde{u} \in [0, 1]^n$ be the vector of estimates computed by the honest prover, as in Protocol 1. Then

$$\begin{aligned}
\mathbb{P}\left[\exists i \in [n] : |\tilde{u}_i - u_i| > \frac{\varepsilon}{16}\right] &\leq \sum_{i \in [n]} \mathbb{P}\left[|\tilde{u}_i - u_i| > \frac{\varepsilon}{16}\right] && \text{(Union bound)} \\
&\leq 2n \cdot \exp\left(-2p_P \cdot \left(\frac{\varepsilon}{16}\right)^2\right) && \text{(Theorem F.1)} \\
&\leq 2n \cdot \exp\left(-2 \cdot \frac{128 \cdot \ln(12n/\varepsilon)}{\varepsilon^2} \cdot \left(\frac{\varepsilon}{16}\right)^2\right) < \frac{1}{6}. && \text{(Choice of } p_P)
\end{aligned} \tag{6}$$

Similarly, denoting $B = \{0, 1, 2, \dots, \lceil \log_4(1/\varepsilon) \rceil\}$, the verifier's estimates \hat{u} satisfy

$$\begin{aligned}
\mathbb{P}\left[\exists b \in B \exists t \in [a_b] : |\hat{u}_{i_{b,t}} - u_{i_{b,t}}| > \frac{\varepsilon_b}{16}\right] \\
&\leq \sum_{b \in B} a_b \cdot \mathbb{P}\left[|\hat{u}_{i_{b,t}} - u_{i_{b,t}}| > \frac{\varepsilon_b}{16}\right] && \text{(Union bound)} \\
&\leq \sum_{b \in B} a_b \cdot 2 \exp\left(-2 \cdot p_b \cdot \left(\frac{\varepsilon_b}{16}\right)^2\right) && \text{(Theorem F.1)} \\
&\leq \sum_{b \in B} a_b \cdot 2 \exp\left(-2 \cdot \frac{128 \cdot \ln(12 \cdot (\log(1/\varepsilon) + 2) \cdot a_b)}{\varepsilon_b^2} \cdot \left(\frac{\varepsilon_b}{16}\right)^2\right) \\
&\leq \sum_{b \in B} a_b \cdot \frac{1}{6 \cdot (\log(1/\varepsilon) + 2) \cdot a_b} < \frac{1}{6}.
\end{aligned} \tag{7}$$

Therefore,

$$\begin{aligned}
\mathbb{P}\left[\exists b \in B \exists t \in [a_b] : |\hat{u}_{i_{b,t}} - \tilde{u}_{i_{b,t}}| > \frac{\varepsilon_b}{8}\right] \\
&\leq \mathbb{P}\left[\exists b \in B \exists t \in [a_b] : |\hat{u}_{i_{b,t}} - u_{i_{b,t}}| > \frac{\varepsilon_b}{16} \vee |\tilde{u}_{i_{b,t}} - u_{i_{b,t}}| > \frac{\varepsilon_b}{16}\right] && \text{(Triangle inequality)} \\
&\leq \frac{1}{6} + \frac{1}{6} = \frac{1}{3}. && \text{(Eqs. (6) and (7), union bound)}
\end{aligned}$$

This implies that $\mathbb{P}[G] \geq 2/3$, where G is the event

$$\left\{ \forall i \in [n] : |\tilde{u}_i - u_i| \leq \frac{\varepsilon}{16} \right\} \cap \left\{ \forall b \in B \forall t \in [a_b] : |\hat{u}_{i_{b,t}} - \tilde{u}_{i_{b,t}}| \leq \frac{\varepsilon_b}{8} \right\}.$$

When G occurs, the verifier does not reject, and by Claim 5, it outputs a strategy π_V that is an optimal σ -smooth strategy for a bandit with expected utilities vector \tilde{u} .

By Claim 3 and the assumption that event G occurs, π_V is also an $(\varepsilon/8)$ -optimal σ -smooth strategy for u .

We conclude that with probability at least $2/3$, the verifier does not reject, and it outputs a strategy π_V that is (better than) an ε -optimal σ -smooth strategy for u , as desired. \square

Proof of Theorem B.5. The completeness property follows from Claim 4.

For soundness, let $q \in (\Delta([0, 1]))^n$ be an n -arm bandit with expected utilities vector $u \in [0, 1]^n$. Assume the verifier of Protocol 1 has access to the bandit oracle for q , and it interacts with some (possibly malicious) prover that sends a vector $\tilde{u} \in [0, 1]^n$ of purported expected utilities. Consider two cases.

- **Case I:** There exists a σ -smooth strategy $\pi \in \Delta([n])$ such that $|\pi \cdot u - \pi \cdot \tilde{u}| \geq \varepsilon/2$. Then by Claim 2, the verifier rejects with probability at least $2/3$.
- **Case II:** For every σ -smooth strategy $\pi \in \Delta([n])$, $|\pi \cdot u - \pi \cdot \tilde{u}| < \varepsilon/2$. In this case, either the verifier rejects, or by Claim 5, it outputs a strategy π_V that is an optimal σ -smooth strategy for \tilde{u} . Let π^* be an optimal σ -smooth strategy for u . Then

$$\pi^* u - \pi_V u = \underbrace{\pi^* u - \pi^* \tilde{u}}_{< \varepsilon/2} + \pi^* \tilde{u} - \underbrace{\pi_V u + \pi_V \tilde{u}}_{< \varepsilon/2}$$

$$\begin{aligned}
&< \varepsilon + \pi^* \tilde{u} - \pi_V \tilde{u} && \text{(By assumption of Case II)} \\
&\leq \varepsilon, && \text{(By optimality of } \pi_V \text{ for } \tilde{u})
\end{aligned}$$

So π_V is an ε -optimal σ -smooth policy for u .

We conclude that in both cases, with probability at least $2/3$, either the verifier rejects or it outputs an ε -optimal σ -smooth policy for u . This establishes the soundness property. \square

D.2 Protocol variant for verifying optimality of a given strategy

In this appendix we prove Lemma B.6.

Assumptions:

- $n \in \mathbb{N}$; q is an n -armed bandit.
- $\sigma \in [1/n, 1]$; $\pi \in [0, 1]^n$; $\varepsilon, \eta, \delta \in (0, 1)$.
- $k = \lceil 18 \ln(8/\delta) \rceil$; $\ell = \lceil \frac{32 \ln(8(k+1)/\delta)}{\eta^2} \rceil$.

VERIFYSMOOTHBANDITSTRATEGY($n, q, \sigma, \pi, \varepsilon, \delta, \eta$):

INTERACTIVE PHASE:

for $i \in [k]$:

The prover and verifier run Protocol 1 with parameters $n, \varepsilon = \eta/4, \sigma$.

if the verifier rejects:

Let $\pi^{(i)} := \perp$.

else:

Let $\pi^{(i)} :=$ the strategy output by the verifier.

