

Square-Root Regret for Adversarial Multiplayer Bandits without Collision Information or Shared Randomness

Chenyu Gan
Tsinghua University

Abstract

We study adversarial multiplayer bandits with K arms and $2 \leq m < K$ labeled players, without collision information, shared randomness, or an external communication channel. We design a constructive communication and synchronization protocol with a Monte Carlo public constructor. With probability at least $1 - CN^{-32}$ over preprocessing, where $N = 2Km(T + 1)$, its fixed published output satisfies

$$R_T \leq CK^{5/2}\sqrt{T} \log^2(2Km(T + 1))$$

simultaneously for every oblivious reward sequence chosen after preprocessing. Here R_T is expected regret over the players' private execution randomness. Positive reward observations establish a common learning schedule and synchronize players before learning begins. The cost of delayed communication is charged to the support of positive rewards, ensuring that periods with little useful feedback incur only limited regret. A slow-fast learning procedure then maintains valid reward estimates while assignments and scores are exchanged.

Contents

1	Introduction	3
2	Related work	5
3	The algorithm in brief	5
4	Initialization	8
5	The slow-fast learner	13
6	Assignment and score communication	22
7	Scheduling and local implementation	32
8	Proof of the main theorem	46
9	Finite computation and sampling	49
A	Exceptional executions	57
B	Public construction and bootstrap communication	61

1 Introduction

In a multiplayer bandit problem, several players choose arms at the same time and see only their own rewards. If two players choose the same arm, both receive reward zero. A player who observes zero cannot tell whether the arm had zero reward or a collision occurred. Thus the same observations must be used for learning and coordination. We study the adversarial model in which the players have neither collision information nor shared randomness.

We give a constructive strategy with regret $O_{m,K}(\sqrt{T} \log^2(T+1))$ for fixed m, K . The players establish a common learning schedule using only their private randomness and ordinary reward observations, then learn while communicating assignment updates.

1.1 Model and result

Let $2 \leq m < K$ and $T \geq 1$ be integers. Write $[q] = \{1, \dots, q\}$ for a positive integer q . There are m players with distinct labels in $\{0, \dots, m-1\}$ and K arms. The players know K, m, T and have a common clock. Before any private random choices, an oblivious adversary fixes rewards $r_t(a) \in [0, 1]$ for every $t \in [T]$ and $a \in [K]$. Player i chooses $A_{i,t} \in [K]$ and observes only

$$Y_{i,t} = r_t(A_{i,t}) \mathbf{1}\{A_{j,t} \neq A_{i,t} \text{ for every } j \neq i\}. \quad (1)$$

Here $\mathbf{1}\{E\}$ equals one when E holds and zero otherwise. An action may depend on the player's label, the time, its previous actions and observations, and its private random variables. The private randomness of different players is initially independent. There is no additional communication channel and no initially common random seed. The constructive model also permits public preprocessing: a Monte Carlo constructor samples finite tables depending only on K, m, T , then publishes all of them before the adversary chooses the reward matrix. No preprocessing randomness remains secret. The execution randomness of the players is independent of preprocessing and initially independent across players. Regret below is conditional on a fixed published output. Fixing a successful output instead gives deterministic tables in the original model.

For $x \in \mathbb{R}^K$ and $S \subseteq [K]$, put $x(S) = \sum_{a \in S} x_a$, and let $\mathcal{S} = \{S \subseteq [K] : |S| = m\}$. Identify r_t with its vector of arm rewards. Expected regret is

$$R_T = \max_{S \in \mathcal{S}} \sum_{t=1}^T r_t(S) - \mathbb{E} \sum_{t=1}^T \sum_{i=0}^{m-1} Y_{i,t}. \quad (2)$$

The expectation is over the players' private randomness. Under the substitution $\ell_t(a) = 1 - r_t(a)$, this is the loss model without collision information studied by Bubeck, Li, Peres, and Sellke [BLPS20].

For the main bound, set

$$N = 2Km(T+1), \quad \Lambda = \log N.$$

Here N is a convenient size parameter and Λ its logarithm. Constants denoted by c, C are positive and universal, and their values may change between appearances. The notation $x \lesssim y$ means $x \leq Cy$, and $x \asymp y$ means both inequalities. In particular, no dependence on K, m, T is hidden in these constants.

Theorem 1 (Constructive regret bound). *For every $2 \leq m < K$ and $T \geq 1$, there is a Monte Carlo public constructor and a finite-arithmetic private-randomness strategy in model (1). With probability at least $1 - CN^{-32}$ over preprocessing, its fixed published output satisfies, simultaneously for all oblivious reward matrices,*

$$R_T \lesssim K^{5/2} \sqrt{T} \Lambda^2. \quad (3)$$

The adversary may select its matrix after inspecting that output. The public description has $O(N^4 \log N)$ bits and setup takes $\tilde{O}(N^4)$ bit operations. Total online bootstrap decoding uses $\tilde{O}(N^{5/2})$ bit operations; the remaining learner also has polynomial bit complexity.

For $m = K$, player i always choosing arm $i + 1$ gives zero regret. The case $m = 1$ is the ordinary adversarial bandit problem and requires no communication.

Comparison with the previous bound. For oblivious adversarial multiplayer bandits with neither collision information nor shared randomness, the only previously known general sublinear upper bound is that of Bubeck et al. [BLPS20, Theorem 21]: $\tilde{O}(mK^{3/2}T^{1-1/(2m)})$. For fixed m, K , our bound improves the exponent of T from $1 - 1/(2m)$ to $1/2$, matching the standard $\Omega(\sqrt{T})$ adversarial-bandit lower bound up to logarithmic factors. In particular, for $m = 2$ and fixed $K \geq 3$, we obtain $O_K(\sqrt{T} \log^2(T + 1)) = o(T^{2/3})$. This disproves the first conjecture in [BLPS20, Section 2.5] that the two-player regret in this setting is $\Omega(T^{2/3})$. Theorem 1 has a successful public output for each K, m, T ; fixing it before rewards are chosen gives deterministic tables visible to the adversary, while the players use only independent private randomness during execution.

1.2 Positive support and regret

A positive reward proves that the observed arm had no collision:

$$Y_{i,t} > 0 \implies A_{j,t} \neq A_{i,t} \quad (j \neq i). \quad (4)$$

The communication protocol uses this implication to reject incorrect messages. Its duration may be large in calendar time. To measure the relevant delay, define the support at time t and its cumulative value by

$$g_t = \sum_{a=1}^K \mathbf{1}\{r_t(a) > 0\}, \quad G(t) = \sum_{u=1}^t g_u. \quad (5)$$

Since $r_t(S) \leq g_t$, the comparator earns at most the positive support on any interval. We therefore charge communication delay to increments of G . After rewards are rounded, the same notation refers to the rounded rewards.

1.3 Why communication is possible in this model

There are two information problems. First, the receiver cannot identify a collision from a zero reward. The usual binary protocol “collide for zero, avoid for one” is therefore not a reliable bit channel: the second action also produces zero whenever the underlying reward is zero. In particular, an all-zero reward interval carries no observable message. We assume neither a positive lower bound on rewards nor a bound on the length of an all-zero interval.

Positive observations still give one-sided information. If a receiver plays arm a and receives a positive reward, every candidate message that would have caused a collision on a is false. A zero observation rejects no candidate. Repeated positive observations can therefore remove false messages. The arm arrays also ensure that the actions of the other players do not remove all such observations.

Second, the players initially have no common random row for arm choices, codes, recording decisions, or assignment updates. The protocol first communicates the root player’s private row index by means of deterministic public arrays and independent private column choices. Only then does it use the selected learning row.

1.4 Organization.

Section 3 outlines the protocol and analysis. Section 4 establishes a shared public row and bounds the cost of initialization. Section 5 develops the slow-fast learner and proves its regret bound. Sections 6 and 7 specify communication, local updates, and scheduling, and bound the cost of delayed messages and interruptions. Section 8 combines these bounds and transfers the product-stream analysis to the published table. Section 9 gives finite samplers and computational bounds. Appendix A analyzes exceptional executions; Appendix B constructs the public table and proves its uniform and bootstrap communication guarantees.

Acknowledgement of AI use. The author used OpenAI’s GPT-6-astra to assist in developing the public-table construction and protocol design, and OpenAI’s GPT-6-sol to assist in drafting and revising proofs. The author verified the correctness and originality of all content, including references.

2 Related work

Stochastic rewards. Anantharam, Varaiya, and Walrand [AVW87] studied the stochastic bandit problem with multiple plays under centralized control. For decentralized players whose simultaneous choices can collide, Liu and Zhao [LZ10] established logarithmic regret bounds; subsequent algorithms include the musical-chairs approach of Rosenski, Shamir, and Szlak [RSS16]. Bubeck and Budzinski [BB20] gave an $O(\sqrt{T \log T})$ regret strategy for two players and three arms with shared randomness and no communication that avoids collisions with high probability. Bubeck, Budzinski, and Sellke [BBS21] extended the high-probability collision-free guarantee and $\tilde{O}(\sqrt{T})$ regret to arbitrary numbers of players and arms, again using shared randomness. Bonnefoi et al. [BBM⁺18] introduced the setting in which players observe rewards but cannot distinguish a collision from a zero reward, and Besson and Kaufmann [BK18] also studied this feedback model. Lugosi and Mehrabian [LM22] gave the first regret guarantees without collision information or shared randomness: an instance-dependent logarithmic bound and, when a positive lower bound on the mean of the m -th best arm is known, a gap-free square-root-type bound. Boursier and Perchet [BP19] obtained an instance-dependent bound in the no-collision-information setting by communicating through deliberate collisions, assuming a known positive lower bound on every arm mean. Shi, Xiong, Shen, and Yang [SXS20] used error-correcting codes for this communication. Huang, Combes, and Trinh [HCT22] gave an instance-dependent algorithm that does not require a lower bound on the smallest arm mean.

Adversarial rewards. Alatur, Levy, and Krause [ALK20] gave the first decentralized adversarial multiplayer bandit algorithm with sublinear regret, obtaining a $\tilde{O}(T^{2/3})$ bound with collision information. Bubeck, Li, Peres, and Sellke [BLPS20] obtained a $\tilde{O}(\sqrt{T})$ bound for two players with collision information. Without collision information or shared randomness, they proved $\tilde{O}(KT^{3/4})$ regret for two players and $\tilde{O}(mK^{3/2}T^{1-1/(2m)})$ for m players, against an oblivious adversary. They also showed that an adaptive adversary can force linear regret without shared randomness. Shi and Shen [SS21] studied the no-collision-information model through an attackability parameter and proved sublinear regret under conditions on that parameter using coded collision communication.

3 The algorithm in brief

The players must learn which arms reward them while keeping their choices collision-free. Following the slow-fast approach of Bubeck et al. [BLPS20], one player acts as a coordinator and can change arms

frequently; the other $m - 1$ players keep assignments long enough to communicate observations. The rule for choosing the coordinator’s arm is deliberately biased: choosing uniformly can make a promising arm hard to observe once the other arms have been assigned. Section 5 gives the rule and its regret analysis.

Messages can take several rounds. The players therefore record a small sample of observations in a way that has the same learning law during ordinary play, assignment delivery, and score collection. A player can save an observation before learning its assignment and decide later whether it belongs in the sample. If a new assignment is proposed while scores are being collected, the next assignment must account for that proposal; simply restarting from the current learning weights would lose information about why collection began. The recording and communication rules appear in Sections 6 and 7.

The delay costs regret only when positive rewards were available. This lets the proof charge communication to the number of arms with positive rewards rather than to calendar time. The main theorem combines this charge with the learner’s regret bound in Section 8.

The players begin without shared random bits. Before play, a constructor publishes a table of independent control rows. The coordinator privately selects one row and communicates its index during initialization. On a successful initialization all players use that row; failures are paid for in expected regret. The analysis first studies an independent random row and then shows that a suitable fixed published table works for every oblivious reward sequence chosen after publication. The initialization and public-table arguments are in Section 4 and Appendix B. Figure 1 shows the physical workflow.

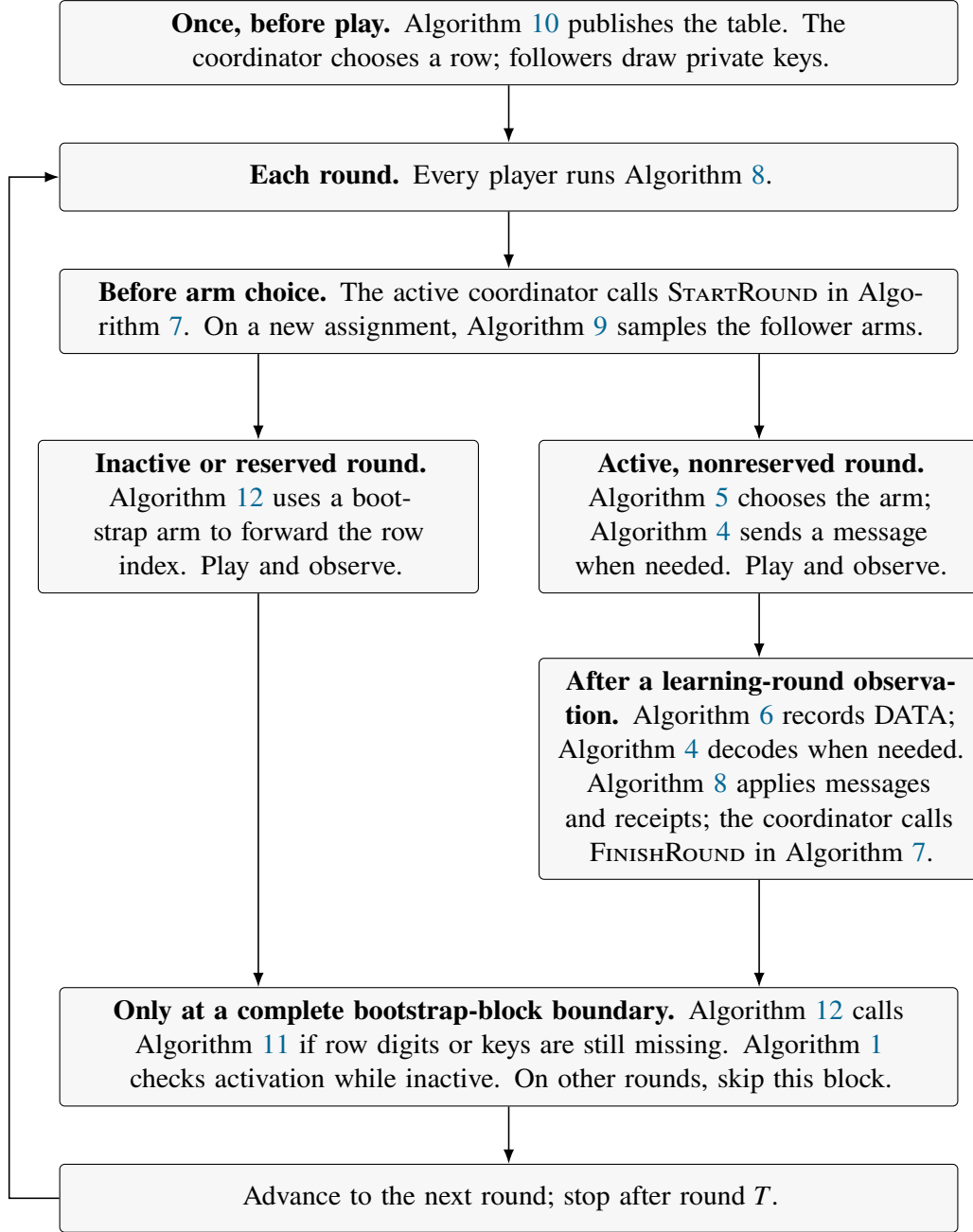


Figure 1: Physical workflow. The reference correction (Algorithm 2) and logical learner (Algorithm 3) are only used for analysis, so they are not included.

4 Initialization

Initialization has two purposes: establish a common table row and ensure that every follower is active before the first learning assignment. This section bounds its cost by positive reward support; the bootstrap communication rules are in Appendix B. We use the per-round support g_t and cumulative support $G(t)$ defined in (5).

The public table has $M = N^2$ rows. For row j , write $C_{j,t}$ for its reserved-round indicator at time t and R_j for its independent learning/control symbols. Let C_A be a sufficiently large universal constant, whose required size is fixed in the proof of Lemma 29, and set $A_{\text{key}} := \lceil C_A N \rceil$. This is the authentication-key alphabet size. After publication, each follower draws an independent key $U_i \sim \text{Unif}([A_{\text{key}}])$, and the root independently draws $J \sim \text{Unif}([M])$. The selected learning/control row is R_J .

The uniform draws in the algorithms below and in Appendix B specify the exact distributions used in the proofs. Their finite implementation uses the following bounded sampler. Set

$$a_{\text{cap}} := \lceil 45 \log_2 N \rceil. \quad (6)$$

For a set S of size $q > 1$, $\text{BUnif}(S)$ makes at most a_{cap} independent attempts. In each attempt it draws $\lceil \log_2 q \rceil$ fair bits, interprets them as $X \in \{0, \dots, 2^{\lceil \log_2 q \rceil} - 1\}$, and accepts the $(X + 1)$ -th element of S if $X < q$. For $q = 1$ it returns the sole element without drawing bits. If all attempts are rejected, it reports *exhaustion*; a private call returns the first element of S , whereas the public constructor publishes the fixed-arm fallback flag and stops. All calls use fresh independent bits. The root index and follower keys, as well as the uniform choices during bootstrap, are implemented by these calls. Each call is coupled to its exact-uniform counterpart by the same rejection stream; their outputs agree unless the cap is exhausted. The finite public constructor obtains a Bernoulli variable with rational parameter r/s by testing whether $\text{BUnif}([s]) \leq r$; it generates permutations by bounded Fisher–Yates exchanges [Dur64].

4.1 Activation and launch

Let C_0 be a sufficiently large universal constant, whose required size is fixed in the proof of Lemma 29. Set

$$n := \lceil C_0^3 \sqrt{KT} \rceil,$$

the number of rounds in one bootstrap block. For each complete block b , write

$$I_b := \{(b-1)n + 1, \dots, bn\}. \quad (7)$$

Let $\mathcal{L}_{i,b} \subseteq I_b$ be player i 's listening positions, defined in Appendix B.3. When player i declares a row, write $\widehat{J}_i$ for its index; the root has $\widehat{J}_0 = J$.

Set

$$p := \min\{1/4, 1/\lceil \sqrt{T} \rceil\}, \quad \alpha_0 := 2^{-40}.$$

Here p is the probability that a public row reserves a given round for bootstrap communication, while α_0 is the probability that an array column listens at a given position. We call a player *informed* exactly when $\widehat{J}_i \neq \perp$. An informed player's local clock is

$$C_i(bn) = \frac{K}{\alpha_0 p} \sum_{b' \leq b} \sum_{t \in I_{b'}} \mathbf{1}\{t \in \mathcal{L}_{i,b'}\} C_{\widehat{J}_i,t} \mathbf{1}\{Y_{i,t} > 0\}. \quad (8)$$

The sum includes observations made before $\widehat{J}_i$ was known.

Let

$$b_N := \lceil \log_2(N+1) \rceil, \quad V := K^{5/2} \sqrt{T} \Lambda^2, \quad \widehat{V} := K^2 \lceil \sqrt{KT} \rceil b_N^2.$$

The quantity V is the target regret scale, and $\widehat{V} \asymp V$ is an integer-valued proxy convenient for parameter choices. Define the follower activation threshold for the support-estimating clock

$$Q_{\text{boot}} := C_0^8 \left(\widehat{V} + K b_N^2 / p \right), \quad (9)$$

and fix an integer $c_{\text{act}} > 128e + 1$. The root uses the larger threshold $c_{\text{act}} Q_{\text{boot}}$; the gap guarantees that the followers activate before the root launches.

Algorithm 1 Activation and launch at player i

```

1: Initially inactive; only the root knows  $J$ .
2: for each bootstrap block in calendar order do
3:   Choose its bootstrap message, then a fresh private array column, by Algorithm 12.
4:   While inactive, play its array arm on every round.
5:   While active, play its array arm on declared reserved rounds and use the learning protocol on other rounds.
6:   Store the bootstrap observations; when active, use the sparse DATA list of Section 7.3.
7:   if the block is complete with endpoint  $t$  then
8:     Decode the block, update digits and child lists, and compute  $C_i(t)$  if informed.
9:     if  $t < T$ , informed, and inactive then
10:      if  $i \neq 0$  and  $C_i(t) \geq Q_{\text{boot}}$  then
11:        Activate from round  $t + 1$ .
12:      else if  $i = 0$  and  $C_0(t) \geq c_{\text{act}} Q_{\text{boot}}$  then
13:        Activate from round  $t + 1$ .
14:      end if
15:    end if
16:  end if
17: end for

```

Lemma 2 (The root launches after every follower is active). *Suppose every player knows J whenever any player activates. For this lemma, define $C_i(t)$ by (8) with J in place of $\widehat{J}_i$, even before player i is informed. Assume that, at every complete-block boundary t , these counts satisfy*

$$\frac{G(t)}{8e} - \frac{Q_{\text{boot}}}{2} \leq C_i(t) \leq 8G(t) + \frac{Q_{\text{boot}}}{2} \quad (0 \leq i < m). \quad (10)$$

If the root activates after boundary t , every follower is active from round $t + 1$. The support before root activation, or throughout the horizon if it never activates, is at most

$$\lceil 8e(c_{\text{act}} + 1/2) + 1 \rceil Q_{\text{boot}},$$

provided $Kn \leq Q_{\text{boot}}$.

Proof. The root's threshold and the clock upper bound give

$$\begin{aligned}
G(t) &\geq \frac{c_{\text{act}} - 1/2}{8} Q_{\text{boot}} \\
&> 16e Q_{\text{boot}}.
\end{aligned}$$

For every follower, the lower bound then gives

$$C_i(t) \geq \frac{G(t)}{8e} - \frac{Q_{\text{boot}}}{2} > \frac{3}{2}Q_{\text{boot}}.$$

Every follower is informed under the hypothesis on declarations, so the activation rule has activated it by the next round.

For the support bound, let s be the first complete boundary with $G(s) \geq 8e(c_{\text{act}} + 1/2)Q_{\text{boot}}$, if it exists. At this boundary the root's lower clock bound gives $C_0(s) \geq c_{\text{act}}Q_{\text{boot}}$. If $s < T$, the root therefore activates by $s + 1$. The first-crossing property and the bound of Kn on one block's support give

$$G(s) \leq 8e(c_{\text{act}} + 1/2)Q_{\text{boot}} + Kn.$$

This also bounds the full horizon when $s = T$. If no such complete boundary exists, the support in complete blocks is below the crossing level, and the final incomplete block adds at most Kn . In both cases $Kn \leq Q_{\text{boot}}$ proves the claim. $\square$

Let

$$k_{\star} := \min\{m, K - m\}.$$

Let C_b be a sufficiently large universal branch constant, chosen last in the proof of Theorem 1. We call

$$k_{\star}T > C_b\widehat{V} \tag{11}$$

the *construction branch*; on the complementary branch the players use a fixed assignment to m distinct arms.

The initialization guarantee also uses the following array parameters. Fix a rational constant $0 < c_{\epsilon} \leq 1$, let C_S be a sufficiently large universal constant fixed in the proof of Lemma 29, and write $d := m - 1$. Define

$$L := \lceil C_0 K b_N \rceil, \quad e_{\star} := \lceil C_0^4 K^2 b_N^2 \rceil, \quad \ell_{\text{col}} := \lceil L/K \rceil.$$

Here L is the candidate-list scale, $e_{\star}$ is the positive-observation quota required before raw bootstrap decoding, and ℓ_{col} is the disjoint-column scale used in the array arguments. Set

$$\begin{aligned} \epsilon_{\text{arr}} &:= c_{\epsilon} \frac{K^3 b_N^2}{m^3 T}, \\ \Gamma &:= Kn + mb_N + b_N, \\ S_{\text{col}} &:= \left\lceil C_S \max \left\{ \frac{d \ell_{\text{col}} \Gamma}{\epsilon_{\text{arr}}}, \frac{m^2 K b_N}{\alpha_0} \right\} \right\rceil. \end{aligned} \tag{12}$$

The quantity ϵ_{arr} is the allowed exceptional-column fraction, Γ is the input-count scale used in the uniform union bounds, and S_{col} is the number of public array columns. Finally, the finite public control sampler uses

$$L_{\text{bit}} := \lceil 64 \log_2 N \rceil$$

bits per sampler coordinate. Let

$$B_{\text{blk}} := \lfloor T/n \rfloor$$

denote the number of complete bootstrap blocks. The exact-uniform private draws in Algorithm 12, including the initial choices of J and U_i , are realized by the bounded sampler of (6).

Proposition 3 (Constructive initialization). *Assume the construction branch (11), and fix public masks and arm arrays satisfying Lemmas 33 and 29. The probability below is over the root index and players' private randomness. For every deterministic reward matrix, set*

$$\Delta_{\text{boot}} = Cm^2 B_{\text{blk}}(\epsilon_{\text{arr}} + d\ell_{\text{col}}/S_{\text{col}} + L/A_{\text{key}}) + N^{-2} + N^{-1} + N^{-40} + CmTe^{-C_0\Lambda}. \quad (13)$$

There is a reference initialization, coupled to the actual protocol by the same root index, private keys, and bootstrap columns, and an event $\mathcal{E}_{\text{boot}}$ with $\Pr(\mathcal{E}_{\text{boot}}^c) \leq \Delta_{\text{boot}}$, such that:

- (i) *On $\mathcal{E}_{\text{boot}}$, the two initializations have identical bootstrap states and activation times. If any player activates, then every player has recovered J and every child-key list is initialized before that activation. Every follower is active when the root launches.*
- (ii) *On every reference path, each follower is active when the root launches, and the support before that launch is at most $[8e(c_{\text{act}} + 1/2) + 1]Q_{\text{boot}}$. Conditional on the reward matrix, masks, and arrays, the reference bootstrap states and activation times are independent of all learning/control rows. Every control-row access in the reference execution uses R_J .*

Proof. Recovery before activation. Call a complete block informative when its support is at least $128Ke_*/\alpha_0$. Apply Proposition 32 with $\delta_{\text{prop}} = N^{-2}$, and write its required number of informative blocks as

$$r_{\text{boot}} = 1 + \lceil 8L(2\lfloor \log_2 m \rfloor + \log(mN^2)) \rceil = O(K\Lambda^2).$$

At any complete-block boundary preceding the r_{boot} th informative block, there are at most $r_{\text{boot}} - 1$ informative blocks. Each such block contributes at most Kn support, and each other block contributes less than $128Ke_*/\alpha_0$. Therefore

$$G(t) \leq H_{\text{boot}} := \frac{128Ke_*T}{\alpha_0 n} + (r_{\text{boot}} - 1)Kn < Q_{\text{boot}}/16. \quad (14)$$

The definitions of e_* , L , and n give $H_{\text{boot}} = O(V)$; the choice of the universal constant C_0 gives the strict inequality and also $Kn \leq Q_{\text{boot}}$.

Reference construction. Let s_f be the first complete boundary with $G(s_f) \geq 16eQ_{\text{boot}}$, and let s_r be the first complete boundary with $G(s_r) \geq 8e(c_{\text{act}} + 1/2)Q_{\text{boot}}$. Set either boundary to $+\infty$ if it does not exist. Let t_a be the first boundary at which the actual activation rule activates a player, or $+\infty$ if none exists, and put $\tau := \min\{s_f, t_a\}$. Since an activation takes effect only in the next round, all actions through τ are bootstrap actions. In particular, no learning/control row is read before the correction at τ . As in Appendix B.2, write $J - 1 = J_0 + A_{\text{key}}J_1$. Here $\widehat{J}_{i,k}$ is player i 's estimate of digit J_k , $\mathcal{H}_{ij}$ is player i 's stored key list for child j , and $\perp$ marks an unknown digit or absent list. The following auxiliary procedure defines the reference on every sample path, including paths on which bootstrap recovery fails.

Algorithm 2 True-row reference correction (proof only)

```
1: Copy actual bootstrap updates through  $\tau$ .
2: if  $\tau < +\infty$  then
3:   Before any control-row access, set  $\widehat{J}_{i,k} = J_k$  for every  $i$  and  $k \in \{0, 1\}$ .
4:   for each parent-child edge  $(i, j)$  do
5:     if  $\mathcal{H}_{ij} = \perp$  or  $U_j \notin \mathcal{H}_{ij}$  then
6:       Set  $\mathcal{H}_{ij} = \{U_j\}$ .
7:     end if
8:   end for
9:   Recompute every clock from saved observations with mask  $C_J$ .
10:  Preserve activation flags already set; continue the local activation rule using the corrected clocks.
11:  Continue bootstrap forwarding with the corrected digits and child-key lists.
12: end if
13: if  $s_f < T$  then
14:   Activate any remaining inactive followers from  $s_f + 1$ .
15: end if
16: if  $s_r < T$  then
17:   Activate the root from  $s_r + 1$  if still inactive, and launch as in Algorithm 1.
18: end if
19: Whenever the root activates, activate every remaining inactive follower from the same next round.
```

Let $\mathcal{C}_{\text{priv}}$ be the set of bounded private sampler calls made during initialization and bootstrap, including the root-index and authentication-key calls, and define

$$\mathcal{E}_{\text{cap}} := \bigcap_{c \in \mathcal{C}_{\text{priv}}} \{\text{call } c \text{ accepts within } a_{\text{cap}} \text{ attempts}\}. \quad (15)$$

Singleton calls satisfy this event automatically.

Probability of agreement. Couple each capped private draw to an uncapped uniform draw using the same rejection stream. Define the events in the following probability estimates using these uncapped draws; their states agree with the implemented protocol on $\mathcal{E}_{\text{cap}}$. Let $\mathcal{E}_{\text{mask}}$ be the event that the uncapped root row J satisfies (100) for the fixed reward support. Property (ii) of Lemma 33 and the uniform choice of J give $\Pr(\mathcal{E}_{\text{mask}}^c) \leq 1/N$. The reference uses its array action on every true reserved round, and each block's message is fixed before its fresh column is drawn. Thus Lemma 34 applies on $\mathcal{E}_{\text{mask}}$. Let $\mathcal{E}_{\text{clk}}$ be the event that (10) holds for every player and complete boundary in the uncapped reference. Then

$$\Pr(\mathcal{E}_{\text{clk}}^c \cap \mathcal{E}_{\text{mask}}) \leq C m T e^{-C_0 \Lambda}.$$

Let $\mathcal{E}_{\text{rec}}$ be the event of Proposition 32 for the all-bootstrap auxiliary execution, coupled to the actual and reference executions by their common private choices. That proposition, with $\delta_{\text{prop}} = N^{-2}$, yields

$$\Pr(\mathcal{E}_{\text{rec}}^c) \leq C m^2 B_{\text{blk}}(\epsilon_{\text{arr}} + d\ell_{\text{col}}/S_{\text{col}} + L/A_{\text{key}}) + N^{-2}.$$

To bound $\Pr(\mathcal{E}_{\text{cap}}^c)$, observe that a draw from a set of size q succeeds in one rejection attempt with probability at least $1/2$; this is also true for $q = 1$, when no rejection is needed. Algorithm 12 makes at most four private draws per player and block. Including the initial keys and root index, the number of draws is at most $4m\lceil T/n \rceil + m \leq N$. Therefore

$$\Pr(\mathcal{E}_{\text{cap}}^c) \leq N 2^{-a_{\text{cap}}} \leq N^{-44} \leq N^{-40}.$$

Define

$$\mathcal{E}_{\text{boot}} := \mathcal{E}_{\text{mask}} \cap \mathcal{E}_{\text{clk}} \cap \mathcal{E}_{\text{rec}} \cap \mathcal{E}_{\text{cap}}.$$

The union bound gives $\Pr(\mathcal{E}_{\text{boot}}^c) \leq \Delta_{\text{boot}}$.

We prove pathwise agreement on $\mathcal{E}_{\text{boot}}$. Through τ , the actual, reference, and all-bootstrap executions have the same bootstrap actions and observations; induction over the completed blocks gives the same digit estimates and child-key lists. Property (i) of Proposition 32 makes every declared row equal to J . Hence the clock used at a first activation boundary t equals its true-row clock. Since the smallest activation threshold is Q_{boot} , the upper bound in (10) implies

$$Q_{\text{boot}} \leq 8G(t) + Q_{\text{boot}}/2, \quad G(t) \geq Q_{\text{boot}}/16.$$

By (14), t cannot precede the r_{boot} th informative block. The same conclusion holds if $\tau = s_f$, because $G(s_f) \geq 16eQ_{\text{boot}}$. Proposition 32 therefore implies that, whenever $\tau < +\infty$, every player knows J and every child list contains the true key at τ . The corrections in Algorithm 2 then change no bootstrap field or clock on $\mathcal{E}_{\text{boot}}$.

At $s_f < T$, the lower clock bound gives, for every follower,

$$C_i(s_f) \geq \frac{16eQ_{\text{boot}}}{8e} - \frac{Q_{\text{boot}}}{2} = \frac{3}{2}Q_{\text{boot}} > Q_{\text{boot}}.$$

At $s_r < T$, it gives

$$C_0(s_r) \geq \frac{8e(c_{\text{act}} + 1/2)Q_{\text{boot}}}{8e} - \frac{Q_{\text{boot}}}{2} = c_{\text{act}}Q_{\text{boot}}.$$

Thus both forced activations coincide with decisions of the actual activation rule. If the root activates earlier, the hypotheses of Lemma 2 hold, so its followers are already active from the next round. Induction over the remaining block boundaries shows that the actual and reference bootstrap states and activation times coincide on $\mathcal{E}_{\text{boot}}$.

Pathwise support and independence. Put $H_r = 8e(c_{\text{act}} + 1/2)Q_{\text{boot}}$. At its first crossing boundary s_r , if one exists, the increase in support over the preceding complete boundary is at most Kn ; hence $G(s_r) \leq H_r + Kn$. If there is no such boundary, the complete blocks contribute less than H_r , and the final incomplete block contributes at most Kn . The reference root activates no later than $s_r + 1$ when $s_r < T$. Consequently, on every reference path, the support before root activation, or over the full horizon if it never activates, is at most $H_r + Kn \leq [8e(c_{\text{act}} + 1/2) + 1]Q_{\text{boot}}$. The final rule of Algorithm 2 activates every follower no later than the root's launch.

Fix the reward matrix, masks, and arrays. Let $\mathcal{F}_{\text{boot}}$ be generated by J , the follower keys, and all private bootstrap randomness; it is independent of the learning/control rows. Through τ , only bootstrap actions occur, so τ and the bootstrap history through τ are $\mathcal{F}_{\text{boot}}$ -measurable. The reference correction, when needed, and subsequent bootstrap updates use only this information. Induction over blocks thus makes every reference bootstrap state and activation time $\mathcal{F}_{\text{boot}}$ -measurable. The correction precedes the first control-row access, and every such access uses R_J . $\square$

5 The slow-fast learner

We analyze the slow-fast learner in the product-stream execution of Section 3.

5.1 A set distribution with one fast arm

The coordinator is player 0, and the slow players are $1, \dots, m-1$. For positive arm weights w , define

$$\begin{aligned}\mathcal{Z}(w) &= e_m(w) = \sum_{|S|=m} \prod_{i \in S} w_i, \\ Q_w(S) &= \frac{\prod_{i \in S} w_i}{\mathcal{Z}(w)}, \\ \theta_w(i) &= \sum_{S \ni i} Q_w(S).\end{aligned}$$

Here e_k denotes the degree- k elementary symmetric sum.

At the start of an active episode, freeze the current weights as the anchor $\bar{w}$. A later assignment does not change $\bar{w}$ or the recording rates below.