VERIFIER:

if π is not σ -smooth or not a valid probability distribution:

reject and terminate execution

if $\frac{1}{k} \sum_{i \in [k]} \mathbb{1}(1)[\pi^{(i)} = \perp] \geq \frac{1}{2}$:

reject and terminate execution

for $i \in [k]$:

for $j \in [\ell]$:

if $\pi^{(i)} \neq \perp$, **sample** $a \sim \pi^{(i)}$; **sample** $r_{i,j} \sim q_a$;

else, let $r_{i,j} = 0$;

$v_i \leftarrow \frac{1}{\ell} \sum_{j \in [\ell]} r_{i,j}$

for $j \in [\ell]$:

sample $a \sim \pi$; **sample** $r_j \sim q_a$;

$v \leftarrow \frac{1}{\ell} \sum_{j \in [\ell]} r_j$

if $\text{median}(\{v_i\}_{i \in [k]}) - v > \varepsilon + \eta/2$:

reject

accept

Protocol 3: A protocol for verifying whether the given bandit strategy is η -close to an ε -approximately optimal σ -smooth strategy for the bandit q . It uses Protocol 1 as a subroutine.

Proof of Lemma B.6. We first show the following useful fact:

Let π' be any valid probability distribution, and let u be the vector of utilities in $[0, 1]$ underlying q . If we sample $a \sim \pi'$ and then for each $j \in [\ell]$ we sample $r_j \sim q_a$, then:

$$\mathbb{P}_{\substack{a_j \sim \pi' \\ r_j \sim q_{a_j} \text{ for } j \in [\ell]}} \left[\left| \pi' \cdot u - \frac{1}{\ell} \sum_{j \in [\ell]} r_j \right| > \eta/8 \right] \leq \frac{\delta}{4(k+1)} \quad (8)$$

This follows from a simple application of Theorem F.1, by observing that the expectation of the average of the r_j 's is $\pi' \cdot u$:

$$\begin{aligned} \mathbb{P}_{\substack{a_j \sim \pi' \\ r_j \sim q_{a_j} \text{ for } j \in [\ell]}} \left[\left| \pi' \cdot u - \frac{1}{\ell} \sum_{j \in [\ell]} r_j \right| > \eta/8 \right] &\leq 2 \exp(-2\ell(\eta/8)^2) \\ &\leq 2 \exp\left(-\frac{64}{\eta^2} \cdot \ln(8(k+1)/\delta) \cdot \frac{\eta^2}{64}\right) \\ &= 2 \exp(-\ln(8(k+1)/\delta)) \\ &= \delta/(4(k+1)). \end{aligned}$$

Completeness. We first consider the case where the prover behaves honestly. Completeness of Protocol 1 ensures that for each protocol run, with probability at least $2/3$ the verifier accepts and the policy sent by the prover is $(\eta/4)$ -optimal. For all $i \in [k]$, define random variables $X_i := 1$ if the verifier accepts, and 0 otherwise. Since the protocol runs are independent, the X_i 's are independent. We can therefore apply Theorem F.1:

$$\begin{aligned} \mathbb{P} \left[\left| \frac{2}{3} - \frac{1}{k} \sum_{i=1}^k X_i \right| > \frac{1}{6} \right] &\leq 2 \exp(-2\ell(1/6)^2) \\ &\leq 2 \exp(-\ln(8/\delta)) \\ &= \delta/4. \end{aligned}$$

Therefore, strictly more than half of the $\pi^{(i)}$'s will be $(\eta/4)$ -optimal. Furthermore, by Equation (8) and a union bound, with probability at least $1 - \delta/2$, all estimates v_i and v will be within $\eta/8$ of the true values of $\pi^{(i)}$ and π respectively. This implies that the median of the v_i 's will be within $\eta/8 + \eta/4$ of the optimal value. Therefore, if the value of π is within ε of optimal, its estimate will be within $\varepsilon + \eta/4 + \eta/8 + \eta/8 = \varepsilon + \eta/2$ of the median of the v_i 's and the verifier will accept.

Soundness. We next consider the case where π is at least $(\varepsilon + \eta)$ -far from optimal, and the prover may behave maliciously. If at least half of the invocations of Protocol 1 result in rejection, the verifier rejects. Therefore, for the remainder of the proof we consider the case where more than half of these protocol invocations result in acceptance.

Soundness of Protocol 1 ensures that in each protocol run, the verifier accepts and outputs π that is $\eta/4$ -far from optimal with probability at most $1/3$. For all $i \in [k]$, define random variables $X_i := 1$ if $\pi^{(i)}$ is more than $\eta/4$ -far from optimal and the verifier accepts; $X_i = 0$ otherwise. $\mathbb{P}[X_i = 1] \leq 1/3$; therefore, it follows by the same application of Theorem F.1 used in Case I that

$$\mathbb{P} \left[\frac{1}{k} \sum_{i=1}^k X_i \geq 1/2 \right] \leq \delta/4.$$

Therefore, with probability at least $1 - \delta/4$, strictly more than half of the π_i 's have value within $\eta/4$ of optimal. Also with probability at least $1 - \delta/4$, by a union bound, all values v and v_i are $\eta/4$ -close to their policies' true values. If both of these events occur, which happens with probability at least $1 - \delta/2$, the median v_i is at least $\eta/4$ -close to optimal. Since v is within $\eta/4$ of its true value, which more than $(\varepsilon + \eta)$ -far from optimal by assumption, v is more than $(\varepsilon + \eta/2)$ -far from the median of the v_i 's. Therefore, the verifier will reject. \square

D.3 Low-communication protocol for verifying smooth MAB strategies

We use a SNARK, Π , for the family of relations parameterized by VC.pp and n :

$$\mathcal{R}_{\text{VC.pp},n} := \left\{ (c_v, t; v) : \begin{array}{l} \forall i \in [n], v_i \in [0, 1] \\ \pi \text{ is an optimal } \sigma\text{-smooth policy for } v \\ \pi \cdot v = t \\ c_v, \text{aux}_v = \text{VC.Commit}_{\text{VC.pp}}(v) \end{array} \right\}$$

Proof of Lemma B.7.

- **Communication.** The prover sends a commitment and SNARK proof, each of which consists of $O(\lambda)$ bits. In the interactive phase, the verifier sends $O(a_b \log(1/\varepsilon)) = O(n\sigma \log^2(1/\varepsilon)/\varepsilon)$ indices, each of which can be written in $\log(n)$ bits. The prover sends $O(a_b \log(1/\varepsilon))$ openings and opening proofs, requiring $O(\sigma n \log n \log^2(1/\varepsilon)/\varepsilon)$ bits in total.⁵

To show completeness and soundness, we invoke properties of the SNARK and VC to reduce the analysis to that of Protocol 1.

- **Soundness.** Towards a contradiction, consider a p.p.t. adversary \mathcal{A} that acts as the prover and with probability at least $1/3 + 1/\text{poly}(\lambda)$ causes the verifier to output t that is ε -far from the true value of the optimal σ -smooth policy for some bandit q . Recall that computational knowledge soundness of Π implies that there exists a p.p.t. extractor $\mathcal{X}_{\mathcal{A}}$ that, with overwhelming probability, computes \tilde{u} such that $(c_v, t; \tilde{u}) \in \mathcal{R}_{\text{VC.pp},n}$. That is, c_v is a commitment to \tilde{u} , and the value of the optimal σ -smooth policy of \tilde{u} is indeed t . Now, position binding of the vector commitment implies that with overwhelming probability all openings of \tilde{u} that the prover sends to the verifier are either rejected, or indeed match the corresponding component of \tilde{u} .