After drawing an m -set S , choose its fast arm j with probability

$$\alpha_j(S) = \frac{\bar{w}_j^{-1}}{\sum_{i \in S} \bar{w}_i^{-1}}. \quad (16)$$

The remaining set $U = S \setminus \{j\}$ has $m-1$ arms, assigned in increasing arm-label order to the slow players. Let

$$p_w(U) = \sum_{j \notin U} Q_w(U \cup \{j\}) \alpha_j(U \cup \{j\}).$$

Writing $s_U = \sum_{i \in U} \bar{w}_i^{-1}$, the conditional law of the fast arm is

$$q_w(j | U) = \frac{w_j / (1 + s_U \bar{w}_j)}{\sum_{k \notin U} w_k / (1 + s_U \bar{w}_k)}. \quad (17)$$

Lemma 4 (Polynomial marginal bounds). *Suppose $\bar{w}_i \leq w_i \leq 2\bar{w}_i$ for every i . Then, for every $(m-1)$ -set U and $i \notin U$,*

$$\begin{aligned}q_w(i | U) &\geq \frac{\theta_{\bar{w}}(i)}{2(K-m+1)}, \\ \theta_w(i) &\leq 2\theta_{\bar{w}}(i).\end{aligned} \quad (18)$$

Proof. Put $v = \bar{w}$, fix $i \notin U$, and write $s = \sum_{b \in U} v_b^{-1}$ and $g_U(x) = x/(1+sx)$. For any m -set B avoiding i , retaining the products containing b in $e_m(v_{-i})$ gives

$$\begin{aligned}e_m(v_{-i}) \sum_{b \in B} v_b^{-1} &\geq \sum_{b \in B} e_{m-1}(v_{-\{i,b\}}) \\ &= \sum_{\substack{V \subseteq [K] \setminus \{i\} \\ |V|=m-1}} |B \setminus V| \prod_{k \in V} v_k \\ &\geq e_{m-1}(v_{-i}).\end{aligned}$$

The last inequality uses $|B \setminus V| \geq 1$. For $j \notin U$ with $j \neq i$, take $B = U \cup \{j\}$ and set $R = v_i(s + v_j^{-1})$. The inclusion odds of i are at most R , so

$$\theta_v(i) \leq \frac{R}{1+R} \leq \frac{R}{1+sv_i} = \frac{g_U(v_i)}{g_U(v_j)}.$$

For $j = i$, the last comparison follows from $\theta_v(i) \leq 1$. Summing $\theta_v(i)g_U(v_j) \leq g_U(v_i)$ over the $K - m + 1$ outside arms gives

$$q_v(i | U) = \frac{g_U(v_i)}{\sum_{j \notin U} g_U(v_j)} \geq \frac{\theta_v(i)}{K - m + 1}.$$

In (17), replacing v by w multiplies each unnormalized mass by $w_j/v_j \in [1, 2]$. Thus $q_w(i | U) \geq q_v(i | U)/2$, which proves the first inequality in (18).

Finally, the inclusion odds of i are $\theta_w(i)/(1 - \theta_w(i)) = w_i e_{m-1}(w_{-i})/e_m(w_{-i})$. Fix $j \neq i$ and let $E_k = e_k(w_{-\{i,j\}})$. Holding all other weights fixed, these odds equal

$$w_i \frac{E_{m-1} + w_j E_{m-2}}{E_m + w_j E_{m-1}}.$$

Their derivative with respect to w_j has the sign of $E_{m-2}E_m - E_{m-1}^2 \leq 0$ by Newton's inequalities for elementary symmetric sums (Hardy, Littlewood, and Pólya [HLP52]). Thus increasing another arm's weight cannot increase inclusion of i . Increasing w_i by at most a factor two increases its inclusion probability by at most two. Increase the other weights first and then w_i to obtain $\theta_w(i) \leq 2\theta_{\bar{w}}(i)$. $\square$

5.2 A sparse estimate and infrequent refreshes

Recall $N = 2Km(T + 1)$. Let $L_{\text{stab}} = \lceil \log_2(2K(T + 1)N^8) \rceil$ be the concentration scale that keeps weights stable within an episode. Set $A = 256$ to be a fixed constant. Let η be the learning rate, subject to the cap below; it determines β and H :

$$\begin{aligned} 0 < \eta &\leq (2AmKL_{\text{stab}})^{-1}, \\ \beta &= 2\eta L_{\text{stab}}, \\ H &= \lfloor (16\eta)^{-1} \rfloor. \end{aligned}$$

Here β is a lower bound on recording rates, and H bounds the episode length. An active episode ends at the next multiple of H , clipped at $T + 1$. If collection is still unfinished, updates remain frozen until the next commitment starts a new active episode. At each episode start, use the frozen anchor $\bar{w}$ to set

$$\begin{aligned} \xi_i &= 2^{\lceil \log_2 \max\{\beta, \theta_{\bar{w}}(i)/(AmK)\} \rceil}, \\ \gamma &= \frac{A}{4}(K - m + 1)\eta L_{\text{stab}}, \\ q^\gamma &= (1 - \gamma)q + \gamma \text{Unif}([K] \setminus U). \end{aligned} \tag{19}$$

Here ξ_i is the recording rate for arm i , and γ is the uniform-exploration probability. The law $q^\gamma(\cdot | U)$ mixes $q_w(\cdot | U)$ with the uniform law on arms outside U . If $\bar{w} \leq w \leq 2\bar{w}$, the rates satisfy

$$\begin{aligned} \beta &\leq \xi_i \leq \frac{2}{AmK}, \\ q^\gamma(i | U) &\geq 16\xi_i \quad (i \notin U), \\ \frac{\theta_w(i)}{\xi_i} &\leq 2AmK, \\ \sum_i \frac{\theta_w(i)}{\xi_i} &\leq 2AmK^2. \end{aligned} \tag{20}$$

To check these bounds, dyadic rounding gives the first line. Lemma 4 and $\gamma \leq (K - m + 1)/(8mK) \leq 1/16$ give, for $i \notin U$,

$$\begin{aligned} (1 - \gamma)q_w(i | U) &\geq \frac{\theta_{\bar{w}}(i)}{8mK} = 32 \frac{\theta_{\bar{w}}(i)}{AmK}, \\ \frac{\gamma}{K - m + 1} &= 32\beta. \end{aligned}$$

The two terms of the mixture therefore give $q^\gamma(i \mid U) \geq 16\xi_i$. The remaining bounds follow from $\theta_w(i) \leq 2\theta_{\bar{w}}(i)$ and $\xi_i \geq \theta_{\bar{w}}(i)/(AmK)$. Also $\sum_i \xi_i \leq 2/(Am) \leq 1$, as required for the sparse law below.

Recall the bit-length scale $b_N = \lceil \log_2(N+1) \rceil$ from Section 4.1. Use $D = 2^{2b_N}$ as the reward quantization denominator: round each reward down to a multiple of $1/D$ and denote the result by $r_t(i)$. This changes regret by at most mT/D . Store integer scores z_i , initially zero, and set $w_i = (1 + 1/D)^{z_i}$.

Fix the oblivious reward matrix and the reserved rounds. In the product-stream experiment, the prescribed symbols at distinct rounds are independent, with the laws specified by the protocol. Each logical refresh uses a uniform variable independent of the history through its current estimate. Let $\mathcal{F}_t$ be the history immediately before the current round's symbols are exposed, including the scores, the current slow set U_t , and all preceding observations and decisions. In particular, the recording rates and whether learning is enabled are $\mathcal{F}_t$ -measurable. The current fast-arm draw is not included.

On an active nonreserved round with learning unfrozen, let I_t be the arm selected for recording, or $\perp$ if none is selected. Its required conditional law is

$$\begin{aligned} \Pr(I_t = i \mid \mathcal{F}_t) &= \xi_i \quad (i \in [K]), \\ \Pr(I_t = \perp \mid \mathcal{F}_t) &= 1 - \sum_i \xi_i. \end{aligned} \tag{21}$$

The rates are frozen for the episode, so this law is the same for every slow set. If $I_t = i$, its quantized estimate is

$$\tilde{r}_t(i) = \frac{1}{\eta D} \left\lfloor \frac{D\eta r_t(i)}{\xi_i} \right\rfloor. \tag{22}$$

All other coordinates of $\tilde{r}_t$ are zero. If no arm is selected, or the round is inactive, reserved, or frozen, then $\tilde{r}_t = 0$. When i is selected, add $\lfloor D\eta r_t(i)/\xi_i \rfloor$ to z_i . Since $\eta/\xi_i \leq 1/(2L_{\text{stab}})$, each increment is at most D and every score lies in $[0, TD]$.

Write $p_t = p_{w_t}$ for the slow-set distribution before round t . Set $h = 4AmK\eta$ as the probability of proposing a new slow set after a score update. The learning-rate cap gives $h \leq 1/4$. The proposed set has distribution

$$\rho_t(U) = \frac{p_{t+1}(U) - (1-h)p_t(U)}{h}. \tag{23}$$

Lemma 5 below shows that ρ_t is a probability distribution while the weights remain stable. After the score update on each active nonreserved round with learning unfrozen, the logical learner uses a bit $B_t \sim \text{Bernoulli}(h)$, independent of the history through that update; set $B_t = 0$ on all other rounds. The event $B_t = 1$ is a *proposal*. On an unfrozen round, the logical learner draws its next slow set from ρ_t if $B_t = 1$, and retains the current set if $B_t = 0$. During a live assignment this is the bit supplied by the public row. During collection the logical learner uses independent auxiliary bits, while the corresponding public coordinates remain unread. Section 7 integrates these logical refreshes into the distribution of the next assignment. No proposal is used on a frozen round.

We call this the *logical learner*: its scores and slow set change at the prescribed update and proposal times, without waiting for messages.

Algorithm 3 Logical slow-fast learner

```
1: Initialize  $z_i = 0$  and  $w_i = 1$  for every arm  $i$ .
2: for each active episode of at most  $H$  rounds do
3:   Freeze  $\bar{w} = w$ ; compute the rates  $\xi_i$  and mixing probability  $\gamma$ .
4:   Draw the slow set  $U \sim p_w$  and assign its arms to slow players in increasing order.
5:   for each active nonreserved round  $t$  with learning unfrozen do
6:     Draw the coordinator's fast arm  $j_t \sim q_{w_t}^\gamma(\cdot | U)$ .
7:     Obtain a recorded arm  $I_t$  with the conditional law (21).
8:     if  $I_t = i \in [K]$  then
9:       Form  $\tilde{r}_t(i)$  by (22) and add  $\lfloor D\eta r_t(i)/\xi_i \rfloor$  to  $z_i$ .
10:      Update  $w_i = (1 + 1/D)^{z_i}$ .
11:     end if
12:     Draw the proposal bit  $B_t \sim \text{Bernoulli}(h)$ .
13:     if  $B_t = 1$  then
14:       Draw the next slow set from  $\rho_t$ .
15:     end if
16:   end for
17: end for
```

For analysis on exceptional paths, use the following bounded continuation after the first weight-doubling update. Keep the logical scores fixed, set all subsequent logical estimates to zero, and perform no further logical refreshes. Whenever a later auxiliary reward is evaluated, use $U_t = \{1, \dots, m-1\}$ and $j_t = m$. This convention affects only paths with a stability failure and keeps each auxiliary reward in $[0, m]$; a decoder failure alone does not stop or alter the logical learner.

Lemma 5 (Stability and fixed refresh probability). *Outside an event of probability at most N^{-8} , every active episode satisfies $\bar{w} \leq w_t < 2\bar{w}$. On this event, each one-coordinate update obeys*

$$p_{t+1}(U) \geq (1-h)p_t(U). \quad (24)$$

Proof. For an episode starting at τ , let g be its next grid boundary and define the first weight-doubling round by

$$\tau_* = \inf\{t \in [\tau, g) : \max_i w_{t+1,i}/\bar{w}_i \geq 2\},$$

with $\inf \emptyset = +\infty$. Fix arm i and condition on the history immediately before the initial slow-set draw. Set

$$X_t(i) = \mathbf{1}\{\tau \leq t < g, t \leq \tau_*\} \log(w_{t+1,i}/w_{t,i}).$$

The indicator $\mathbf{1}\{t \leq \tau_*\}$ means that no earlier update has doubled a weight, so it is $\mathcal{F}_t$ -measurable. It includes the update at $t = \tau_*$ and excludes every later update. On a learning round with $t \leq \tau_*$, (20) gives $\xi_i \geq \beta = 2\eta L_{\text{stab}}$ and hence $0 \leq X_t(i) \leq \eta/\xi_i \leq 1/(2L_{\text{stab}})$. The event $I_t = i$ has conditional probability ξ_i by (21). The oblivious reward $r_t(i)$ and frozen rate ξ_i are $\mathcal{F}_t$ -measurable. Therefore, on an enabled round with $t \leq \tau_*$,

$$\begin{aligned} \mathbb{E}[X_t(i) | \mathcal{F}_t] &= \xi_i \log(1 + 1/D) \left\lfloor \frac{D\eta r_t(i)}{\xi_i} \right\rfloor \\ &\leq \eta r_t(i) \leq \eta. \end{aligned}$$

On other rounds $X_t(i) = 0$, so the same bounds hold. Convexity of the exponential gives

$$\mathbb{E}[e^{2L_{\text{stab}}X_t(i)} | \mathcal{F}_t] \leq 1 + 2(e-1)L_{\text{stab}}\mathbb{E}[X_t(i) | \mathcal{F}_t] \leq e^{2(e-1)L_{\text{stab}}\eta}.$$

An episode contains at most H rounds. Iterating this bound, with zero increments after the episode ends, gives

$$\mathbb{E} \exp \left(2L_{\text{stab}} \sum_t X_t(i) \right) \leq \exp(2(e-1)L_{\text{stab}}H\eta) \leq \exp((e-1)L_{\text{stab}}/8).$$

If arm i doubles during this episode, then $\sum_t X_t(i) \geq \log 2$. Markov's inequality bounds this probability by

$$\exp(-[2 \log 2 - (e-1)/8]L_{\text{stab}}) \leq 2^{-L_{\text{stab}}}.$$

By the definition of L_{stab} , this is at most $N^{-8}/(2K(T+1))$. A union bound over at most KT arm-episode pairs proves the stability claim.

For a one-arm update at i , write $x = \log(w_{t+1,i}/w_{t,i}) \leq \eta r_t(i)/\xi_i \leq 1$. Since $e^x - 1 \leq 2x$, (20) gives

$$\frac{\mathcal{Z}(w_{t+1})}{\mathcal{Z}(w_t)} = 1 + \theta_{w_t}(i)(e^x - 1) \leq 1 + 2\theta_{w_t}(i)\eta/\xi_i \leq 1 + h.$$

For every $j \notin U$, the weights only increase, so the preceding bound on $\mathcal{Z}(w_{t+1})/\mathcal{Z}(w_t)$ gives

$$\begin{aligned} \frac{Q_{w_{t+1}}(U \cup \{j\})}{Q_{w_t}(U \cup \{j\})} &= \frac{\mathcal{Z}(w_t)}{\mathcal{Z}(w_{t+1})} \prod_{i \in U \cup \{j\}} \frac{w_{t+1,i}}{w_{t,i}} \\ &\geq \frac{1}{1+h}. \end{aligned}$$

The coefficients $\alpha_j(U \cup \{j\})$ depend only on the frozen anchor $\bar{w}$ and are nonnegative. Summing the last inequality yields

$$\begin{aligned} p_{t+1}(U) &= \sum_{j \notin U} \alpha_j(U \cup \{j\}) Q_{w_{t+1}}(U \cup \{j\}) \\ &\geq \frac{1}{1+h} \sum_{j \notin U} \alpha_j(U \cup \{j\}) Q_{w_t}(U \cup \{j\}) \\ &= \frac{p_t(U)}{1+h} \geq (1-h)p_t(U). \end{aligned}$$

□

By (24), every term of the proposed law (23) is nonnegative; its terms sum to one. Thus the refresh step in Algorithm 3 is valid on the stability event.

5.3 Learning bound

For an episode starting at τ , let $\mathcal{H}_\tau$ be the full history immediately before its initial slow-set draw. Its next grid boundary $g = \min\{T+1, H(\lfloor \tau/H \rfloor + 1)\}$ and recording rates are fixed under this conditioning. Define the smaller history

$$\mathcal{E}_t = \sigma(\mathcal{H}_\tau, \tilde{r}_\tau, \dots, \tilde{r}_{t-1}), \quad \tau \leq t < g.$$

Thus $\mathcal{E}_t$ records the episode's initial history and subsequent estimates, and omits subsequent proposals and communication observations. The scores and p_t are $\mathcal{E}_t$ -measurable.

Lemma 6 (Conditional slow-set law). *For each round t of the episode,*

$$\Pr(U_t = U \mid \mathcal{E}_t) = p_t(U) \quad \text{on } \{t \leq \tau_*\}.$$

The event that no previous weight doubling has occurred is $\mathcal{E}_t$ -measurable. The identity therefore applies on the weight-doubling round itself.