Soundness of the protocol now follows exactly from the analysis of Protocol 1.

- **Completeness.** By construction, π is an optimal σ -smooth policy for \tilde{u} , and $(c_{\tilde{u}}, \text{aux}_{\tilde{u}})$ is indeed the output of $\text{VC.Commit}(\tilde{u})$. Therefore, $(c_v, t; \tilde{u}) \in \mathcal{R}_{\text{VC.pp},n}$ and completeness of Π ensures that pf verifies with probability 1. Correctness of VC ensures that all vector commitment openings verify with probability $1 - \text{negl}(\lambda)$. The remaining checks performed by the verifier are exactly those from Protocol 1; therefore, if the prover behaves honestly the verifier accepts with probability at least $2/3 - \text{negl}(\lambda)$. \square

⁵The numbers of queries made by the prover and verifier to the bandit oracle are the same as in Protocol 1.

Assumptions:

- $n \in \mathbb{N}$; $\sigma \in [1/n, 1]$; $\varepsilon \geq 0$; $\lambda \in \mathbb{N}$.
- $q = (q_1, \dots, q_n) \in (\Delta([0, 1]))^n$ is an n -arm bandit.
- $p_P = \lceil 128 \ln(12n/\varepsilon)/\varepsilon^2 \rceil$; $p_V(b) = \lceil 128 \ln(12 \cdot (\log_4(1/\varepsilon) + 2) \cdot a(b))/(4^b \varepsilon)^2 \rceil$.
- $a(b) = \lceil 4^b \varepsilon \cdot 4n\sigma \cdot (\log_4(1/\varepsilon) + 2) \cdot \ln(6)/\varepsilon \rceil$.

TRUSTED SETUP:

VC.pp \leftarrow VC.KeyGen($1^\lambda, n$).
 Let R be the relation in \mathcal{R} parameterized by VC.pp and n .
 II.pp, $\tau \leftarrow$ II.Setup($1^\lambda, R$).

PROVER($n, \varepsilon, \text{VC.pp}, \text{II.pp}$):

for $i \in [n]$
 for $j \in [p_P]$
 sample $r_{i,j} \sim q_i$
 $\tilde{u}_i \leftarrow \frac{1}{p_P} \sum_{j=1}^{p_P} r_{i,j}$
 $\tilde{u} \leftarrow (\tilde{u}_1, \dots, \tilde{u}_n)$
 $\pi \leftarrow \text{COMPUTEOPTIMALSMOOTHBANDITSTRATEGY}(n, \sigma, \tilde{u})$; $t \leftarrow \pi \cdot \tilde{u}$
 $c_{\tilde{u}}, \text{aux}_{\tilde{u}} \leftarrow \text{VC.Commit}_{\text{VC.pp}}(\tilde{u})$; $\text{pf} \leftarrow \text{II.Prove}(R, \text{II.pp}, (c_{\tilde{u}}, t), \tilde{u})$.
send $c_{\tilde{u}}, t, \text{pf}$ to verifier

VERIFIER($n, \varepsilon, \text{VC.pp}, \text{II.pp}$):

receive $c_{\tilde{u}}, t, \text{pf}$ from prover
if reject = II.Verify($R, \text{II.pp}, (c_{\tilde{u}}, t), \text{pf}$):
 reject and terminate execution

INTERACTIVE PHASE:

for $b \in \{0, 1, 2, \dots, \lceil \log_4(1/\varepsilon) \rceil\}$: ▷ Iterate over all bins.
 $\varepsilon_b \leftarrow \varepsilon \cdot 4^b$; $a_b \leftarrow a(b)$; $p_b \leftarrow p_V(b)$
 for $t \in [a_b]$:
 Verifier **samples** $i_{b,t} \sim U([n])$ ▷ Select a bandit arm at random.
 for $j \in [p_b]$:
 Verifier **samples** $r_{b,t,j} \sim q_{i_{b,t}}$ ▷ Pull (query) the bandit arm.
 $\hat{u}_{i_{b,t}} \leftarrow \frac{1}{p_b} \sum_{j=1}^{p_b} r_{b,t,j}$ ▷ Estimate the bandit arm's utility.
 Verifier **sends** $i_{b,t}$
 Prover **sends** $\text{pf}_{\tilde{u}}, \tilde{u}_{i_{b,t}} \leftarrow \text{VC.Open}_{\text{VC.pp}}(c_{\tilde{u}}, i_{b,t}, \text{aux}_{\tilde{u}})$
 if reject = VC.Verify_{VC.pp}($c_{\tilde{u}}, \tilde{u}_{i_{b,t}}, i_{b,t}, \text{pf}_{\tilde{u}}$): ▷ Check opening proof.
 Verifier **rejects** and terminates execution
 if $|\tilde{u}_{i_{b,t}} - \hat{u}_{i_{b,t}}| > \varepsilon_b/8$:
 Verifier **rejects** and terminate execution
 Verifier **outputs** t

Protocol 4: A low communication-complexity verification protocol for bandits, satisfying the requirements of Lemma B.7.

D.4 Computing an optimal smooth policy for a known bandit

Assumptions:

- $n \in \mathbb{N}; \sigma \in [1/n, 1]; u \in [0, 1]^n$.

COMPUTEOPTIMALSMOOTHBANDITSTRATEGY(n, σ, u):

$i_1, \dots, i_n \leftarrow \text{sort } \{1, \dots, n\} \text{ such that } u_{i_1} \geq u_{i_2} \geq \dots u_{i_n}$
for $j \in [n]$:

$$\pi_{i_j} \leftarrow \begin{cases} \sigma & j \leq \lfloor 1/\sigma \rfloor \\ 1 - \sigma \cdot \lfloor 1/\sigma \rfloor & j = \lfloor 1/\sigma \rfloor + 1 \\ 0 & \text{otherwise} \end{cases}$$

$\pi \leftarrow (\pi_1, \dots, \pi_n)$

output π

Algorithm 1: A subroutine of Protocol 1. Computes an optimal σ -smooth strategy for an n -arm bandit with a given expected utilities vector u .

Observe that the strategy returned by the above algorithm is indeed σ -smooth since $1 - \sigma \cdot \lfloor 1/\sigma \rfloor \leq \sigma$.

Claim 5. *Let $n \in \mathbb{N}$, let $\sigma \in [1/n, 1]$, and let q be an n -armed bandit with expected utilities vector $u \in [0, 1]^n$. Then executing COMPUTEOPTIMALSMOOTHBANDITSTRATEGY(n, σ, u) as in Algorithm 1 yields an optimal σ -smooth strategy π for q .*

Proof of Claim 5. Let $\Delta_{\sigma, n} = [0, \sigma]^n$ be the set of all σ -smooth strategies. $\Delta_{\sigma, n}$ is compact, and the expected utility function $f : \Delta_{\sigma, n} \rightarrow \mathbb{R}$ given by

$$f(\pi) = \sum_{i=1}^n \pi_i u_i$$

is continuous. By the extreme value theorem, f attains a maximum in $\Delta_{\sigma, n}$.

Let $\pi^* = (\pi_1^*, \dots, \pi_n^*)$ be the strategy constructed by the algorithm. Namely, for indices $i_1, i_2, \dots, i_n \in [n]$ such that $u_{i_1} \geq u_{i_2} \geq \dots \geq u_{i_n}$, we have that $\pi_{i_j}^* = \sigma$ for $j \in [\lfloor 1/\sigma \rfloor]$, and the remaining weight, if any, is at index i_j for $j = \lfloor 1/\sigma \rfloor + 1$.

We argue that π^* is a maximum of f in $\Delta_{\sigma, n}$. Indeed, assume for contradiction that there exists $\pi' \in \Delta_{\sigma, n}$ such that $f(\pi') > f(\pi^*)$.

Let $s \in [n]$ be the smallest index such that $\pi'_{i_s} u_{i_s} \neq \pi_{i_s}^* u_{i_s}$ (such an index exists because $f(\pi') \neq f(\pi^*)$). Then $\pi'_{i_s} \neq \pi_{i_s}^*$. However, by construction of π^* , it must be that $\pi'_{i_s} < \pi_{i_s}^* \leq \sigma$.

Let $t \in [n]$ be the largest index such that $\pi'_{i_t} \neq \pi_{i_t}^*$. Note that $t > s$ and $\pi'_{i_t} > \pi_{i_t}^*$ (otherwise, that would be a contradiction to $f(\pi') > f(\pi^*)$).

By the choice of i_1, \dots, i_n , it must be that

$$u_{i_s} > u_{i_t}, \tag{9}$$

since otherwise, $u_{i_s} = u_{i_{s+1}} = \dots = u_{i_t}$, and that would be a contradiction to $f(\pi') > f(\pi^*)$ (because π' and π^* differ only on indices i such that $i \in \{i_s, i_{s+1}, \dots, i_t\}$).

Now, let $\delta = \min \{\sigma - \pi'_{i_s}, \pi'_{i_t}\}$. Note that $\delta > 0$ (because $\pi'_{i_s} < \sigma$ and $\pi'_{i_t} > \pi_{i_t}^* \geq 0$). Consider the strategy $\pi^\delta = (\pi_1^\delta, \dots, \pi_n^\delta)$ which is a modification of π' at two entries, such that

- $\pi_{i_s}^\delta = \pi'_{i_s} + \delta \leq \sigma$,
- $\pi_{i_t}^\delta = \pi'_{i_t} - \delta$, and
- $\pi_i^\delta = \pi'_i$ for all $i \notin \{i_s, i_t\}$.

Note that π^δ is σ -smooth, and furthermore,

$$\begin{aligned}
f(\pi^\delta) - f(\pi') &= \sum_{i \in [n]} u_i \cdot (\pi_i^\delta - \pi'_i) \\
&= u_{i_s} \cdot (\pi_{i_s}^\delta - \pi'_{i_s}) + u_{i_t} \cdot (\pi_{i_t}^\delta - \pi'_{i_t}) \\
&= u_{i_s} \cdot \delta + u_{i_t} \cdot (-\delta) \\
&= \delta(u_{i_s} - u_{i_t}) > 0.
\end{aligned}$$

(by Eq. (9) and $\delta > 0$)

This is a contradiction to the maximality of π' . \square

D.5 Proof of lower bound for verification of smooth bandit strategies

The proof of Theorem B.8 uses ideas from the proof of Lemma 3 in Even-Dar et al. [2002]. Specifically, it uses a reduction to the coin bias problem, which is defined as follows.

Definition D.1 (Coin Bias Problem). *Let $\varepsilon, \delta \in (0, 1)$ and let $m \in \mathbb{N}$. An algorithm A solves the coin bias problem with precision ε , confidence $1 - \delta$, and sample complexity m if for every $b \in \{-1, 1\}$,*

$$\mathbb{P}_{X \sim (\text{Ber}(\frac{1}{2} + b\varepsilon))^m} [A(X) = b] \geq 1 - \delta.$$

The probability is over the randomness of A and of $X = (X_1, \dots, X_m)$, which is an i.i.d. sample of size m from the Bernoulli distribution with parameter $\frac{1}{2} + b\varepsilon$.

The following well-known claim gives a sample complexity lower bound for the coin bias problem (see, e.g., Lemma 5.1 of Anthony and Bartlett, 2002; cf. Theorem 11.8.3 in ?).

Claim 6 (Lower Bound for the Coin Bias Problem). *Let $\varepsilon, \delta \in (0, 1/4)$ and $m \in \mathbb{N}$. Consider the following experiment:*

1. Sample $b \sim \mathcal{U}(\{-1, 1\})$.
2. Sample X_1, X_2, \dots, X_m i.i.d. from $\text{Ber}(\frac{1}{2} + b\varepsilon)$.

Let $f : \{0, 1\}^m \rightarrow \{-1, 1\}$. If

$$\mathbb{P}[f(X_1, X_2, \dots, X_m) = b] \geq 1 - \delta$$

then

$$m = \Omega\left(\frac{\log(1/\delta)}{\varepsilon^2}\right).$$

The reduction from the coin bias problem to bandit verification is depicted in Figure 1.

| | 1 | 2 | 3 | 4 | ... | $n-1$ | n |
|--------------------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----|-----------------------------|-----------------------------|
| Prover P_1: | $\frac{1}{2} - \varepsilon$ | $\frac{1}{2} - \varepsilon$ | $\frac{1}{2} - \varepsilon$ | $\frac{1}{2} - \varepsilon$ | ... | $\frac{1}{2} - \varepsilon$ | $\frac{1}{2} - \varepsilon$ |
| Verifier V_1: | $\frac{1}{2} - \varepsilon$ | x_t | x_t | $\frac{1}{2} - \varepsilon$ | ... | x_t | $\frac{1}{2} - \varepsilon$ |
| Prover P_{-1}: | $\frac{1}{2} - \varepsilon$ | $\frac{1}{2} - \varepsilon$ | $\frac{1}{2} - \varepsilon$ | $\frac{1}{2} - \varepsilon$ | ... | $\frac{1}{2} - \varepsilon$ | $\frac{1}{2} - \varepsilon$ |
| Verifier V_{-1}: | $1 - x_t$ | $\frac{1}{2} - \varepsilon$ | $1 - x_t$ | $\frac{1}{2} - \varepsilon$ | ... | $\frac{1}{2} - \varepsilon$ | $1 - x_t$ |

Figure 1: Illustration of the reduction from the coin bias problem to bandit verification. Algorithm 2 is given access to a coin with distribution $\text{Ber}(\frac{1}{2} + b^* \cdot \varepsilon)$ for some unknown $b^* \in \{-1, 1\}$. Algorithm 2 defines two instances of bandit verification, (V_1, P_1) and (V_{-1}, P_{-1}) . The prover in both cases has access to a bandit oracle where all arms have utility $\text{Ber}(\frac{1}{2} - \varepsilon)$. The bandit oracles for the verifiers are similar, except that a subset of $1/\sigma$ arms selected at random (depicted here in green) has different utilities. For V_1 , the reward for these arms is generated by simply flipping the coin each time the arm is queried to get a fresh sample $x_t \sim \text{Ber}(\frac{1}{2} + b^* \cdot \varepsilon)$. For V_{-1} the process is similar, except that the reward is $1 - x_t$, i.e., the outcome from the coin is reversed. This design ensures that verifier V_{-b^*} has an oracle where *all* arms have reward $\text{Ber}(\frac{1}{2} - \varepsilon)$, whereas V_{b^*} has a subset of $1/\sigma$ arms with reward $\text{Ber}(\frac{1}{2} + \varepsilon)$.