Proof. The initial draw gives the identity at $t = \tau$. Suppose it holds at t and that no earlier update doubled a weight. The law in (21), followed by the deterministic formula (22), defines a probability kernel K_t for $\tilde{r}_t$. This kernel depends only on the frozen rates, the fixed reward vector, and whether the round is enabled. All these quantities are $\mathcal{E}_t$ -measurable. In particular, for every set C of possible estimates,

$$\Pr(\tilde{r}_t \in C \mid \mathcal{F}_t) = K_t(C).$$

For each slow set U , the tower property now gives

$$\begin{aligned} \Pr(U_t = U, \tilde{r}_t \in C \mid \mathcal{E}_t) &= \mathbb{E}[\mathbf{1}\{U_t = U\}K_t(C) \mid \mathcal{E}_t] \\ &= p_t(U)K_t(C). \end{aligned}$$

Consequently revealing the estimate leaves the conditional law of U_t equal to p_t . On an enabled nonreserved round, the proposal bit is independent of the history through that estimate. Its refresh draw has law ρ_t and is independent of the preceding slow set. Therefore

$$\begin{aligned} \Pr(U_{t+1} = U \mid \mathcal{E}_{t+1}) &= (1 - h)p_t(U) + h\rho_t(U) \\ &= p_{t+1}(U). \end{aligned}$$

On a reserved round the estimate is zero and neither distribution changes. This proves the induction until the first doubling. Whether an earlier update doubled a weight is determined by the estimates already included in $\mathcal{E}_t$. Restricting to this event therefore does not add information about U_t beyond $\mathcal{E}_t$, which proves the last assertion. $\square$

Let $\mathcal{A}$ be the rounds of the product-stream execution on which learning is active, the round is nonreserved, and score updates are unfrozen. Before the first weight-doubling update, the logical learner's U_t changes immediately at each proposal, as in Algorithm 3, and on $t \in \mathcal{A}$ its fast arm j_t is drawn from $q_{w_t}^\gamma(\cdot \mid U_t)$. Afterwards use the bounded continuation specified above. For a fixed comparator m -set S , define

$$R_{\text{aux}}(S) = \mathbb{E} \sum_{t \in \mathcal{A}} (r_t(S) - r_t(U_t) - r_t(j_t)).$$

Proposition 7 (Auxiliary learning bound). *For any fixed m -set S chosen independently of the players' randomness,*

$$R_{\text{aux}}(S) \leq \frac{2 \log \binom{K}{m}}{\eta} + (2AmK^2 + \frac{A}{4}(K - m + 1)L_{\text{stab}})\eta T + \frac{mT}{\eta D} + CmTN^{-8}. \quad (25)$$

In particular, S may maximize the original oblivious reward matrix; it is not chosen after seeing the random active rounds.

Proof. In this proof, τ_* denotes the first weight-doubling round in any episode, with $\tau_* = +\infty$ if no weight doubles. The event $\{t \leq \tau_*\}$ is determined by the current episode's initial history and its estimates before t .

First relate the distribution of the played arms to Q_w . For $j \notin U$, (16) gives

$$\alpha_j(U \cup \{j\}) = \frac{1}{1 + s_U \bar{w}_j}.$$

Hence

$$Q_w(U \cup \{j\})\alpha_j(U \cup \{j\}) = \frac{\prod_{i \in U} w_i}{\mathcal{Z}(w)} \frac{w_j}{1 + s_U \bar{w}_j}.$$

The factor before the last fraction does not depend on j . Summing over $j \notin U$ gives $p_w(U)$, and normalizing gives (17). Consequently,

$$p_w(U)q_w(j | U) = Q_w(U \cup \{j\})\alpha_j(U \cup \{j\}).$$

If the fast arm is drawn from q_{w_t} without uniform mixing, then for every m -set S' ,

$$\begin{aligned} \Pr(U_t \cup \{j_t\} = S' | \mathcal{E}_t) &= \sum_{j \in S'} p_{w_t}(S' \setminus \{j\})q_{w_t}(j | S' \setminus \{j\}) \\ &= Q_{w_t}(S') \sum_{j \in S'} \alpha_j(S') \\ &= Q_{w_t}(S'). \end{aligned}$$

Its conditional expected reward is therefore

$$\begin{aligned} \mathbb{E}[r_t(U_t) + r_t(j_t) | \mathcal{E}_t] &= \sum_{S'} Q_{w_t}(S')r_t(S') \\ &= \sum_i \theta_{w_t}(i)r_t(i). \end{aligned}$$

The actual law $q_{w_t}^\gamma$ mixes the fast arm with a uniform draw with probability γ . This leaves U_t unchanged and can lower the expected reward by at most γ , because each arm's reward lies in $[0, 1]$. Thus, under the actual law,

$$\mathbb{E}[r_t(U_t) + r_t(j_t) | \mathcal{E}_t] \geq \sum_i \theta_{w_t}(i)r_t(i) - \gamma.$$

For an m -set S , write $\tilde{r}_t(S) = \sum_{i \in S} \tilde{r}_t(i)$. Put $\lambda = D \log(1 + 1/D)\eta \in [\eta/2, \eta]$. If arm i is recorded, its log-weight increment is $x = \lambda \tilde{r}_t(i) \in [0, 1]$. Hence

$$\log \frac{\mathcal{Z}(w_{t+1})}{\mathcal{Z}(w_t)} = \log(1 + \theta_{w_t}(i)(e^x - 1)) \leq \theta_{w_t}(i)(x + x^2).$$

The same inequality is trivial when no arm is recorded. Since at most one coordinate of $\tilde{r}_t$ is nonzero, summing the one-step bound and telescoping the logarithms gives

$$\begin{aligned} \log \frac{\mathcal{Z}(w_{T+1})}{\mathcal{Z}(w_1)} &= \sum_t \log \frac{\mathcal{Z}(w_{t+1})}{\mathcal{Z}(w_t)} \\ &\leq \lambda \sum_{t,i} \theta_{w_t}(i)\tilde{r}_t(i) + \lambda^2 \sum_{t,i} \theta_{w_t}(i)\tilde{r}_t(i)^2. \end{aligned}$$

On the other hand, the weight product for S is one term of $\mathcal{Z}(w_{T+1})$. Since $\mathcal{Z}(w_1) = \binom{K}{m}$ and $\prod_{i \in S} w_{T+1,i} = \exp(\lambda \sum_t \tilde{r}_t(S))$,

$$\begin{aligned} \log \frac{\mathcal{Z}(w_{T+1})}{\mathcal{Z}(w_1)} &\geq \log \frac{\prod_{i \in S} w_{T+1,i}}{\binom{K}{m}} \\ &= \lambda \sum_t \tilde{r}_t(S) - \log \binom{K}{m}. \end{aligned}$$

Combine the upper and lower bounds, subtract the linear term from both sides, and divide by λ to obtain

$$\sum_t \left(\tilde{r}_t(S) - \sum_i \theta_{w_t}(i)\tilde{r}_t(i) \right) \leq \frac{\log \binom{K}{m}}{\lambda} + \lambda \sum_t \sum_i \theta_{w_t}(i)\tilde{r}_t(i)^2.$$

The weights persist across episodes, so this pays $\log \binom{K}{m}$ only once. On an enabled round before a previous weight doubling, the recording law and the floor in the update give

$$\mathbb{E}[\tilde{r}_t(i) \mid \mathcal{F}_t] = \frac{\xi_i}{\eta D} \left\lfloor \frac{D\eta r_t(i)}{\xi_i} \right\rfloor,$$

$$r_t(i) - \frac{\xi_i}{\eta D} \leq \mathbb{E}[\tilde{r}_t(i) \mid \mathcal{F}_t] \leq r_t(i).$$

Likewise,

$$\mathbb{E}[\tilde{r}_t(i)^2 \mid \mathcal{F}_t] = \frac{\xi_i}{(\eta D)^2} \left\lfloor \frac{D\eta r_t(i)}{\xi_i} \right\rfloor^2$$

$$\leq \frac{r_t(i)^2}{\xi_i}.$$

Since $\theta_{w_t}(i) \geq 0$ and the weights are $\mathcal{F}_t$ -measurable, the first pair of bounds implies

$$r_t(S) - \sum_i \theta_{w_t}(i) r_t(i)$$

$$\leq \sum_{i \in S} \left(\mathbb{E}[\tilde{r}_t(i) \mid \mathcal{F}_t] + \frac{\xi_i}{\eta D} \right) - \sum_i \theta_{w_t}(i) \mathbb{E}[\tilde{r}_t(i) \mid \mathcal{F}_t]$$

$$= \mathbb{E} \left[r_t(S) - \sum_i \theta_{w_t}(i) \tilde{r}_t(i) \mid \mathcal{F}_t \right] + \sum_{i \in S} \frac{\xi_i}{\eta D}.$$

Here $\sum_{i \in S} \xi_i \leq m$, since $0 \leq \xi_i \leq 1$. The second-moment bound and (20) give

$$\mathbb{E} \left[\sum_i \theta_{w_t}(i) \tilde{r}_t(i)^2 \mid \mathcal{F}_t \right] \leq \sum_i \frac{\theta_{w_t}(i) r_t(i)^2}{\xi_i}$$

$$\leq \sum_i \frac{\theta_{w_t}(i)}{\xi_i} \leq 2AmK^2.$$

To sum the conditional reward inequalities over random episode starts, condition first on $\mathcal{H}_\tau$. If the episode starts at τ , its rounds with enabled estimates, before a stability stop, are exactly

$$\{t \in [T] : \tau \leq t < g, t \text{ is nonreserved}\}.$$

The displayed set is fixed under that conditioning; collection within the episode does not freeze estimates. Multiply its round- t summand by $\mathbf{1}\{t \leq \tau_*\}$. This indicator is $\mathcal{E}_t$ -measurable, so Lemma 6 still gives the conditional reward inequality on each retained round. At a later episode's random start, condition on its new $\mathcal{H}_\tau$ and apply the same argument. Summing these conditional expectations accounts for every retained round exactly once. No conditioning on successful decoding is used.

In the potential inequality, set all estimates after the first weight-doubling round to zero and keep the weights fixed thereafter. The potential calculation remains valid for this stopped sequence. Take its expectation. Write $\mathcal{A}_*$ for the rounds of $\mathcal{A}$ up to and including that first doubling round. The conditional reward bound just proved gives

$$\mathbb{E} \sum_{t \in \mathcal{A}_*} (r_t(S) - r_t(U_t) - r_t(j_t))$$

$$\leq \mathbb{E} \sum_{t \in \mathcal{A}_*} \left(r_t(S) - \sum_i \theta_{w_t}(i) r_t(i) \right) + \gamma T.$$

Here $|\mathcal{A}_*| \leq T$. Applying the conditional estimate comparison on each retained round and then the potential inequality gives

$$\begin{aligned}
& \mathbb{E} \sum_{t \in \mathcal{A}_*} \left(r_t(S) - \sum_i \theta_{w_t}(i) r_t(i) \right) \\
& \leq \mathbb{E} \sum_{t \in \mathcal{A}_*} \left(\tilde{r}_t(S) - \sum_i \theta_{w_t}(i) \tilde{r}_t(i) \right) + \frac{mT}{\eta D} \\
& \leq \frac{\log \binom{K}{m}}{\lambda} + \lambda \mathbb{E} \sum_{t \in \mathcal{A}_*} \sum_i \theta_{w_t}(i) \tilde{r}_t(i)^2 + \frac{mT}{\eta D} \\
& \leq \frac{2 \log \binom{K}{m}}{\eta} + 2AmK^2 \eta T + \frac{mT}{\eta D}.
\end{aligned}$$

For the last line we used the conditional second-moment inequality, $\lambda \leq \eta$, and $1/\lambda \leq 2/\eta$. Substituting $\gamma = \frac{A}{4}(K - m + 1)\eta L_{\text{stab}}$ into the conditional reward bound therefore yields

$$\begin{aligned}
& \mathbb{E} \sum_{t \in \mathcal{A}_*} (r_t(S) - r_t(U_t) - r_t(j_t)) \\
& \leq \frac{2 \log \binom{K}{m}}{\eta} + \left(2AmK^2 + \frac{A}{4}(K - m + 1)L_{\text{stab}} \right) \eta T + \frac{mT}{\eta D}.
\end{aligned}$$

For the first doubling round, its pre-round weights still satisfy $\bar{w}_i \leq w_{t,i} < 2\bar{w}_i$. Thus both conditional moment bounds and (20) apply to its update. Every omitted round belongs to an execution with a weight doubling. Since each omitted regret summand is at most m ,

$$R_{\text{aux}}(S) \leq \mathbb{E} \sum_{t \in \mathcal{A}_*} (r_t(S) - r_t(U_t) - r_t(j_t)) + mT \Pr(\text{some weight doubles}).$$

Lemma 5 bounds the last probability by N^{-8} . This completes the proof. $\square$

6 Assignment and score communication

Use the history $\mathcal{F}_t$ of Section 5 after any commitment at the start of round t and before its physical arm choices. It includes the resulting assignment and all previously used coordinates; it excludes the current slot, pilots, arm tuple, hash, fast-arm draw, thinning coordinate, and all still-unread proposal coordinates. Thus the local states and outgoing words are $\mathcal{F}_t$ -measurable. In the product-stream execution the current physical coordinates have their prescribed independent laws conditional on $\mathcal{F}_t$. Write $Y_{i,t}$ for the quantized observation, using the convention for r_t in Section 5. The slot τ_t on a nonreserved round has the following law.

Slot	Probability	Role
DATA	1/2	Play arms and possibly record one reward.
DOWN	1/4	Send an assignment and confirm that its recipient decoded it.
UP	1/4	Return a score, or certify that the coordinator received all assignment receipts.

6.1 Message decoding and support cost

On each DOWN or UP round, the row selects $k_t \in [d]$ uniformly and supplies an independent uniform injective arm tuple $(R_{0,t}, \dots, R_{d,t}) \in [K]^{d+1}$. Conditional on $k_t = i$, DOWN goes from the coordinator to player i ; UP goes from player i to the coordinator. Write R_j when the round is clear.

Every integer field uses $\lceil \log_2(TD + 1) \rceil$ bits, where D is the score quantization denominator of Section 5. A DOWN record has four fields; an UP record has three fields followed by a zero field. Both have length $B = 4\lceil \log_2(TD + 1) \rceil$. Section 7.2 gives the encoding and field bounds. For each round and each direction and selected player, independently draw $c_t \in \mathbb{F}_2^B$ and $b_t \in \mathbb{F}_2$, and define

$$H_t(u) = c_t \cdot u \oplus b_t.$$

On DOWN the sender and receiver arms are R_0 and R_i ; on UP they are R_i and R_0 . Hash zero forces a collision, so a positive receiver observation supplies the equation

$$c_t \cdot u = 1 \oplus b_t.$$

A zero observation supplies no equation.

Set $\delta = N^{-8}$ as the communication-error budget. The decoder uses batches of

$$Q_{\text{dec}} = B + \lceil \log_2(2N^{12}/\delta) \rceil \quad (26)$$

positive observations. Each directed channel stores one initially empty batch.

Call a message M^* *awaited* on a directed channel while the receiver has not accepted it and, on every selected active nonreserved round, the sender encodes M^* and the receiver runs DecodeStep.

Algorithm 4 One affine message channel

```

1: procedure SENDARM( $u, R_{\text{send}}, R_{\text{receive}}$ )
2:   if  $H_t(u) = 0$  then
3:     return  $R_{\text{receive}}$ .
4:   else
5:     return  $R_{\text{send}}$ .
6:   end if
7: end procedure
8: procedure DECODESTEP( $W, Y$ )
Require: An enabled receiver channel, its equation batch  $W$ , and the current observed reward  $Y$ .
9:   if  $Y > 0$  then
10:    Append  $c_t \cdot u = 1 \oplus b_t$  to  $W$ .
11:   end if
12:   if  $|W| < Q_{\text{dec}}$  then
13:     return  $\perp$ .
14:   end if
15:   Solve the system  $W$  over  $\mathbb{F}_2^B$ .
16:   Let  $M$  be its unique solution if one exists, and  $\perp$  otherwise.
17:   Set  $W \leftarrow \emptyset$ .
18:   return  $M$ .
19: end procedure

```

Lemma 8 (Message decoding). *Consider the product-stream protocol with correct initialization, stopped before a stability or feasibility failure. With probability at least $1 - \delta/N^4$, no completed batch has a unique solution different from the word currently encoded on its channel that passes (40), and no completed batch formed entirely from one fixed awaited word has multiple solutions. On this event, an awaited word is accepted within $2Q_{\text{dec}}$ positive receiver observations after its sender fixes that word, provided service continues. Each decoder decision takes $O(Q_{\text{dec}}B^2) = O(\log^3 N)$ bit operations.*

Appendix A.2 proves this lemma. A decoder failure is a wrong acceptance or a nonunique system in a complete batch whose observations all come from the fixed awaited word. A decoder failure contributes at most $mT\delta/N^4$ to expected regret.

The next lemma bounds the cumulative charge of parallel tasks when each pending task's conditional service probability is at least a fixed multiple of the round's charge and each task has a service quota. It is used for assignment delivery (Lemma 12), follower release (Lemma 13), and score collection (Lemma 21).

Fix one *service phase*. Let σ be its first round and ϑ its first excluded round, where $1 \leq \sigma \leq \vartheta \leq T + 1$ are stopping times for the pre-round histories $(\mathcal{F}_t)$. Thus a phase terminated after round t has $\vartheta = t + 1$. For $\sigma \leq t < \vartheta$, let $P_t \subseteq [d]$ be the $\mathcal{F}_t$ -measurable set of tasks still unresolved at the start of round t ; assume $P_{t+1} \subseteq P_t$ while the phase continues. A task may represent a fixed-message decoder, a receipt, or a release certificate. Task i is *pending* at t precisely when $i \in P_t$.

Define the predictable *eligibility* indicator

$$A_t = \mathbf{1}\{\sigma \leq t < \vartheta, t \text{ nonreserved}, P_t \neq \emptyset\}.$$

For task i , specify an event observable by the end of round t and let $X_{i,t} \in \{0, 1\}$ indicate its occurrence; set $X_{i,t} = 0$ if the slot or selected player cannot serve i . One occurrence $X_{i,t} = 1$ is a *service observation*, which need not itself complete the task. Write

$$J_{i,t} = A_t \mathbf{1}\{i \in P_t\}, \quad N_i(v) = \sum_{t=1}^v J_{i,t} X_{i,t}.$$

For t outside the phase set $P_t = \emptyset$. The phase's support charge is

$$G_{\text{wait}} = \sum_{t=1}^T A_t g_t,$$

where g_t is defined in (5).

We impose a positive integer *quota* q through the condition $J_{i,t} = 1 \implies N_i(t-1) < q$: a task cannot be pending on an eligible round after its q -th service observation.

Lemma 9 (Cost of parallel service). *For a phase as above, fix an integer $q \geq 1$, $a > 0$, and $\mathcal{F}_t$ -measurable $\omega_t \in [0, K]$. Suppose the quota q condition holds and, on $\{J_{i,t} = 1\}$,*

$$\mathbb{E}[X_{i,t} \mid \mathcal{F}_t] \geq a\omega_t$$

for every $i \in [d]$ and $t \leq T$. Then each task satisfies

$$\mathbb{E} \sum_{t=1}^T J_{i,t} \omega_t \leq \frac{q}{a}.$$

The total phase charge satisfies

$$\mathbb{E} \sum_{t=1}^T A_t \omega_t \leq \frac{q + \log d}{(1 - e^{-1})a}. \quad (27)$$

In particular, $\omega_t = g_t$ bounds $\mathbb{E}G_{\text{wait}}$. Both bounds also hold conditional on the phase-start history $\mathcal{F}_\sigma$.

Proof. The variables $J_{i,t}$ and ω_t are $\mathcal{F}_t$ -measurable. For $i \in [d]$ and $v \leq T$, set

$$G_i(v) = \sum_{t=1}^v J_{i,t} \omega_t.$$

The quota condition gives $N_i(T) \leq q$: a q -th success can be counted, but no later eligible round can count another. Therefore

$$\begin{aligned} a\mathbb{E}G_i(T) &\leq \sum_{t=1}^T \mathbb{E}[J_{i,t} \mathbb{E}[X_{i,t} \mid \mathcal{F}_t]] \\ &= \mathbb{E}N_i(T) \leq q. \end{aligned}$$

For the total charge, put $\kappa = 1 - e^{-1}$ and define

$$Z_i(v) = \exp(\kappa a G_i(v) - N_i(v)).$$

On $\{J_{i,t} = 1\}$, since $X_{i,t}$ is Bernoulli,

$$\mathbb{E}[e^{-X_{i,t}} \mid \mathcal{F}_t] = 1 - \kappa \mathbb{E}[X_{i,t} \mid \mathcal{F}_t] \leq e^{-\kappa a \omega_t}.$$

On $\{J_{i,t} = 0\}$, $Z_i(t) = Z_i(t-1)$. Hence

$$\mathbb{E}[Z_i(t) \mid \mathcal{F}_t] \leq Z_i(t-1).$$

Thus $(Z_i(v))_{v=0}^T$ is a nonnegative supermartingale for the post-round histories, with $Z_i(0) = 1$.

Let $G(v) = \sum_{t=1}^v A_t \omega_t$. Monotonicity of the pending set gives the pathwise identity $G(T) = \max_{i \in [d]} G_i(T)$. Indeed, if a round is positively charged, a task pending at the last such round was pending at every earlier charged round. Its total charge is therefore $G(T)$; the reverse inequality follows from $J_{i,t} \leq A_t$. If no round is charged, both sides are zero.

Since $N_i(T) \leq q$ and $\mathbb{E}Z_i(T) \leq 1$,

$$\begin{aligned} \mathbb{E}e^{\kappa a G(T)} &= \mathbb{E} \max_{i \in [d]} e^{\kappa a G_i(T)} \\ &\leq \sum_{i=1}^d \mathbb{E}e^{\kappa a G_i(T)} \leq d e^q. \end{aligned}$$

Jensen's inequality yields $\kappa a \mathbb{E}G(T) \leq q + \log d$, which is (27). Before σ , all $J_{i,t}$ vanish, so $G_i(\sigma-1) = N_i(\sigma-1) = 0$. The same tower, supermartingale, and conditional Jensen arguments, started at $\mathcal{F}_\sigma$, prove the conditional bounds. $\square$

6.2 One assignment: delivery and play

At a nonreserved assignment time σ , the coordinator chooses $U = \{a_1 < \dots < a_d\}$ and assigns a_i to player i . Write $F = [K] \setminus U$. Choose the auxiliary arm $e \in F$ to minimize $\bar{w}_e$, breaking ties by arm label. The coordinator retains its score register z_{a_i} for each assigned arm and sends the fixed DOWN records

$$M_i^{\text{DOWN}} = (\sigma, a_i, e, -\log_2 \xi_{a_i}). \quad (28)$$

A follower accumulates only its score increments after σ , starting from zero; it does not need the initial score.

During an assignment, $\mathcal{P} \subseteq [d]$ contains the followers whose receipts are missing. A *receipt* for i is the event

$$\{\tau_t = \text{DOWN}, k_t = i, H_t(M_i^{\text{DOWN}}) = 0, Y_{0,t} > 0\}.$$

For the action state $\text{state}_i \in \{0, 1, 2\}$ specified in Section 7.2, call follower i *unoccupied* if $\text{state}_i = 0$, *pending* if $\text{state}_i = 1$, and *released* if $\text{state}_i = 2$. It is *occupied* exactly when $\text{state}_i \in \{1, 2\}$. These terms concern the current live assignment; bootstrap row knowledge is defined separately in Section 4.

Follower i stores its last decoded start σ_i , increment sum S_i , and last increment δ_i , initially zero. Their update rules are in Section 7.2. On DATA, the row supplies independent uniform pilot arms $P_0, \dots, P_d \in [K]$. On DOWN or UP selecting i , define the receiver arm

$$R = \begin{cases} R_i, & \text{on DOWN,} \\ R_0, & \text{on UP.} \end{cases}$$

Algorithm 5 Choose the arms on one active nonreserved round

```

1: On DOWN or UP, let  $i$  be the selected follower and define  $R$  as above.
2: if the coordinator's assignment is live and  $\mathcal{P} = \emptyset$  then
3:   The coordinator draws and plays  $X_t \sim q_{w_t}^\gamma(\cdot | U)$  using its independent sampler coordinate.
4: else if the slot is DATA then
5:   The coordinator plays  $P_0$ .
6: else if DOWN selects  $i \in \mathcal{P}$  then
7:   The coordinator plays  $\text{SENDARM}(M_i^{\text{DOWN}}, R_0, R_i)$ .
8: else
9:   The coordinator plays  $R_0$ .
10: end if
11: for each follower  $j$  do
12:   if  $j$  is unoccupied then
13:     if the slot is DATA then
14:       Player  $j$  plays  $P_j$ .
15:     else if UP selects  $j$  then
16:       Player  $j$  plays  $\text{SENDARM}((\sigma_j, S_j, \delta_j), R_j, R_0)$ .
17:     else
18:       Player  $j$  plays  $R_j$ .
19:     end if
20:   else if  $j$  is released or the slot is DATA then
21:     Player  $j$  plays  $a_j$ .
22:   else if UP selects  $j$  and  $R_0 = e$  then
23:     Player  $j$  plays  $e$ .
24:   else if  $a_j = R$  then
25:     Player  $j$  plays  $e$ .
26:   else
27:     Player  $j$  plays  $a_j$ .
28:   end if
29: end for

```

The arm rule also applies during score collection. A pending follower i becomes released exactly on the event

$$\{Y_{i,t} > 0\} \cap (\{\tau_t = \text{UP}, k_t = i, R_0 = e\} \cup \{\tau_t = \text{DATA}, P_0 = a_i\}).$$

These are the UP and DATA certificates.

Lemma 10 (The auxiliary arm has bounded fast probability). *Suppose $\bar{w}_j \leq w_j \leq 2\bar{w}_j$ for every arm j , and let e minimize $\bar{w}_j$ over $j \in F = [K] \setminus U$. Then*

$$q_w^\gamma(e \mid U) \leq \frac{2}{3}.$$

Proof. Recall $s_U = \sum_{i \in U} \bar{w}_i^{-1}$ from (17). The function $x/(1 + s_U x)$ is increasing on $(0, \infty)$. For every $j \in F \setminus \{e\}$, minimality of $\bar{w}_e$ and the weight bounds give

$$\begin{aligned} \frac{w_j}{1 + s_U \bar{w}_j} &\geq \frac{\bar{w}_j}{1 + s_U \bar{w}_j} \geq \frac{\bar{w}_e}{1 + s_U \bar{w}_e}, \\ \frac{\bar{w}_e}{1 + s_U \bar{w}_e} &\geq \frac{1}{2} \frac{w_e}{1 + s_U \bar{w}_e}. \end{aligned}$$

Using the conditional fast-arm formula, we obtain

$$\begin{aligned} \frac{1}{q_w(e \mid U)} &= 1 + \sum_{j \in F \setminus \{e\}} \frac{w_j/(1 + s_U \bar{w}_j)}{w_e/(1 + s_U \bar{w}_e)} \\ &\geq 1 + \frac{|F| - 1}{2}. \end{aligned}$$

Thus $q_w(e \mid U) \leq 2/(|F| + 1)$. Since $|F| = K - m + 1 \geq 2$ and $0 \leq \gamma \leq 1$, the mixture satisfies

$$q_w^\gamma(e \mid U) \leq (1 - \gamma) \frac{2}{|F| + 1} + \frac{\gamma}{|F|} \leq \frac{2}{3}.$$

□

Lemma 11 (Deterministic avoidance and receipts). *Consider a reference execution during a live assignment. While $\mathcal{P} \neq \emptyset$, no follower is released. On DOWN selecting $i \in \mathcal{P}$, every pending follower j satisfies $A_{j,t} \neq R_i$. On such a round,*

$$\begin{aligned} Y_{i,t} &= H_t(M_i^{\text{DOWN}}) r_t(R_i) && \text{if } i \text{ is unoccupied,} \\ Y_{0,t} &= r_t(R_i) && \text{if } i \text{ is pending and } H_t(M_i^{\text{DOWN}}) = 0. \end{aligned}$$

In particular, a receipt for i implies that i was occupied at the start of the round.

Proof. At commitment no follower has decoded the new assignment, so none is released. Suppose $\mathcal{P} \neq \emptyset$ at the start of round t . The two possible release events imply, respectively,

$$\begin{aligned} \tau_t = \text{UP}, k_t = i, R_0 = e &\implies A_{0,t} = A_{i,t} = e, \\ \tau_t = \text{DATA}, P_0 = a_i &\implies A_{0,t} = A_{i,t} = a_i. \end{aligned}$$

Both give $Y_{i,t} = 0$. Thus neither event can change a pending follower into a released one. Induction over rounds proves the first assertion.

Now fix a DOWN round selecting $i \in \mathcal{P}$. Every pending follower j obeys

$$A_{j,t} = \begin{cases} a_j, & a_j \neq R_i, \\ e, & a_j = R_i. \end{cases}$$

In the second case $R_i = a_j \in U$ whereas $e \notin U$. Both cases therefore give $A_{j,t} \neq R_i$.

An unoccupied player i plays R_i on DOWN. If the hash bit is zero, the coordinator also plays R_i , making its reward zero. Therefore $Y_{i,t} = Y_{0,t} = 0$ in this case, and a positive receipt is impossible while i is unoccupied.

Suppose i is unoccupied and the hash bit is one. The coordinator plays $R_0 \neq R_i$. Each other unoccupied player j plays $R_j \neq R_i$ because the tuple is injective. Every pending player avoids R_i by the first paragraph. Hence player i is alone on R_i and observes $r_t(R_i)$.

If i is pending and the hash bit is zero, the coordinator plays R_i . Pending players avoid that arm. Every unoccupied player has a label different from i and uses its distinct tuple arm. Thus the coordinator is alone on R_i and observes $r_t(R_i)$.

□

Lemma 12 (Assignment and receipt cost). *Before a decoder failure, each unresolved DOWN message has positive-test probability at least*

$$\Pr(X_{i,t} = 1 \mid \mathcal{F}_t) \geq \frac{g_t}{8dK} \quad (29)$$

on every nonreserved round of its assignment. Once player i has decoded, its missing receipt has the same conditional lower bound. Let σ be a commitment time, and let G_{hand} be the support on nonreserved rounds from σ through the last receipt, stopped before the next cutoff or at the horizon, and including the round of a decoder failure if one occurs first. Conditional on the pre-round history $\mathcal{F}_\sigma$,

$$\mathbb{E}[G_{\text{hand}} \mid \mathcal{F}_\sigma] \leq 2K + \frac{8dK}{1 - e^{-1}}(2Q_{\text{dec}} + 2\log d + 3).$$

Proof. The slot is DOWN and selects i with probability $1/(4d)$. Conditional on $\mathcal{F}_t$ and the arm tuple, the fresh affine hash of the fixed message is a fair bit. Also R_i is uniform in $[K]$. If i is unoccupied, Lemma 11 gives a positive receiver observation whenever the bit is one and $r_t(R_i) > 0$. Therefore

$$\Pr(X_{i,t} = 1 \mid \mathcal{F}_t) = \frac{1}{4} \cdot \frac{1}{d} \cdot \frac{1}{2} \cdot \frac{g_t}{K} = \frac{g_t}{8dK}.$$

If i has decoded and its receipt is missing, the same lemma gives a positive coordinator observation whenever the bit is zero and $r_t(R_i) > 0$. Its receipt probability is again $g_t/(8dK)$. Every receipt is valid.

For the first phase, let P_t contain the followers whose DOWN word has not been accepted before round t , and set

$$X_{i,t} = \mathbf{1}\{\tau_t = \text{DOWN}, k_t = i, Y_{i,t} > 0\}.$$

Stop this phase when $P_t = \emptyset$, at an interruption, at a decoder failure, or at the horizon. Before a decoder failure, Lemma 8 gives quota $q = 2Q_{\text{dec}}$ for each fixed DOWN word, and (29) gives $a = 1/(8dK)$ with $\omega_t = g_t$. Lemma 9 therefore bounds its expected support by

$$K + \frac{8dK}{1 - e^{-1}}(2Q_{\text{dec}} + \log d + 1).$$

If all DOWN words are accepted before truncation, start the second phase on the next round. Now $P_t = \{i : i\text{'s receipt is missing before round } t\}$ and

$$X_{i,t} = \mathbf{1}\{\tau_t = \text{DOWN}, k_t = i, H_t(M_i^{\text{DOWN}}) = 0, Y_{0,t} > 0\}.$$

Every such event removes i from P_t , so $q = 1$; the receipt hazard in (29) again gives $a = 1/(8dK)$. Conditioning on this phase's start and applying Lemma 9 gives

$$K + \frac{8dK}{1 - e^{-1}}(\log d + 2).$$

Earlier receipts require no further service. The two phase bounds hold conditional on their respective start histories because the later slot, tuple, and hash coordinates are fresh. Applying the tower property at the second phase's start gives the displayed bound conditional on $\mathcal{F}_\sigma$. Truncation only decreases the nonnegative support sums. $\square$

Write $\chi_t(j) = \mathbf{1}\{r_t(j) > 0\}$.

Lemma 13 (Local release and its cost). *After the last assignment receipt, while player i remains pending and the stability bounds hold, its UP-certificate probability is at least*

$$\frac{\chi_t(e)}{12dK}. \quad (30)$$

Its DATA-certificate probability is $\chi_t(a_i)/(2K)$.

Use the coordinator's draw X_t as the logical fast arm. Let $\mathcal{T}$ consist of the nonreserved rounds after the last receipt and before the next interruption, truncated at the horizon or just after a charged failure. Then

$$\mathbb{E} \sum_{t \in \mathcal{T}} \left(r_t(U) + r_t(X_t) - \sum_{i=0}^d Y_{i,t} \right) \leq 3dK + \frac{12dK}{1 - e^{-1}} (\log d + 2). \quad (31)$$

Proof. By Lemma 11, every follower is occupied when the last receipt arrives, and no follower has yet been released. Thereafter the coordinator always plays $X_t \in F$.

Suppose UP selects a pending player i and $R_0 = e$. Player i plays e . Every other pending player j plays a_j , since $a_j \neq e$; released players also stay on their distinct assigned arms. Therefore, conditional on this slot and tuple event, i receives a positive reward exactly when $r_t(e) > 0$ and $X_t \neq e$. The independent fast sampler and Lemma 10 give

$$\begin{aligned} & \Pr(\tau_t = \text{UP}, k_t = i, R_0 = e, Y_{i,t} > 0 \mid \mathcal{F}_t) \\ &= \frac{1}{4dK} (1 - q_{w_t}^\gamma(e \mid U)) \chi_t(e) \geq \frac{\chi_t(e)}{12dK}. \end{aligned}$$

On DATA, every slow player stays on its assigned arm and the coordinator chooses from F . Thus player i observes $r_t(a_i)$. The independent event $P_0 = a_i$ has probability $1/K$, and DATA has probability $1/2$. Therefore

$$\Pr(\tau_t = \text{DATA}, P_0 = a_i, Y_{i,t} > 0 \mid \mathcal{F}_t) = \frac{\chi_t(a_i)}{2K}.$$

Define the predictable charge

$$\omega_t = q_{w_t}^\gamma(e \mid U) \chi_t(e) \in [0, 1].$$

On the interval $\mathcal{T}$, take $P_t = \{i : \text{state}_i = 1 \text{ before round } t\}$ and

$$A_t = \mathbf{1}\{t \in \mathcal{T}, P_t \neq \emptyset\}.$$

For this phase set

$$X_{i,t} = \mathbf{1}\{\tau_t = \text{UP}, k_t = i, R_0 = e, Y_{i,t} > 0\}.$$

The displayed event is an UP certificate, so it removes i from P_t by the next round; no follower reenters P_t before the interruption. Thus $q = 1$. On $\{J_{i,t} = 1\}$ the preceding hazard calculation gives

$$\mathbb{E}[X_{i,t} \mid \mathcal{F}_t] \geq \frac{\chi_t(e)}{12dK} \geq \frac{\omega_t}{12dK}.$$

Lemma 9 with $a = 1/(12dK)$ gives

$$\mathbb{E} \sum_{t \in \mathcal{T}} \mathbf{1}\{P_t \neq \emptyset\} \omega_t \leq K + \frac{12dK}{1 - e^{-1}} (\log d + 2). \quad (32)$$

The follower's local certificate determines its removal; the coordinator need not observe that removal.