Assumptions:

- $n, m \in \mathbb{N}, \sigma \in [24/n, 1], \varepsilon \geq 0$. For simplicity, assume $1/\sigma \in \mathbb{N}$.
- (V, P) is an interactive proof system for verification of ε -optimal σ -smooth policies for n -arm bandits, as in Definition B.4.
- $x_1, x_2, \dots, x_m \in \{0, 1\}$ are sampled as in the coin experiment of Claim 6.

DECIDECOINBIAS(x_1, x_2, \dots, x_m):

```

for  $b \in \{-1, 1\}$ :
    sample  $I_b \sim \mathcal{U}\left(\binom{[n]}{1/\sigma}\right)$   $\triangleright I_b \subseteq [n]$  is a subset of cardinality  $|I_b| = 1/\sigma$ .
     $(V_b, P_b) \leftarrow$  fresh copy of  $(V, P)$ 
     $q \leftarrow (\text{Ber}(1/2 - \varepsilon), \dots, \text{Ber}(1/2 - \varepsilon))$   $\triangleright q$  is an  $n$ -arm bandit.
     $b \leftarrow 1$   $\triangleright$  Simulate interactive proof  $(P_b, V_b)$ .
     $t \leftarrow 1$ 
    while  $t \leq m$ :  $\triangleright$  Use coin sample  $x_t$  for  $t = 1, 2, \dots, m$ .
        continue simulation of  $(V_b, P_b^{\mathcal{O}_q})$  until  $V_b$  terminates or queries an oracle arm
        if  $V_b$  queried an arm  $i \in [n]$ :
            if  $i \in I_b$ :
                 $r \leftarrow \begin{cases} x_t & b = 1 \\ 1 - x_t & b = -1 \end{cases}$   $\triangleright$  Simulate arm  $i \in I_b$  using coin sample  $x_t$ .
                send  $r$  to  $V_b$ 
                 $b \leftarrow (-b)$   $\triangleright$  Switch to simulating  $(P_{-b}, V_{-b})$ .
                 $t \leftarrow t + 1$   $\triangleright$  Proceed to next coin sample.
            else:
                sample  $r \sim \text{Ber}(1/2 - \varepsilon)$   $\triangleright$  Simulate arm  $i \notin I_b$ .
                send  $r$  to  $V_b$ 
            else:  $\triangleright V_b$  has terminated.
                 $\pi \leftarrow$  output returned by  $V_b$ 
                if  $\pi = \text{reject}$  or  $\sum_{i \in I_b} \pi_i \geq 1/2$ :
                    output  $b$  and terminate
                else:
                    output  $-b$  and terminate
    output  $\perp$  and terminate  $\triangleright$  We have exhausted our sample supply  $x_1, \dots, x_m$ .
```

Algorithm 2: A reduction from the coin bias problem of Claim 6 to verification for smooth bandits (as in Definition B.4).

Proof of Theorem B.8. For simplicity, assume that $1/\sigma \in \mathbb{N}$. We show a reduction from the coin bias problem of Claim 6 to verification for smooth bandits (Definition B.4).

The reduction is given in Algorithm 2, which solves the coin bias problem by simulating two copies of an interactive proof (V, P) for bandit verification, denoted (V_1, P_1) and (V_{-1}, P_{-1}) .

In both copies, the prover is given oracle access to an n -armed bandit where all arms have utility $\text{Ber}(1/2 - \varepsilon)$. For each $b \in \{-1, 1\}$, the verifier V_b is given access to a bandit oracle corresponding to a subset $I_b \subseteq [n]$ of cardinality $1/\sigma$ that is chosen uniformly at random. For each arm $i \notin I_b$, the

utility is distributed $\text{Ber}(1/2 - \varepsilon)$, the same as in the prover's oracle. However, for arms $i \in I_b$, the utility is simulated using a sequence of i.i.d. samples x_1, x_2, \dots, x_m from the coin bias problem. Specifically, for verifier V_1 , the utility for an arm $i \in I_1$ is simulated by returning x_t , where x_t is the next unused sample from the sequence x . In contrast, for verifier V_{-1} , the utility for an arm $i \in I_{-1}$ is simulated by returning $1 - x_t$ (flipping the value of the next available sample).

Algorithm 2 alternates between simulating the interaction of (V_1, P_1) and the interaction of (V_{-1}, P_{-1}) . Each interaction is simulated until a sample x_t is used, at which point Algorithm 2 switches to simulating the other interaction. This ensures that the numbers of samples x_t used for the two simulations differ by at most 1.

Assume that the bias of the coin is $1/2 + b^* \cdot \varepsilon$ for some fixed but unknown $b^* \in \{-1, 1\}$. Observe that for all $b \in \{-1, 1\}$, all arms in the bandit oracle for V_b that are not in I_b have utility distribution $\text{Ber}(1/2 - \varepsilon)$; however, in simulation (V_{b^*}, P_{b^*}) , the arms in I_{b^*} have utility distribution $\text{Ber}(1/2 + (b^*)(b^*) \cdot \varepsilon) = \text{Ber}(1/2 + \varepsilon)$, while for (V_{-b^*}, P_{-b^*}) the arms in I_{-b^*} have utility distribution $\text{Ber}(1/2 + (-b^*)(b^*) \cdot \varepsilon) = \text{Ber}(1/2 - \varepsilon)$. In particular, for exactly one of the simulations, the expected utility of *all* n arms the verifier's oracle is $1/2 - \varepsilon$, and in the other simulation there is a randomly-chosen subset of $1/\sigma$ arms that have expected utility $1/2 + \varepsilon$, and the remaining $n - 1/\sigma$ arms have expected utility $1/2 - \varepsilon$.

We now show that Algorithm 2 solves the coin bias problem correctly. In Algorithm 2, either the simulation of (V_{b^*}, P_{b^*}) or of (V_{-b^*}, P_{-b^*}) terminates first and determines the output of Algorithm 2. If (V_{b^*}, P_{b^*}) terminates first, then the soundness of (V_{b^*}, P_{b^*}) implies that with probability at least $2/3$, V_{b^*} outputs π such that $\pi = \text{reject}$ or π is a σ -smooth ε -optimal policy for the bandit that V_{b^*} had oracle access to, which is an oracle where each arm $i \in [n]$ has expected utility

$$u_i = 1/2 - (-1)^{\mathbb{1}(i \in I_{b^*})} \cdot \varepsilon.$$

The optimal σ -smooth policy for that bandit is given by $\pi_i^* = \mathbb{1}(i \in I_{b^*}) \cdot \sigma$, which has expected utility $\pi^* \cdot u = 1/2 + \varepsilon$. Thus, with probability at least $2/3$,

$$\pi \cdot u \geq \pi^* \cdot u - \varepsilon = \frac{1}{2}. \quad (10)$$

But

$$\pi \cdot u = \left(\frac{1}{2} + \varepsilon\right) \cdot \sum_{i \in I_{b^*}} \pi_i + \left(\frac{1}{2} - \varepsilon\right) \cdot \sum_{i \notin I_{b^*}} \pi_i. \quad (11)$$

Combining Eqs. (10) and (11) gives that with probability at least $2/3$,

$$\sum_{i \in I_{b^*}} \pi_i \geq \frac{1}{2}. \quad (12)$$

Therefore, if (V_{b^*}, P_{b^*}) terminates first, then with probability at least $2/3$, V_{b^*} reaches the first output statement in Algorithm 2 and outputs b^* , representing a coin bias of $\text{Ber}(1/2 + b^* \cdot \varepsilon)$, which is the correct answer.