For the reward comparison, each pending follower satisfies $A_{i,t} \in \{a_i, e\}$ and every released follower satisfies $A_{i,t} = a_i$. Since the a_i are distinct and $e \in F$, a released follower has $Y_{i,t} = r_t(a_i)$. A pending follower has $Y_{i,t} \geq 0$, so in either case

$$Y_{i,t} \geq r_t(a_i) - \chi_t(a_i) \mathbf{1}\{i \text{ is pending}\}.$$

The coordinator can collide with a follower only if $X_t = e$ and some follower is pending. Hence

$$Y_{0,t} \geq r_t(X_t) - r_t(e) \mathbf{1}\{X_t = e, \text{ some follower pending}\}.$$

Subtracting these inequalities from the logical reward gives

$$r_t(U) + r_t(X_t) - \sum_{i=0}^d Y_{i,t} \leq \sum_{i \text{ pending}} \chi_t(a_i) + r_t(e) \mathbf{1}\{X_t = e, \text{ some follower pending}\}. \quad (33)$$

Membership in $\mathcal{T}$ and the set P_t are determined before round t . For each $i \in [d]$, the DATA-certificate probability calculated above therefore gives

$$\begin{aligned} \frac{1}{2K} \mathbb{E} \sum_{t \in \mathcal{T}} \mathbf{1}\{i \in P_t\} \chi_t(a_i) &= \mathbb{E} \sum_{t \in \mathcal{T}} \mathbf{1}\{i \in P_t\} \mathbf{1}\{\tau_t = \text{DATA}, P_0 = a_i, Y_{i,t} > 0\} \\ &\leq 1. \end{aligned}$$

The last inequality is pathwise: that certificate changes state _{i} from 1 to 2, so it cannot be counted again before the interruption. Summing over i bounds the expected first term on the right of (33) by $2dK$.

For the second term, the fast-arm draw has conditional law $q_{w_t}^\gamma(\cdot \mid U)$, and $r_t(e) \leq \chi_t(e)$. Thus

$$\begin{aligned} \mathbb{E}[\mathbf{1}\{t \in \mathcal{T}, P_t \neq \emptyset\} r_t(e) \mathbf{1}\{X_t = e\} \mid \mathcal{F}_t] &= \mathbf{1}\{t \in \mathcal{T}, P_t \neq \emptyset\} r_t(e) q_{w_t}^\gamma(e \mid U) \\ &\leq \mathbf{1}\{t \in \mathcal{T}, P_t \neq \emptyset\} \omega_t. \end{aligned}$$

Taking expectations in (33), summing over $t \in \mathcal{T}$, and using (32) yields

$$\begin{aligned} \mathbb{E} \sum_{t \in \mathcal{T}} \left(r_t(U) + r_t(X_t) - \sum_{i=0}^d Y_{i,t} \right) &\leq 2dK + K + \frac{12dK}{1 - e^{-1}} (\log d + 2) \\ &\leq 3dK + \frac{12dK}{1 - e^{-1}} (\log d + 2), \end{aligned}$$

where the last step uses $d \geq 1$. □

6.3 Recording rewards and delayed local sums

During an assignment, follower i owns arm a_i and the coordinator owns F . During collection the coordinator owns all arms. Ownership specifies whose observation can supply an arm's increment.

On DATA the row independently selects the recorder source: the coordinator with probability $s_0 = 1/2$, or follower i with probability $s_i = 1/(2d)$. Write $V_t \in [0, 1]$ for the independent coordinate used to thin the

observation. For an owner i of arm j , let $\pi_i(j)$ be the conditional probability that it plays j on DATA. The other d players' pilots avoid j with probability

$$\zeta = (1 - 1/K)^d \geq \frac{1}{4}.$$

Define the recording mask for player i observing arm j by

$$\text{Mask}_{i,t}(j) := \mathbf{1}\{P_{\ell,t} \neq j \text{ for every } \ell \neq i\}. \quad (34)$$

The test uses every other player's public pilot coordinate, including dummy pilots of players whose DATA action is an assigned or fast arm. These d coordinates are independent of the owner's action, so the conditional probability that the mask equals one is always ζ . For the coordinator, score updates are frozen exactly when $\sigma > 0$ and $t \geq g$, using its registers after $\text{StartRound}(t)$; a new commitment resets $g > t$ and enables updates again.

Algorithm 6 Record a DATA increment or one candidate observation

Require: One player's local arm and observation; the output increment is initially zero.

```

1: On a non-DATA round, stop.
2: Let  $i$  be this player. If the public recorder source is not  $i$ , stop.
3: Let  $j$  and  $Y_{i,t}$  be its own arm and reward; read  $V_t$ .
4: if  $i$  is an unoccupied follower then
5:   if  $P_{\ell,t} \neq j$  for every  $\ell \neq i$  then
6:     Append  $(t, j, Y_{i,t}, V_t)$  to  $\mathcal{B}_i$ .
7:   end if
8:   Stop.
9: end if
10: if  $i = 0$ ,  $\sigma > 0$ , and  $t \geq g$  after  $\text{StartRound}(t)$  then
11:   Stop.
12: end if
13: if  $i$  does not own  $j$  or  $\text{Mask}_{i,t}(j) = 0$  then
14:   Stop.
15: end if
16: if  $V_t \leq \xi_j / ((1/2)s_i\pi_i(j)\zeta)$  then
17:   Set the output increment to  $\Delta_t(j) = \lfloor D\eta Y_{i,t} / \xi_j \rfloor$ .
18:   if  $i = 0$  then
19:     Outside collection, add  $\Delta_t(j)$  to  $z_j$ ; during collection, pass it to the cumulative update in
       Algorithm 7.
20:   else
21:     Add  $\Delta_t(j)$  to follower  $i$ 's local sum  $S_i$ .
22:   end if
23: end if
```

For analysis, let $I_t = j$ when arm j 's owner passes the DATA source, recording-mask, and thinning tests of Algorithm 6; otherwise let $I_t = \perp$. For an unoccupied follower, these tests use its saved observation and the true assignment, even if it has not decoded by the horizon. On an active nonreserved round with learning unfrozen, define, for $j \in [K]$,

$$\Delta_t(j) := \mathbf{1}\{I_t = j\} \left\lfloor \frac{D\eta r_t(j)}{\xi_j} \right\rfloor.$$

Set $\Delta_t = 0$ on every other round. At most one coordinate of Δ_t is nonzero.

Lemma 14 (Physical recording law). *In the reference product-stream execution, on every active nonreserved round with learning unfrozen, the arm I_t defined above satisfies (21). On $I_t = j$, the owner's observation is $r_t(j)$, so its increment is the one specified by (22).*

Proof. First, $d = m - 1 \leq K - 2$ and $\log(1 - x) \geq -x/(1 - x)$ for $0 \leq x < 1$ give

$$\zeta = (1 - 1/K)^d \geq \exp(-d/(K - 1)) \geq e^{-1} > \frac{1}{4}.$$

Fix arm j and its owner i conditional on $\mathcal{F}_t$. The owner's DATA action is independent of all other pilot coordinates. Thus the probability that DATA occurs, source i is selected, the owner plays j , and the mask holds is

$$\frac{1}{2}s_i\pi_i(j)\zeta.$$

On this event the owner is alone on j . An unoccupied nonowner uses its pilot, which the mask makes different from j . An occupied slow nonowner uses its distinct assigned arm. A pilot coordinator avoids j by the mask; a fast coordinator chooses from F , outside every slow assigned arm. During collection every physical player uses its pilot. This covers every DATA state.

An unoccupied follower has $\pi_i(j) = 1/K$ for its assigned arm, whereas an occupied follower has $\pi_i(j) = 1$. The coordinator has $\pi_0(j) = 1/K$ during collection and while assignment receipts are missing. After all receipts it has $\pi_0(j) = q_{w_t}^\gamma(j | U)$ for $j \in F$. Therefore

$$\frac{1}{2}s_i\pi_i(j)\zeta \geq \begin{cases} 1/(16mK), & \text{pilot coordinator or slow owner,} \\ q_{w_t}^\gamma(j | U)/16, & \text{fast owner.} \end{cases}$$

The first lower bound is at least ξ_j , since $\xi_j \leq 2/(AmK) \leq 1/(16mK)$. The second is at least ξ_j by (20). In particular, the thinning probability

$$\frac{\xi_j}{(1/2)s_i\pi_i(j)\zeta} \tag{35}$$

belongs to $[0, 1]$. Multiplication of the raw observation and thinning probabilities gives

$$\begin{aligned} \Pr(I_t = j | \mathcal{F}_t) &= \frac{1}{2}s_i\pi_i(j)\zeta \frac{\xi_j}{(1/2)s_i\pi_i(j)\zeta} \\ &= \xi_j. \end{aligned}$$

The remaining probability is $1 - \sum_j \xi_j$. The retained observation equals $r_t(j)$, so its increment is $\lfloor D\eta r_t(j)/\xi_j \rfloor$, as required. $\square$

For an unoccupied follower i assigned arm a_i , we have $s_i = 1/(2d)$ and $\pi_i(a_i) = 1/K$. Thus the raw DATA/source/action/mask event has probability $\zeta/(4dK)$, and its conditional thinning probability is

$$\frac{\xi_{a_i}}{\zeta/(4dK)} = \frac{4dK 2^{-h_i}}{\zeta}.$$

This gives the threshold in ψ_{h_i} of (41).

7 Scheduling and local implementation

For analysis, the *reference execution* uses the corrected initialization of Section 4 and rejects any decoded word other than the word currently encoded on that channel. Such a rejection does not stop score updates.

7.1 The cutoff and the collection state

For an assignment committed at round σ , define

$$g(\sigma) = \min\{T + 1, H(\lfloor \sigma/H \rfloor + 1)\}$$

as the next grid boundary, clipped at the end of the horizon. This clipping changes no action through round T and keeps the stored counter in $[T + 1]$. Recall that B_t is the proposal bit of Section 5 and equals zero on reserved rounds. A *public interruption* occurs at the start of round u if $u = g(\sigma)$ or $B_{u-1} = 1$. The assignment's *cutoff* is

$$u(\sigma) = \min(\{g(\sigma)\} \cup \{s + 1 : \sigma \leq s < T, B_s = 1\} \cup \{T + 1\}).$$

For the reference execution, define the calendar-round score by

$$Z_j(r) := \sum_{s < r} \Delta_s(j), \quad j \in [K].$$

Here Δ_s is the increment of Section 6.3; an increment computed after decoding is attributed to its observation round. Hence $Z_j(r + 1) - Z_j(r) = \Delta_r(j)$.

After $\text{StartRound}(t)$, the coordinator's score updates are frozen exactly when $\sigma > 0$ and $t \geq g$. Thus collection continues recording with the old rates before the old grid boundary, and records no increments from that boundary until a new commitment. At that commitment the anchor and rates are renewed and the new g exceeds t .

For a collection interval $[u, t)$, define the cumulative increment vector

$$c(t) = \sum_{r=u}^{t-1} \Delta_r.$$

Algorithm 7 Coordinator updates before and after a round

Require: The coordinator is active; initially $\sigma = 0$ and $g = T + 1$.

```
1: procedure STARTROUND( $t$ )
2:   if  $\sigma > 0$ ,  $u = T + 1$ , and either  $t = g$  or  $B_{t-1} = 1$  then
3:     Set  $u \leftarrow t$  and  $\mathcal{P} \leftarrow [d]$ .
4:     Set  $c \leftarrow 0$  and preserve the known entries of  $z(u)$  and  $\Delta_{u-1}$ .
5:     Set  $\kappa \leftarrow (1 - h)/h$  if  $u < g$ , and  $\kappa \leftarrow 0$  otherwise.
6:   end if
7:   if  $t$  is nonreserved and either  $\sigma = 0$  or  $(u \leq t$  and  $\mathcal{P} = \emptyset)$  then
8:     if  $\sigma = 0$  then
9:       Initialize every score to zero and every weight to one.
10:    else
11:      Recover  $z(u)$  and  $\Delta_{u-1}$ ; set  $z(t) \leftarrow z(u) + c$ .
12:    end if
13:    if  $\sigma = 0$  or  $t \geq g$  then
14:      Renew the anchor and rates by (19); draw  $U \sim p_{w_t}$  by Algorithm 9.
15:    else
16:      Recover  $w_{u-1}, w_u$  by Lemma 15; draw  $U \sim v_t$  by Algorithm 9.
17:    end if
18:    Write  $U = \{a_1 < \dots < a_d\}$  and choose  $e \in \arg \min_{j \notin U} \bar{w}_j$ , breaking ties by label.
19:    Retain  $z_{a_i}(t)$  as base scores and construct  $(t, a_i, e, -\log_2 \xi_{a_i})$  for each follower  $i$ .
20:    Set  $\sigma \leftarrow t$  and  $u \leftarrow T + 1$ .
21:    Set  $g \leftarrow g(t)$  and  $\mathcal{P} \leftarrow [d]$ .
22:  end if
23: end procedure
24: procedure FINISHROUND( $t, \Delta_t$ ) ▷ Called once after a nonreserved round
25:   if  $u \leq t$  then
26:     Set  $c \leftarrow c + \Delta_t$ .
27:     if  $t < g$  then
28:       Set  $\kappa \leftarrow (1 - h)\kappa$ .
29:     end if
30:   else
31:     Read the live assignment's proposal bit  $B_t$ .
32:   end if
33: end procedure
```

Lemma 15 (Recovery from accumulated increments). *Suppose an assignment committed at σ has cutoff u , and its collection ends with a commitment at t . Assume the coordinator retained $z(\sigma) = Z(\sigma)$ and, for every $i \in [d]$, obtains $S_i(u) = \sum_{s=\sigma}^{u-1} \Delta_s(a_i)$ and $\Delta_{u-1}(a_i)$. These values and the coordinator's increments determine*

$$Z(t) = Z(u) + c(t), \quad Z(u-1) = Z(u) - \Delta_{u-1}. \quad (36)$$

At this commitment t , the coordinator's reconstructed register vector satisfies $z(t) = Z(t)$. Consequently the coordinator can compute w_{u-1}, w_u, w_t without retaining the intermediate collection updates.

Proof. Ownership changes to the coordinator at u . It knows the old recording rates and every DATA observation used for an increment from u onward. It can therefore compute Δ_r even if some old slow-arm

scores $Z_j(u)$ are missing. Starting from $c(u) = 0$, induction over the collection rounds gives

$$c(t) = \sum_{r=u}^{t-1} \Delta_r = Z(t) - Z(u).$$

For each slow arm a_i , the returned S_i sums exactly its increments on $[\sigma, u)$. Adding the retained initial score $z_{a_i}(\sigma) = Z_{a_i}(\sigma)$ gives $Z_{a_i}(u)$. The coordinator knows the other cutoff scores from its own observations, so its reconstructed snapshot satisfies $z(u) = Z(u)$.

The supplied data give $\Delta_{u-1}(a_i)$ for every slow arm; the coordinator knows the remaining coordinates from its own observations. Subtracting this vector from $Z(u)$ gives $Z(u-1)$, while $z(t) = z(u) + c(t) = Z(t)$ by Algorithm 7. Thus (36) holds. Finally, $w_{r,j} = (1 + 1/D)^{Z_j(r)}$ determines the three weight vectors. No other collection score vector is needed. $\square$

7.2 Local state, message words, and acceptance

For an assignment starting at σ , the coordinator retains $(\sigma, (a_i)_{i \in [d]}, e, (z_{a_i}(\sigma))_{i \in [d]})$. Its structured DOWN record is $M_i^{\text{DOWN}} = (\sigma, a_i, e, -\log_2 \xi_{a_i})$ from (28).

Follower i retains the last accepted DOWN record $(\sigma_i, a_i, e_i, h_i)$, its increment sum S_i , its last increment δ_i , and an action state with values

$$\text{state}_i = \begin{cases} 0, & \text{unoccupied,} \\ 1, & \text{pending,} \\ 2, & \text{released.} \end{cases}$$

Initially $(\sigma_i, S_i, \delta_i, \text{state}_i) = (0, 0, 0, 0)$; the other DOWN fields are unused until the first acceptance. At an interruption, states 1 and 2 become 0, while $(\sigma_i, S_i, \delta_i)$ is retained.

On an active nonreserved round, the structured UP record offered by follower i at the arm-choice step is

$$M_{i,t}^{\text{UP}} = \begin{cases} (\sigma_i, S_i, \delta_i), & \tau_t = \text{UP}, k_t = i, \text{state}_i = 0, \\ \emptyset, & \text{otherwise.} \end{cases} \quad (37)$$

Before any DOWN acceptance, the UP record in (37), when offered, is $(0, 0, 0)$. During a known live assignment, $\text{state}_i \in \{1, 2\}$, so the follower encodes no UP record. While it is unoccupied, its triple remains fixed until it accepts a newer DOWN record.

For an assignment (σ, a_i) with cutoff u , define its *ENDPOINT* record by

$$M_i^{\text{END}} = \left(\sigma, \sum_{s=\sigma}^{u-1} \Delta_s(a_i), \Delta_{u-1}(a_i) \right), \quad (38)$$

Once the follower has formed this record, it offers it on its UP channel while $\text{state}_i = 0$, according to (37).

Set $L = \lceil \log_2(TD + 1) \rceil$ and $B = 4L$. Let $\text{bin}_L : \{0, \dots, 2^L - 1\} \rightarrow \mathbb{F}_2^L$ be the binary decomposition. Writing $\parallel$ for concatenation, encode the two record types by

$$\begin{aligned} \text{Enc}_{\text{DOWN}}(\sigma, a, e, h) &= \text{bin}_L(\sigma) \parallel \text{bin}_L(a) \parallel \text{bin}_L(e) \parallel \text{bin}_L(h), \\ \text{Enc}_{\text{UP}}(\sigma, S, \delta) &= \text{bin}_L(\sigma) \parallel \text{bin}_L(S) \parallel \text{bin}_L(\delta) \parallel 0^L. \end{aligned} \quad (39)$$

Here 0^L is the all-zero L -bit block.

Define the sets of syntactically valid records by

$$\mathcal{M}_{\text{DOWN}} = \{(\sigma, a, e, h) : \sigma \in [T], a, e \in [K], a \neq e, h \in \{0, \dots, 4b_N\}, \beta \leq 2^{-h} \leq \frac{2}{AmK}\},$$

and

$$\mathcal{M}_{\text{UP}} = \{(0, 0, 0)\} \cup ([T] \times \{0, \dots, TD\} \times \{0, \dots, D\}).$$

For a decoded bit word $w = w_1 \| w_2 \| w_3 \| w_4$ with $w_r \in \mathbb{F}_2^L$, put $x_r = \text{bin}_L^{-1}(w_r)$. The slot determines which parser is applied:

$$\begin{aligned} \text{Parse}_{\text{DOWN}}(w) &= \begin{cases} (x_1, x_2, x_3, x_4), & (x_1, x_2, x_3, x_4) \in \mathcal{M}_{\text{DOWN}}, \\ \perp, & \text{otherwise,} \end{cases} \\ \text{Parse}_{\text{UP}}(w) &= \begin{cases} (x_1, x_2, x_3), & w_4 = 0^L, (x_1, x_2, x_3) \in \mathcal{M}_{\text{UP}}, \\ \perp, & \text{otherwise.} \end{cases} \end{aligned}$$

Set $\text{Parse}_{\text{DOWN}}(\perp) = \text{Parse}_{\text{UP}}(\perp) = \perp$. For a parsed record M , write $M.\sigma$ for its first field, and $M.S, M.\Delta$ for the second and third fields of an UP record.

Lemma 16 (Message field bounds). *On every locally valid prefix of Algorithms 7 and 8, the outgoing DOWN records belong to $\mathcal{M}_{\text{DOWN}}$, and the offered UP records belong to $\mathcal{M}_{\text{UP}}$. Every field lies in $[0, TD]$, so (39) is valid.*

Proof. An assignment start lies in $[T]$, and its distinct arms a_i, e lie in $[K] \subseteq [0, D]$. By (19) and (20), $\xi_{a_i} = 2^{-h_i}$ is dyadic and satisfies $\beta \leq \xi_{a_i} \leq 2/(AmK)$. The construction-branch tuning of Section 8 gives $\eta \geq N^{-4}$ and therefore $\xi_{a_i} \geq N^{-4}$. Since $h_i = -\log_2 \xi_{a_i}$ is an integer, $0 \leq h_i \leq 4 \log_2 N \leq 4b_N \leq D$. Thus $M_i^{\text{DOWN}} \in \mathcal{M}_{\text{DOWN}}$.

Initially the UP record is $(0, 0, 0)$. Subsequently $\sigma_i \in [T]$. For $y \in [0, 1]$ and an admissible exponent h , (41) and $\beta \leq 2^{-h}$ give

$$0 \leq \psi_h(y, v) \leq D\eta 2^h y \leq D\eta/\beta \leq D.$$

The recorder also returns increments in $[0, D]$. The sum S_i contains at most one such integer per round by (43), and δ_i is one increment; hence $0 \leq S_i \leq TD$, $0 \leq \delta_i \leq D$. Thus each offered UP record lies in $\mathcal{M}_{\text{UP}}$. As $T \geq 1$, all four fields of either record lie in $[0, TD]$. $\square$

At a full-batch decoder decision after round t , let M be the parsed DOWN or UP record, or $\perp$. The local acceptance tests are

$$\begin{aligned} \text{Acc}_{i,t}^{\text{DOWN}}(M) &= 1 \iff M \in \mathcal{M}_{\text{DOWN}}, \quad \sigma_i < M.\sigma \leq t, \\ \text{Acc}_{i,t}^{\text{UP}}(M) &= 1 \iff M \in \mathcal{M}_{\text{UP}}, \quad u \leq t, \quad i \in \mathcal{P}, \quad M.\sigma = \sigma. \end{aligned} \tag{40}$$

Parsing checks the word's format; acceptance checks its relation to the receiver's state. Neither test alone establishes that the decoded record was actually sent.

7.3 Delayed increments from a sparse observation list

In Algorithm 6, each follower maintains a log to record its observation. For each follower $i \in [d]$, let $\mathcal{B}_i$ be its initially empty, time-ordered list of candidate observation records. On a round s that i 's declared public row marks nonreserved and DATA, it appends $(s, P_{i,s}, Y_{i,s}, V_s)$ if and only if

$$\begin{aligned} i \text{ is active and unoccupied,} \quad & \text{source} = i, \\ P_{\ell,s} &\neq P_{i,s} \quad (\forall \ell \neq i). \end{aligned}$$

The slot, source, pilot arms, and V_s are read from that row; $Y_{i,s}$ is the follower's quantized observation. The append rule does not test $Y_{i,s} > 0$ or apply thinning. It also runs before learning launch and during frozen collection whenever the displayed conditions hold.

In the product experiment V_s is uniform in $[0, 1]$; the finite implementation uses the corresponding rational coordinate of Section 9.1. On accepting a DOWN word, the follower filters the current list by (43) and then empties it, as specified in Algorithm 8.

Recall $\zeta = (1 - 1/K)^d$ from Section 6.3. For a rate exponent h , define the contribution of one saved observation by

$$\psi_h(y, v) = \mathbf{1}\left\{v \leq \frac{4dK 2^{-h}}{\zeta}\right\} \lfloor D\eta 2^h y \rfloor. \quad (41)$$

This is the recorder's thinning and increment rule for a pilot owner. In particular, a record with reward zero contributes zero.

After accepting $M = (\sigma, a, e, h)$ on round t , the follower computes the exclusive endpoint of the observations it must process:

$$\ell = \min(\{t + 1\} \cup \{g(\sigma) : g(\sigma) \leq t\} \cup \{s + 1 : \sigma \leq s < t, B_s = 1\}). \quad (42)$$

It uses the clipped boundary $g(\sigma)$ and the public row's past proposal coordinates. To evaluate the minimum, the follower scans B_s , $\sigma \leq s < t$, stopping at the first $B_s = 1$ or $g(\sigma)$; reserved rounds contribute $B_s = 0$ without reading a coordinate. It has not read these bits before acceptance, and if $\ell \leq t$, proposal coordinates from collection rounds in $[\ell, t)$ remain unread. The proposal B_t acts only at the start of round $t + 1$. Thus $\ell = t + 1$ means that the decoded assignment is still live through round t ; $\ell \leq t$ is its first cutoff. Its recovered sum and last increment are

$$\begin{aligned} S_i &= \sum_{\substack{(s,j,y,v) \in \mathcal{B}_i \\ \sigma \leq s < \ell, j=a}} \psi_h(y, v), \\ \delta_i &= \sum_{\substack{(s,j,y,v) \in \mathcal{B}_i \\ s=\ell-1, j=a}} \psi_h(y, v). \end{aligned} \quad (43)$$

The second sum contains at most one record. An empty sum is zero. Set $\text{state}_i = 1$ if $\ell = t + 1$, and set $\text{state}_i = 0$ otherwise.

Lemma 17 (Exact recovery from sparse records). *On a reference execution, the values in (43) equal $\sum_{s=\sigma}^{\ell-1} \Delta_s(a)$ and $\Delta_{\ell-1}(a)$.*

Proof. Fix a DOWN acceptance after round t , with decoded word (σ, a, e, h) , and let u be this assignment's first cutoff. Equation (42) gives $\ell = \min\{u, t + 1\}$. For $s \in [\sigma, \ell)$, player i is active, unoccupied, and owns a . Define the round event

$$E_s = \{s \text{ is nonreserved DATA, source} = i, P_{i,s} = a, P_{j,s} \neq a \text{ (} j \neq i \text{)}\}.$$

The mask and source tests in Algorithm 6 imply $\Delta_s(a) = 0$ on E_s^c . On E_s , the raw observation probability and thinning threshold are, respectively,

$$\frac{1}{2} \frac{1}{2d} \frac{1}{K} \zeta = \frac{\zeta}{4dK}, \quad \frac{\xi_a}{\zeta/(4dK)} = \frac{4dK 2^{-h}}{\zeta}.$$

Thus the recorder's integer update is the pathwise identity

$$\Delta_s(a) = \mathbf{1}\{E_s\} \psi_h(Y_{i,s}, V_s) \quad (\sigma \leq s < \ell). \quad (44)$$

The logging test is exactly E_s , and no record in $[\sigma, \ell)$ is erased before this DOWN acceptance: a newer assignment cannot start before the current ENDPOINT is accepted. There is at most one record at each time. Hence

$$\sum_{\substack{(s,j,y,v) \in \mathcal{B}_i \\ j=a}} \mathbf{1}\{\sigma \leq s < \ell\} \psi_h(y, v) = \sum_{s=\sigma}^{\ell-1} \Delta_s(a).$$

Restricting the same equality to $s = \ell - 1$ proves the assertion for δ_i . □

7.4 One local event rule

Algorithm 8 One round of the local protocol at player i

```

1: if  $i = 0$ , the coordinator is active, and its local state is not  $\dagger$  then
2:   Apply StartRound( $t$ ) of Algorithm 7.
3: else if  $i > 0$ , its state is not  $\dagger$ ,  $\text{state}_i \in \{1, 2\}$ , and the assignment is interrupted at  $t$  then
4:   Set  $\text{state}_i \leftarrow 0$  and keep  $S_i, \delta_i$  fixed.
5: end if
6: if the player is inactive or its declared row reserves  $t$  then
7:   Play its bootstrap-array arm and observe the reward.
8:   If  $i > 0$ , its state is not  $\dagger$ , and  $\text{state}_i \in \{1, 2\}$ , set  $\delta_i \leftarrow 0$ .
9:   Apply any complete-block bootstrap update and activation; return.
10: end if
11: if the local state is  $\dagger$  then
12:   Play arm  $i + 1$ ; apply any complete-block bootstrap update; return.
13: end if
14: Use Algorithm 5 to choose an arm; observe and quantize its reward.
15: Apply Algorithm 6 once.
16: If  $i > 0$  was known and live at the arm-choice step, set  $\delta_i$  to its returned increment, or zero if none was returned.
17: if DOWN selects  $i > 0$  and  $i$  was unoccupied at the arm-choice step then
18:   Feed its observation to DecodeStep in Algorithm 4; parse its output as DOWN to obtain  $M$ .
19:   if  $\text{Acc}_{i,t}^{\text{DOWN}}(M) = 1$  then
20:     Store  $(\sigma_i, a_i, e_i, h_i) \leftarrow M$ .
21:     Compute  $\ell$  by (42) and  $(S_i, \delta_i)$  by (43).
22:     Set  $\text{state}_i \leftarrow \mathbf{1}\{\ell = t + 1\}$ ; discard  $\ell$  and empty  $\mathcal{B}_i$ .
23:   end if
24: else if  $i = 0$ ,  $u \leq t$ , and UP selects  $j \in \mathcal{P}$  then
25:   Feed its observation to  $j$ 's DecodeStep; parse its output as UP to obtain  $M$ .
26:   if  $\text{Acc}_{j,t}^{\text{UP}}(M) = 1$  then
27:     Store  $z_{a_j}(u) \leftarrow z_{a_j}(\sigma) + M.S$  and  $\Delta_{u-1}(a_j) \leftarrow M.\Delta$ .
28:     Remove  $j$  from  $\mathcal{P}$ .
29:   end if
30: end if
31: if  $i = 0$ ,  $t < u$ ,  $\mathcal{P}$  was nonempty at the arm-choice step, and DOWN selected  $j$  then
32:   If  $H_t(M_j^{\text{DOWN}}) = 0$  and its reward is positive, remove  $j$  from  $\mathcal{P}$ .
33: else if  $i > 0$  was pending at the arm-choice step then
34:   Set  $\text{state}_i \leftarrow 2$  if UP selected  $i$ ,  $R_0 = e_i$ , and its reward is positive, or if DATA has  $P_0 = a_i$  and its reward is positive.
35: end if
36: if  $i = 0$  then
37:   Apply FinishRound( $t, \Delta_t$ ) of Algorithm 7 to its returned increment.
38: else if  $\text{state}_i \in \{1, 2\}$  then
39:   Read  $B_t$  after the update.
40: end if
41: Apply any complete-block bootstrap update and activation.

```

Each player has an absorbing local error state $\dagger$. In this state it performs no learning or transport updates, plays arm $i + 1$ on active nonreserved rounds, and follows its bootstrap-array rule on declared reserved rounds. A player enters $\dagger$ if a local operation has an input or computed output outside its prescribed domain. Appendix A defines the checked operations precisely.

Lemma 18 (Local state and sum invariants). *Consider the reference execution immediately after round t . If follower i has not applied a DOWN word, then $\text{state}_i = \sigma_i = S_i = \delta_i = 0$. Otherwise let $\sigma = \sigma_i$ be its last applied assignment and let u be that assignment's first cutoff, with $u = T + 1$ if no cutoff occurs in the horizon. Then*

$$\begin{aligned} S_i &= \sum_{s=\sigma}^{\min\{u, t+1\}-1} \Delta_s(a_i), \\ \delta_i &= \Delta_{\min\{u, t+1\}-1}(a_i), \\ \text{state}_i = 0 &\iff u \leq t. \end{aligned}$$

When $t < u$, the state is 2 if a release certificate has occurred since this DOWN acceptance, and is 1 otherwise.

Proof. Put $L_t = \min\{u, t + 1\}$. We induct over rounds, restarting at each DOWN acceptance. Before the first acceptance, Algorithm 8 gives $(\text{state}_i, \sigma_i, S_i, \delta_i) = (0, 0, 0, 0)$.

At a DOWN acceptance after round t , the cutoff scan gives $\ell = L_t$, and Lemma 17 gives

$$S_i = \sum_{s=\sigma}^{L_t-1} \Delta_s(a_i), \quad \delta_i = \Delta_{L_t-1}(a_i).$$

The follower was unoccupied at the arm-choice step, so no release test occurs on this round. The algorithm sets $\text{state}_i = \mathbf{1}\{L_t = t + 1\}$, proving the state assertion at acceptance.

Suppose the claim holds after round $t - 1$ and no DOWN word is accepted on round t . If $t < u$, then $L_t = t + 1$. On a nonreserved round the recorder supplies $\Delta_t(a_i)$; on a reserved round both this increment and the reset value of δ_i are zero. Hence

$$S_i(t) = S_i(t - 1) + \Delta_t(a_i), \quad \delta_i(t) = \Delta_t(a_i).$$

The state stays in $\{1, 2\}$ and changes from 1 to 2 precisely upon a release certificate. If $t = u$, the start-of-round interruption sets the state to 0 and preserves S_i, δ_i ; if $t > u$, these registers remain unchanged. In both cases $L_t = u$, so the formulas hold. This completes the induction. $\square$

Lemma 19 (Assignment identities and fixed replies). *In the reference execution, every follower applies each assignment at most once. Each assignment has one cutoff and one ENDPOINT per follower. After its cutoff is known, the derived UP word stays fixed until the next DOWN word is applied. During a known live assignment no UP message is encoded. No later assignment is committed before all current ENDPOINTS have been accepted. When a known live follower becomes unoccupied, its DOWN batch is empty. At each assignment's first cutoff, every coordinator UP batch is empty. An interruption that leaves a follower unoccupied preserves its existing partial DOWN batch.*

Proof. For successive DOWN acceptances by player i , the predicate (40) requires $\sigma_i^{\text{new}} > \sigma_i^{\text{old}}$. Hence an assignment start is applied at most once.

Fix an assignment (σ, a_i) with first cutoff u . If its DOWN word is accepted before u , Lemma 18 at the end of round $u - 1$ gives

$$(\sigma_i, S_i, \delta_i) = \left(\sigma, \sum_{s=\sigma}^{u-1} \Delta_s(a_i), \Delta_{u-1}(a_i) \right) = M_i^{\text{END}}.$$

If acceptance occurs after round $t \geq u$, then $\ell = u$ in (42), and Lemma 17 gives the same triple. In either case the state becomes 0 at the cutoff or acceptance. While it is 0, Algorithm 8 does not change S_i or δ_i except at a newer DOWN acceptance. Thus the ENDPOINT is unique and fixed. When the state is 1 or 2, Algorithm 5 does not encode an UP word.

At the cutoff, $\mathcal{P} = [d]$. An accepted UP word removes i only if $i \in \mathcal{P}$ and its start field equals σ . In the reference execution that word is M_i^{END} . Algorithm 7 commits a successor only after $\mathcal{P} = \emptyset$, so every current ENDPOINT is accepted before the next assignment.

Let W_i^{D} and W_i^{U} denote the DOWN and UP equation batches. Initially both are empty. At a DOWN acceptance after round v , the full-batch decision gives $W_i^{\text{D}}(v) = \emptyset$. For $v < s < u$, the known live follower's DOWN decoder is disabled, so $W_i^{\text{D}}(s) = W_i^{\text{D}}(v) = \emptyset$. Thus its batch is empty at the first cutoff. If the cutoff precedes DOWN acceptance, the interruption performs no decoder step and leaves the existing partial W_i^{D} unchanged.

If ENDPOINT i is accepted after round v , then $W_i^{\text{U}}(v) = \emptyset$ and $i \notin \mathcal{P}$. The UP decoder is disabled while $i \notin \mathcal{P}$ and throughout the next live assignment. Hence W_i^{U} remains empty from v through that assignment's first cutoff. Every i is removed before the next commitment; induction over assignments proves the UP assertion. $\square$

7.5 Integrating the unobserved refreshes

Let t_0 be the first commitment. Algorithm 7 initializes the coordinator's scores, so $z(t_0) = 0 = Z(t_0)$ because all earlier increments vanish. Suppose $z(\sigma) = Z(\sigma)$ at a commitment σ . Lemma 19 supplies the correct ENDPOINTS, and Lemma 15 yields $z(t) = Z(t)$ at the next commitment t . Hence, by induction, the coordinator's retained register satisfies $z_{a_i}(t) = Z_{a_i}(t)$ for each newly selected arm a_i at every commitment t .