On the other hand, in the copy (V_{-b^*}, P_{-b^*}) , all the arms in the verifier's bandit oracle and in the prover's bandit oracle have expected utility $1/2 - \varepsilon$. By completeness of (V_{-b^*}, P_{-b^*}) , with probability at least $2/3$, the verifier V_{-b^*} outputs a policy $\pi \neq \text{reject}$ that is a distribution over $[n]$. Because in that simulation all the arms in $[n]$ are indistinguishable, and the set I_{-b^*} is chosen uniformly at random, the expected weight that π assigns to arms in I_{-b^*} is

$$\mathbb{E} \left[\sum_{i \in I_{-b^*}} \pi_i \right] = \frac{|I_{-b^*}|}{n} = \frac{(\frac{1}{\sigma})}{n} = \frac{1}{\sigma n}.$$

By Markov's inequality,

$$\mathbb{P} \left[\sum_{i \in I_{-b^*}} \pi_i \geq \frac{1}{2} \right] \leq \frac{2}{n\sigma} \leq \frac{1}{12}, \quad (13)$$

where we have used the assumption that $\sigma \geq 24/n$. Therefore, if (V_{-b^*}, P_{-b^*}) terminates first, then with probability at least

$$\frac{2}{3} - \frac{1}{12} = \frac{7}{12},$$

V_{-b^*} reaches the second output statement in Algorithm 2 and outputs $-(-b^*) = b^*$, which again is the correct answer. This establishes that Algorithm 2 solves the coin bias problem correctly with probability at least $7/12$.

We now show that the correctness of Algorithm 2 implies a lower bound on the number of oracle queries used by the verifier in bandit verification. Let m_{b^*} and m_{-b^*} be the number of coin samples x_t used by V_{b^*} and V_{-b^*} , respectively. Let k be the total number of oracle queries performed by V_{-b^*} (some of which used coin samples, so $m_{-b^*} \leq k$). We want to show a lower bound on k .

Seeing as Algorithm 2 solves the coin bias problem correctly with probability at least $7/12$, Claim 6 implies that there exists a constant $c_0 > 0$ such that the total number $m_{b^*} + m_{-b^*}$ of coin samples used by Algorithm 2 is lower bounded by

$$m_{b^*} + m_{-b^*} \geq c_0 \cdot \frac{1}{\varepsilon^2}. \quad (14)$$

In the simulation (V_{-b^*}, P_{-b^*}) , all the arms in the bandit oracles for V_{-b^*} and P_{-b^*} have expected utility $1/2 - \varepsilon$. For these oracles, the arms in I_{-b^*} are indistinguishable from the other arms in $[n]$. Hence, if V_{-b^*} makes k queries to the bandit oracle, then the expected number queries that V_{-b^*} makes to arms in I_{-b^*} , and therefore the number of a coin samples V_{-b^*} uses, is

$$\mathbb{E}[m_{-b^*}] = k \cdot \frac{|I_{-b^*}|}{n} = k \cdot \frac{(\frac{1}{\sigma})}{n} = \frac{k}{\sigma n}.$$

By Markov's inequality,

$$\mathbb{P}\left[m_{-b^*} \geq 10 \cdot \frac{k}{\sigma n}\right] \leq \frac{1}{10}.$$

In other words, with probability at least $9/10$,

$$m_{-b^*} < 10 \cdot \frac{k}{\sigma n}. \quad (15)$$

Because Algorithm 2 alternates between simulating (V_{b^*}, P_{b^*}) and (V_{-b^*}, P_{-b^*}) , the numbers m_{b^*} and m_{-b^*} of coin samples used by each simulation differ by at most 1. Hence,

$$m_{b^*} + m_{-b^*} \leq 2m_{-b^*} + 1. \quad (16)$$

We conclude that

$$\begin{aligned} k &> \frac{n\sigma}{10} \cdot m_{-b^*} && \text{(By Eq. (15))} \\ &\geq \frac{n\sigma}{10} \cdot \frac{m_{b^*} + m_{-b^*} - 1}{2} && \text{(By Eq. (16))} \\ &\geq \frac{n\sigma}{10} \cdot \frac{\frac{c_0}{\varepsilon^2} - 1}{2} && \text{(By Eq. (14))} \\ &= \frac{c_0}{20} \cdot \frac{n\sigma}{\varepsilon^2} - \frac{1}{20}, \end{aligned}$$

as desired. \square

D.6 Proof of lower bound for learning smooth bandit strategies

Proof of Claim 1. For simplicity, we assume that $1/\sigma$ is an integer (the proof for the general case is similar). We will assume in the proof that $m \leq n/2$ (otherwise, there is nothing to prove).

For each $b \in \{0, 1\}$, let \mathcal{D}_b denote the degenerate distribution such that $\mathbb{P}_{x \sim \mathcal{D}_b}[x = b] = 1$. Let $\mathcal{S}_\sigma = \binom{[n]}{1/\sigma}$ be the collection of all subsets of $[n]$ of size $1/\sigma$. For every set $S \in \mathcal{S}_\sigma$, let $q^S \in (\Delta([0, 1]))^n$ be an n -armed bandit such that for each $i \in [n]$, $q_i^S = \mathcal{D}_{\mathbf{1}(i \in S)}$. In words, q^S is a bandit where arms in S always give utility 1, and the remaining arms always give utility 0. Let $u^S = \text{utility}(q^S)$ denote the expected utilities vector of the bandit q^S . For $S \in \mathcal{S}_\sigma$ and $i \in [n]$, let $S_i = \mathbf{1}(i \in S)$.

For any $S \in \mathcal{S}_\sigma$, let $\pi^S = (\pi_1^S, \dots, \pi_n^S)$ be the uniform distribution on S , i.e., $\pi_i^S = \mathbf{1}(i \in S)\sigma$. Note that π^S is a σ -smooth strategy, and it has expected utility

$$\pi^S \cdot u^S = \sum_{i \in S} \sigma \cdot 1 = 1. \quad (17)$$

Consider the following experiment:

1. Sample a subset S uniformly at random from \mathcal{S}_σ .
2. Execute A with respect to the bandit q^S .

Let $I = \{I_1, \dots, I_m\}$ be the indices of the arms pulled by A , and let $G = S \cap I$ be the “good” arms pulled by A (i.e., the arms queried by A that have utility 1).

In the experiment, for each $i \in [n]$, $\mathbb{E}[S_i] = |S|/n = 1/(\sigma n)$. We may assume without loss of generality that A issues precisely m queries, each to a different arm.⁶ It follows that

$$\mathbb{E}[|G|] = \mathbb{E}\left[\sum_{t=1}^m S_{I_t}\right] = \sum_{t=1}^m \mathbb{E}[S_{I_t}] = m \cdot \mathbb{E}[S_1] = \frac{m}{\sigma n}. \quad (18)$$

The strategy π that A outputs in the experiment has expected utility

$$\mathbb{E}[\pi \cdot u^S] = \mathbb{E}\left[\sum_{i=1}^n \pi_i \cdot u_i^S\right] = \mathbb{E}\left[\sum_{i \in G} \pi_i \cdot u_i^S\right] + \mathbb{E}\left[\sum_{i \in [n] \setminus G} \pi_i \cdot u_i^S\right]. \quad (19)$$

Consider each term separately. For the first term,

$$\mathbb{E}\left[\sum_{i \in G} \pi_i \cdot u_i^S\right] \leq \mathbb{E}\left[\sum_{i \in G} \pi_i \cdot 1\right] = \mathbb{E}[\pi_G], \quad (20)$$

where $\pi_G = \sum_{i \in G} \pi_i$. For the second term,

$$\begin{aligned} \mathbb{E}\left[\sum_{i \in [n] \setminus G} \pi_i \cdot u_i^S\right] &= \mathbb{E}\left[\sum_{i \in [n] \setminus G} \pi_i \cdot S_i\right] \\ &\leq \frac{1}{\sigma(n-m)} \cdot \mathbb{E}\left[\sum_{i \in [n] \setminus G} \pi_i\right] \\ &\leq \frac{2}{\sigma n} \cdot \mathbb{E}[1 - \pi_G]. \end{aligned} \quad (m \leq n/2) \quad (21)$$

Combining Eqs. (19) to (21) yields

$$\mathbb{E}[\pi \cdot u^S] \leq \mathbb{E}\left[\pi_G + \frac{2}{\sigma n} \cdot (1 - \pi_G)\right] = \frac{2}{\sigma n} + \left(1 - \frac{2}{\sigma n}\right) \mathbb{E}[\pi_G].$$

From Eq. (18) and the σ -smoothness of π ,

$$\mathbb{E}[\pi_G] = \mathbb{E}\left[\sum_{i \in G} \pi_i\right] \leq \mathbb{E}[\sigma \cdot |G|] = \frac{m}{n}.$$

Namely,

$$\mathbb{E}[\pi \cdot u^S] \leq \frac{2}{\sigma n} + \left(1 - \frac{2}{\sigma n}\right) \cdot \frac{m}{n}. \quad (22)$$

From the utility of the optimal σ -smooth strategy (Eq. (17)) and the assumption that with probability at least $2/3$, A outputs an ε -optimal σ -smooth strategy,

$$\mathbb{P}[\pi \cdot u^S \geq 1 - \varepsilon] \geq 2/3,$$

so

$$\mathbb{E}[\pi \cdot u^S] \geq \mathbb{P}[\pi \cdot u^S \geq 1 - \varepsilon] \cdot (1 - \varepsilon) \geq \frac{2}{3} \cdot (1 - \varepsilon) \geq \frac{1}{2}. \quad (23)$$

⁶In the experiment, the distributions of each arm is degenerate. So an algorithm that issues less than m queries, or queries the same arm more than once, can be transformed into an algorithm with the same output that pulls precisely m distinct arms.

Combining Eqs. (22) and (23) yields,

$$\frac{2}{\sigma n} + \left(1 - \frac{2}{\sigma n}\right) \cdot \frac{m}{n} \geq \frac{1}{2}.$$

We conclude that

$$m \geq n \cdot \frac{\sigma n}{\sigma n - 2} \cdot \left(\frac{1}{2} - \frac{2}{\sigma n}\right) \geq \frac{n}{6}. \quad (\sigma \geq 5/n)$$

□

E Proofs for games

E.1 Proof for game verification

In this appendix we prove Theorem C.5.

Proof of Theorem C.5. Protocol 2 reduces the problem of verifying approximate optimality of a smooth Nash equilibrium to several bandit verification tasks, one for each player.

That is, verifying that each player $i \in [k]$ has no deviation to a σ -smooth strategy increasing its expected utility by at least ε is equivalent to verifying a bandit problem defined as follows. Observe that for each action $j \in [n]$, π specifies a distribution over player i 's utility, given by the output of the game oracle $\mathcal{O}_u(\pi)_i$. This induces an n -arm bandit, denoted $\mathcal{B}(i, u, \pi)$. A strategy π_i for this bandit is σ -smooth if and only if π_i is a σ -smooth strategy for player i in the given game. Therefore, player i has no smooth deviation increasing their utility by at least ε under π if and only if π_i is an ε -approximate σ -smooth policy for $\mathcal{B}(i, u, \pi)$.

Protocol 2 leverages this equivalence between each player's optimality in the game and bandit optimality, simply by having the prover and verifier engage in a verification protocol for each of these bandits. The prover sends k length- n vectors of empirical average rewards \hat{t}_i , one for each bandit. The verifier then checks optimality of π for player i under bandit $\mathcal{B}(i, u, \pi)$ with maximum error probability $\delta = 1/(3k)$. If any of these subprotocols fails, the verifier rejects and terminates.

Completeness. Observe that if π is indeed an ε -approximately optimal σ -smooth strategy for the game, every induced bandit policy is approximately optimal as well. By Lemma B.6, for each i Algorithm 1 succeeds with probability at least $1/3k$. By a union bound, all k subprotocols succeed with probability at least $2/3$.

Soundness. If π is not an $(\varepsilon + \eta)$ -approximately optimal smooth strategy, there must be a player i for which the induced bandit policy is not approximately optimal. For this player, the bandit verification subprotocol rejects with probability at least $1 - 1/3k \geq 2/3$.

Query complexity. The prover and verifier engage in k runs of Protocol 3 with $\delta = 1/3k$. The query complexity follows. □

E.2 Proof of lower bound for smooth Nash verification

Proof of Theorem C.6. For simplicity, assume that $1/\sigma$ is an integer. Let $\mathcal{S} = \binom{[n]}{1/\sigma}$ be the collection of all subsets of $[n]$ of size $1/\sigma$. For every vector $s \in \mathcal{S}^k$ and integer $i^* \in [k]$, let $u^{s, i^*} = (u_1^{s, i^*}, \dots, u_k^{s, i^*})$ be a k -player n -action game as follows. For each $i \in [k]$, $u_i^{s, i^*} : [n]^k \rightarrow [0, 1]$ is a utility functions such that for every action vector $a \in [n]^k$,

$$u_i^{s, i^*}(a) = \begin{cases} 1 & i = i^* \wedge (\forall j \in [k] : a_j \in s_j) \\ 0 & \text{otherwise.} \end{cases}$$

Note that there exists a σ -smooth profile strategy π^{s, i^*} for u^{s, i^*} where player i^* has expected utility 1, and all other players have expected utility 0. Namely, π^{s, i^*} is the profile where each player $i \in [k]$ selects an action uniformly at random from the set $s_i \in \mathcal{S}$.

For every vector $s \in \mathcal{S}^k$ and integer $i^* \in [k]$, define a distribution \mathcal{D}^{s, i^*} over strategy profiles as follows. When $\pi \sim \mathcal{D}^{s, i^*}$, then with probability 1, for every $i \in [k] \setminus \{i^*\}$, $\pi_i = U(s_i)$ (as in the strategy π^{s, i^*} described in the previous paragraph).

For player i^* , the strategy π_{i^*} is sampled as follows.

1. Sample $R \sim U(\mathcal{S}) \mid (R \cap s_{i^*} = \emptyset)$.
2. Set $\pi_{i^*} = U(R)$.

Let $u^0 = (u_1^0, \dots, u_k^0)$ denote the game where all players always receive utility 0 ($u_i^0 \equiv 0$ is a constant function for all $i \in [k]$).

Now, consider the following experiment.

1. Sample $i^* \sim U([k])$
2. Sample $s \sim U(\mathcal{S}^k)$
3. Sample $\pi \sim \mathcal{D}^{s, i^*}$.
4. Sample a bit $b \sim U(\{0, 1\})$
5. Execute the honest prover $P(k, n, \varepsilon, \sigma, \eta, \pi)$ with access to the game oracle \mathcal{O}_{u^0} .
6. If $b = 0$, execute the verifier $V(k, n, \varepsilon, \sigma, \eta, \pi)$ with access to the game oracle \mathcal{O}_{u^0} . Otherwise, if $b = 1$, execute the verifier $V(k, n, \varepsilon, \sigma, \eta, \pi)$ with access to the game oracle $\mathcal{O}_{u^{s, i^*}}$.

Observe that if $b = 0$, then V should accept, because in the game u^0 , every action profile is a Nash equilibrium (and in addition, the profile π sampled from \mathcal{D}^{s, i^*} is also σ -smooth). On the other hand, when $b = 1$ then V should usually reject, because π will have expected utility close to 0 for all players, but player i^* has a deviation that would give it an expected utility of 1.

We now argue that V cannot distinguish between the case $b = 0$ and $b = 1$ unless it makes at least $\Omega(kn\sigma)$ queries to the game oracle.

To see this, make a few observations:

- Without loss of generality, we can assume that every query π' that V sends to the game oracle is a strategy profile where each player $i \in [k]$ plays a pure strategy. (If V wants to send a query that includes a mixed strategy, then V can simulate the result of that query by first sampling an action vector $a \sim \pi'$, and then sending the query consisting of a pure profile corresponding to a to the oracle. The result is identical.)
- Every action vector $a \in [n]^k$ where the number of deviations is not 1, namely,

$$|\{i \in [k] : a_i \notin \text{supp}(\pi_i)\}| \neq 1,$$

satisfies that $u^{s, i^*}(a) = \mathbf{0} = u^0(a)$, where $\mathbf{0} = (0, \dots, 0)$. Hence, we may assume without loss of generality that each query that V sends to the game oracle has precisely one player $i \in [k]$ that deviates to an action not in $\text{supp}(\pi_i)$.

- Assume $b = 1$. Fix an action vector $a \in [n]^k$.⁷ If there exists a unique $j \in [k]$ such that $a_j \notin \text{supp}(\pi_j)$, then the probability that the result of the query π_a corresponding to action profile a is not $\mathbf{0}$ is

$$\begin{aligned} \alpha &:= \mathbb{P}[\mathcal{O}_{u^{s, i^*}}(\pi_a) \neq \mathbf{0}] \\ &= \mathbb{P}[j = i^* \wedge a_j \in s_j] \\ &= \mathbb{P}[j = i^*] \cdot \mathbb{P}[a_j \in s_j \mid j = i^*] \end{aligned} \tag{24}$$

⁷ a is simply a fixed action vector. In particular, it does not depend on any of the random variables in the experiment.

$$= \frac{1}{k} \cdot \frac{\frac{1}{\sigma}}{n - \frac{1}{\sigma}} \leq \frac{2}{kn\sigma}. \quad (\sigma \geq 2/n)$$

From Eq. (24), it follows that the number N of queries that V sends before it has its first success (i.e., the number of queries that receives result $\mathbf{0}$ prior to the first query that receives a nonzero result), is distributed geometrically, $N \sim \text{Geom}(\alpha)$. In particular,

$$\mathbb{P}[N \geq m] = (1 - \alpha)^m.$$

Assume for contradiction that V performs at most m_V queries and has success probability $2/3$. Then for some $c \in [0, 1)$,

$$1 > c \geq \mathbb{P}[N \geq m_V] = (1 - \alpha)^{m_V} \geq \left(e^{-\frac{\alpha}{1-\alpha}}\right)^{m_V}.$$

This implies that

$$m_V \geq \Omega\left(\frac{1-\alpha}{\alpha}\right) = \Omega\left(\frac{1}{\alpha}\right) = \Omega(kn\sigma),$$

as desired. \square

F Concentration of measure

Theorem F.1 (Hoeffding, 1963). *Let $a, b, \mu \in \mathbb{R}$ and $m \in \mathbb{N}$. Let Z_1, \dots, Z_m be a sequence of i.i.d. real-valued random variables and let $Z = \frac{1}{m} \sum_{i=1}^m Z_i$. Assume that $\mathbb{E}[Z] = \mu$, and for every $i \in [m]$, $\mathbb{P}[a \leq Z_i \leq b] = 1$. Then, for every $\varepsilon > 0$,*

$$\mathbb{P}[|Z - \mu| > \varepsilon] \leq 2 \exp\left(\frac{-2m\varepsilon^2}{(b-a)^2}\right).$$

References

- Martin Anthony and Peter L. Bartlett. *Neural Network Learning - Theoretical Foundations*. Cambridge University Press, 2002. ISBN 978-0-521-57353-5. URL http://www.cambridge.org/gb/knowledge/isbn/item1154061/?site_locale=en_GB.
- Dario Catalano and Dario Fiore. Vector commitments and their applications. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography*, Nara, Japan, February 26 - March 1, 2013. *Proceedings*, volume 7778 of *Lecture Notes in Computer Science*, pages 55–72. Springer, 2013. doi:10.1007/978-3-642-36362-7_5. URL https://doi.org/10.1007/978-3-642-36362-7_5.
- Constantinos Daskalakis, Noah Golowich, Nika Haghtalab, and Abhishek Shetty. Smooth Nash equilibria: Algorithms and complexity. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 to February 2, 2024, Berkeley, CA, USA*, volume 287 of *LIPICs*, pages 37:1–37:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICs.ITCS.2024.37. URL <https://doi.org/10.4230/LIPICs.ITCS.2024.37>.
- Eyal Even-Dar, Shie Mannor, and Yishay Mansour. PAC bounds for multi-armed bandit and markov decision processes. In Jyrki Kivinen and Robert H. Sloan, editors, *Computational Learning Theory, 15th Annual Conference on Computational Learning Theory, COLT 2002, Sydney, Australia, July 8-10, 2002, Proceedings*, volume 2375 of *Lecture Notes in Computer Science*, pages 255–270. Springer, 2002. doi:10.1007/3-540-45435-7_18. URL https://doi.org/10.1007/3-540-45435-7_18.
- Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 305–326. Springer, 2016. doi:10.1007/978-3-662-49896-5_11. URL https://doi.org/10.1007/978-3-662-49896-5_11.
- Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, pages 13–30, 1963. doi:doi.org/10.2307/2282952. URL <https://doi.org/10.2307/2282952>.