Lemma 20 (The assignment law after collection). *Let $(\widehat{U}_r, \widehat{z}(r))$ be the slow set and score vector of Algorithm 3. Let $Z(r)$ be the calendar-round score defined above, and let $U^{\text{phys}}(r)$ be its assigned slow set on a live round. These executions admit a coupling that preserves the product-uniform law of the reference sampler coordinates and, almost surely, satisfies*

$$\begin{aligned} Z(r) &= \widehat{z}(r) \quad \text{from launch, before the first stability failure,} \\ U^{\text{phys}}(r) &= \widehat{U}_r \quad \text{on every live round before that failure.} \end{aligned}$$

Suppose the reference coordinator commits at round t after an assignment starting at σ with cutoff u , before stability failure. Let $\mathcal{G}_t^-$ be generated by the physical states, actions, observations, and coordinates read before the commitment sampler at t , together with $(\Delta_s)_{s < t}$; unread coordinates are excluded. Write $p_r = p_{w_r}$. Under the coupling, for every $U \in \binom{[K]}{d}$,

$$\Pr(\widehat{U}_t = U \mid \mathcal{G}_t^-) =: \pi_t(U) = \begin{cases} \gamma_t(U) := p_t(U) + \kappa_t(p_u(U) - p_{u-1}(U)), & t < g(\sigma), \\ p_t(U), & t \geq g(\sigma), \end{cases} \quad (45)$$

where κ_t is the coefficient of Algorithm 7.

Proof. We construct the coupling inductively over physical commitments.

Scores and live assignments. Initially both executions have zero scores. Suppose that at a physical commitment the two score vectors agree and the physical slow set is coupled to the logical slow set.

On every subsequent round on which recording is enabled, Lemma 14 gives the physical recorded arm and increment the same conditional law as the estimate step of Algorithm 3. We therefore use the same (I_r, Δ_r)

in both executions. On rounds on which recording is disabled, set $\Delta_r = 0$ in both. Hence, since both score vectors receive the same increments,

$$Z(r) = \widehat{z}(r)$$

until the first stability failure.

While the assignment is live, use the same proposal bit in the two executions. When the physical coordinator draws a fast arm, use that same draw for the logical learner; otherwise draw the logical fast arm independently from its prescribed conditional law. Thus the two slow sets agree throughout the live assignment, up to their common cutoff u .

The logical slow-set law during collection. Fix an assignment starting at σ , with cutoff u , whose next physical commitment is at t . During collection, the physical protocol does not use the logical slow set or the auxiliary proposal/refresh coordinates of Algorithm 3. Moreover, the physical proposal coordinates of collection rounds are not read. Hence the physical collection path, the commitment time t , and $\mathcal{G}_t^-$ may be generated first, and the independent logical refresh coordinates may be generated afterwards without changing their joint law.

Let $\mathcal{H}_t$ be $\mathcal{G}_t^-$ enlarged by all earlier physical-row coordinates, including those not read by the protocol. For $u \leq r \leq t$, write

$$\mu_r(U) := \Pr(\widehat{U}_r = U \mid \mathcal{H}_t).$$

First suppose that $t < g(\sigma)$. Then the cutoff at u was caused by $B_{u-1} = 1$, so $\mu_u = \rho_{u-1}$. On a nonreserved collection round r , Algorithm 3 gives

$$\mu_{r+1} = (1 - h)\mu_r + h\rho_r.$$

Using

$$h\rho_r = p_{r+1} - (1 - h)p_r$$

from (23), we obtain

$$\mu_{r+1} - p_{r+1} = (1 - h)(\mu_r - p_r).$$

On reserved rounds both sides remain unchanged. If

$$n_{u,t} := \#\{r \in [u, t) : r \text{ is nonreserved}\},$$

iteration therefore gives

$$\begin{aligned} \mu_t - p_t &= (1 - h)^{n_{u,t}}(\rho_{u-1} - p_u) \\ &= \frac{(1 - h)^{n_{u,t}+1}}{h}(p_u - p_{u-1}). \end{aligned}$$

Algorithm 7 initializes $\kappa = (1 - h)/h$ at the cutoff and multiplies it by $1 - h$ on each nonreserved collection round. Thus

$$\kappa_t = \frac{(1 - h)^{n_{u,t}+1}}{h},$$

and consequently

$$\mu_t(U) = p_t(U) + \kappa_t(p_u(U) - p_{u-1}(U)) = v_t(U).$$

If instead $t \geq g(\sigma)$, the old episode is frozen at $g(\sigma)$ and the commitment at t starts a new episode. Hence Algorithm 3 draws

$$\widehat{U}_t \sim p_t,$$

so $\mu_t = p_t$. In either case,

$$\Pr(\widehat{U}_t = U \mid \mathcal{H}_t) = \pi_t(U).$$

Since π_t is $\mathcal{G}_t^-$ -measurable, the tower property also gives

$$\Pr(\widehat{U}_t = U \mid \mathcal{G}_t^-) = \pi_t(U),$$

which proves (45).

Coupling the commitment sampler. It remains to make the physical commitment choose exactly $\widehat{U}_t$ without changing the product law of its sampler coordinates. Fix a realization h_t of the enlarged history $\mathcal{H}_t$. Let $F_{h_t}(v)$ be the commitment sampler's output when its $K + 1$ scalar coordinates are $v \in [0, 1)^{K+1}$, and let λ be product Lebesgue measure on that cube. By Lemma 24, the law of $F_{h_t}(v)$ is π_t : it is v_t within the episode and p_t at an initial or post-grid commitment. The logical slow set has the same conditional law even given $\mathcal{H}_t$.

Conditional on $\widehat{U}_t = U$ with $\pi_t(U) > 0$, sample the physical coordinate vector with distribution

$$\frac{\mathbf{1}\{F_{h_t}(v) = U\}\lambda(dv)}{\pi_t(U)}.$$

The two outputs agree. For every measurable set D_0 in the cube, averaging over the logical set gives

$$\Pr(v \in D_0 \mid \mathcal{H}_t = h_t) = \sum_U \lambda(D_0 \cap F_{h_t}^{-1}(U)) = \lambda(D_0).$$

Zero-probability sets U require no conditional choice. Hence the physical vector remains product-uniform and independent of all earlier physical-row coordinates, including unread ones. Subsequent coordinates are drawn freshly. The coupling may correlate this vector with the auxiliary logical randomness, which imposes no restriction on the physical protocol's marginal law.

Induction over commitments now gives $Z(r) = \widehat{z}(r)$ before the first stability failure and $U^{\text{phys}}(r) = \widehat{U}_r$ on every live round before that failure, completing the proof. $\square$

7.6 Score collection and total cost of interruptions

In the product-stream reference execution, fix the rounded reward matrix and reserved mask. Fix a transfer starting at a stopping time s : a cutoff starts score collection, while the first commitment starts the initial transfer. Let C contain nonreserved rounds from the cutoff through the round before the next commitment. Let $\mathcal{H}$ contain nonreserved rounds from that commitment through the last DOWN receipt, or through the round before the next cutoff if a receipt remains missing. Let $\mathcal{R}$ contain the subsequent nonreserved rounds before that cutoff. For the initial transfer $C = \emptyset$ and $\mathcal{H}$ begins at s ; any later phase is empty if its start does not occur. Include a round on which a failure is detected, and stop all phases thereafter or at the horizon. Here a failure is a decoder failure of Lemma 8 or the first weight-doubling update.

Use the capped support envelope

$$\bar{g}_t := \min\{m, g_t\}.$$

Both a comparator's reward and the reward of any logical m -set are at most $\bar{g}_t$. This nonnegative envelope permits comparison with the full auxiliary regret even when that regret is negative on selected communication rounds.

Define the phase bounds

$$\begin{aligned} B_{\text{col}} &:= 2K + \frac{16dK}{1 - e^{-1}}(2Q_{\text{dec}} + \log d + 1), \\ B_{\text{hand}} &:= 2K + \frac{8dK}{1 - e^{-1}}(2Q_{\text{dec}} + 2\log d + 3), \\ B_{\text{rel}} &:= 3dK + \frac{12dK}{1 - e^{-1}}(\log d + 2), \end{aligned}$$

and their sum

$$\begin{aligned} C_{\text{tr}} &:= B_{\text{col}} + B_{\text{hand}} + B_{\text{rel}} \\ &= O(dK[Q_{\text{dec}} + \log d + 1]) = O(mK\Lambda). \end{aligned} \quad (46)$$

Lemma 21 (Conditional support-envelope cost of one transfer). *Define the stopped transfer charge by*

$$J_s := \sum_{t \in C \cup \mathcal{H}} \bar{g}_t + \sum_{t \in \mathcal{R}} \left(r_t(U_t) + r_t(X_t) - \sum_{i=0}^d Y_{i,t} \right). \quad (47)$$

Then conditional on the pre-round history $\mathcal{F}_s$,

$$\mathbb{E}[J_s \mid \mathcal{F}_s] \leq C_{\text{tr}}.$$

Proof. By Lemma 19, an old DOWN message stays fixed while it remains undecoded. On DOWN selecting its unoccupied recipient i , every other slow player uses a distinct tuple arm. Hash bit one makes the coordinator use $R_0 \neq R_i$. While this DOWN decoder is pending, its conditional positive-test probability is

$$\Pr(\tau_t = \text{DOWN}, k_t = i, Y_{i,t} > 0 \mid \mathcal{F}_t) = \frac{1}{4} \cdot \frac{1}{d} \cdot \frac{1}{2} \cdot \frac{g_t}{K} = \frac{g_t}{8dK}.$$

This is the same lower bound as (29).

Once player i forms its UP score, the word stays fixed. On UP selecting i , the coordinator listens on R_0 . Every other slow player uses a distinct R_j , and hash bit one makes player i use $R_i \neq R_0$. While this UP decoder is pending, its conditional positive-test probability is likewise

$$\Pr(\tau_t = \text{UP}, k_t = i, Y_{0,t} > 0 \mid \mathcal{F}_t) = \frac{1}{4} \cdot \frac{1}{d} \cdot \frac{1}{2} \cdot \frac{g_t}{K} = \frac{g_t}{8dK}.$$

In the first phase, let P_t contain exactly the followers whose old DOWN word remains unaccepted before round t , and take $X_{i,t} = \mathbf{1}\{\tau_t = \text{DOWN}, k_t = i, Y_{i,t} > 0\}$. The phase ends when this set empties or collection is truncated. Each old DOWN word is fixed; Lemma 8 gives $q = 2Q_{\text{dec}}$, and the DOWN hazard above gives $a = 1/(8dK)$ for $\omega_t = g_t$.

If the first phase is not truncated, the second phase starts at u when all old DOWN words were accepted before u ; otherwise it starts on the round after their last acceptance. In that phase let P_t be the followers whose current ENDPOINT remains unaccepted, and set $X_{i,t} = \mathbf{1}\{\tau_t = \text{UP}, k_t = i, Y_{0,t} > 0\}$. Lemma 19 makes each owed ENDPOINT fixed; the same decoder quota and the UP hazard above apply. Lemma 9, conditional on the start of either phase, bounds that phase's expected support by

$$K + \frac{8dK}{1 - e^{-1}}(2Q_{\text{dec}} + \log d + 1).$$

If $C = \emptyset$ there is no charge. Otherwise the tower property at the second phase's start gives

$$\mathbb{E} \left[\sum_{t \in C} g_t \mid \mathcal{F}_s \right] \leq 2 \left[K + \frac{8dK}{1 - e^{-1}}(2Q_{\text{dec}} + \log d + 1) \right] = B_{\text{col}}.$$

DOWN and UP channels may both operate during either phase; the partition only assigns each round's g_t to one bound. Earlier accepted ENDPOINTS are omitted from the second P_t . Repeated interruptions do not change the fixed cutoff or either message, so they do not reset a batch.

If handover starts at a commitment σ_H , Lemma 12, applied conditional on $\mathcal{F}_{\sigma_H}$, gives

$$\mathbb{E} \left[\sum_{t \in \mathcal{H}} g_t \mid \mathcal{F}_{\sigma_H} \right] \leq B_{\text{hand}}.$$

If release starts at round σ_R , the DATA-certificate calculation and weighted service bound in the proof of Lemma 13 apply conditional on $\mathcal{F}_{\sigma_R}$. Thus (31) gives

$$\mathbb{E} \left[\sum_{t \in \mathcal{R}} \left(r_t(U_t) + r_t(X_t) - \sum_{i=0}^d Y_{i,t} \right) \middle| \mathcal{F}_{\sigma_R} \right] \leq B_{\text{rel}}.$$

For $t \in \mathcal{C} \cup \mathcal{H}$, $\bar{g}_t \leq g_t$, so the collection and handover support bounds apply to the new charge. Empty phases contribute zero. Applying the tower property at the starts of handover and release, and summing the three phase bounds, gives

$$\mathbb{E}[J_s \mid \mathcal{F}_s] \leq B_{\text{col}} + B_{\text{hand}} + B_{\text{rel}} = C_{\text{tr}}.$$

□

Proposition 22 (Total stopped transport envelope). *Consider the product-stream reference execution launched with zero scores and a fixed reserved mask. Let τ_{bad} be the first decoder failure or first weight-doubling update, with value $T + 1$ if neither occurs. Include its round in the phase charges and stop afterwards. Let $\mathcal{C}_*$, $\mathcal{H}_*$, $\mathcal{R}_*$ be the disjoint unions of these stopped collection, handover, and release phases. Define*

$$J_{\text{stop}} := \sum_{t \in \mathcal{C}_* \cup \mathcal{H}_*} \bar{g}_t + \sum_{t \in \mathcal{R}_*} \left(r_t(U_t) + r_t(X_t) - \sum_{i=0}^d Y_{i,t} \right).$$

With C_{tr} from (46), $\delta = N^{-8}$, and $A = 256$,

$$\mathbb{E}J_{\text{stop}} \leq C_{\text{tr}} [(4AmK + 32)\eta T + 2]. \quad (48)$$

Moreover, $\Pr(\tau_{\text{bad}} \leq T) \leq 2\delta$.

Proof. Number of transfers. For each round t , let L_t indicate that the live assignment's proposal coordinate is read before the stopping event, and let $P = \sum_{t=1}^T L_t B_t$. Immediately before that coordinate is read, L_t is determined and B_t is Bernoulli with parameter $h = 4AmK\eta$. Hence

$$\mathbb{E}[L_t B_t \mid \mathcal{F}_t^{\text{prop}}] = hL_t, \quad \mathbb{E}P = h \sum_{t=1}^T \mathbb{E}L_t \leq hT,$$

where $\mathcal{F}_t^{\text{prop}}$ is the history immediately before a possible proposal read. Let M count transfers started through τ_{bad} . Apart from the initial transfer, a transfer begins only after a read proposal or at a grid boundary; a grid boundary inside collection starts no additional transfer. Allowing one final incomplete transfer,

$$M \leq P + \lfloor T/H \rfloor + 2.$$

Since $H = \lfloor (16\eta)^{-1} \rfloor \geq (32\eta)^{-1}$,

$$\mathbb{E}M \leq (4AmK + 32)\eta T + 2. \quad (49)$$

Conditional envelope charges. For $k \in [T + 1]$, let σ_k be the start of the k -th counted transfer, or $T + 1$ if it does not occur. Set $J_k = J_{\sigma_k}$ for an existing transfer, using Lemma 21's stopped phases, and set $J_k = 0$ otherwise. Each summand is in $[-m, m]$, so these charges are integrable. That lemma gives

$$\mathbb{E}[J_k \mid \mathcal{F}_{\sigma_k}] \leq C_{\text{tr}} \mathbf{1}\{\sigma_k \leq T\}.$$

Since $M = \sum_k \mathbf{1}\{\sigma_k \leq T\}$, the tower property yields

$$\mathbb{E} \sum_k J_k \leq C_{\text{tr}} \mathbb{E}M. \quad (50)$$

Collection ends before the next commitment round; handover includes its last receipt round; release starts on the following round and ends before the next cutoff. Thus the phase sets of distinct transfers are disjoint and cover every nonreserved round from launch through $\min\{\tau_{\text{bad}}, T\}$. In particular, $\sum_k J_k = J_{\text{stop}}$, which proves (48).

Failure probability. The decoder bound applies before a weight doubling, and the stability bound controls the first weight doubling. Consequently,

$$\Pr(\tau_{\text{bad}} \leq T) \leq \frac{\delta}{N^4} + N^{-8} \leq 2\delta.$$

The failure round is included because its actions and rewards occur before the decoder decision or weight update detecting the failure. No learning-regret estimate has been conditioned on this event or truncated at a decoder failure; the full auxiliary sum is used in Section 8. $\square$

8 Proof of the main theorem

Proof of Theorem 1. Use the array constants chosen in the proof of Lemma 29 and the message quota (26). The transport proofs give an absolute constant in $C_{\text{tr}} = O(mK\Lambda)$. Choose C_1 large enough to dominate it and all learning coefficients, and then enlarge the branch constant C_b one final time if necessary.

Rational tuning. Let $\mathcal{J} = \lceil \log_2(\binom{K}{m} + 1) \rceil$ and $W = C_1 m^2 K^2 b_N$. Use the largest power of two η satisfying

$$\eta \leq (2AmKL_{\text{stab}})^{-1}, \quad \eta^2 WT \leq \mathcal{J}.$$

Here $L_{\text{stab}} = \lceil \log_2(2K(T+1)N^8) \rceil = O(\Lambda)$. The quantity $\sqrt{\mathcal{J}/(WT)}$ satisfies the first constraint whenever

$$T \geq \frac{4A^2 m^2 K^2 L_{\text{stab}}^2 \mathcal{J}}{W} = \frac{4A^2 L_{\text{stab}}^2 \mathcal{J}}{C_1 b_N}.$$

The right side is at most $C\mathcal{J}\Lambda$. Since $\binom{K}{m} \leq 2^K$, we have $\mathcal{J} \leq K+1$. On the construction branch, $k_\star \leq K$ gives

$$\begin{aligned} KT &\geq k_\star T > C_b \widehat{V} \geq C_b K^{5/2} \sqrt{T} \Lambda^2, \\ T &> C_b^2 K^3 \Lambda^4. \end{aligned}$$

Choosing C_b sufficiently large therefore ensures the rate cap. The largest-dyadic rule yields

$$\frac{1}{2} \sqrt{\frac{\mathcal{J}}{WT}} < \eta \leq \sqrt{\frac{\mathcal{J}}{WT}}. \quad (51)$$

Also $\mathcal{J} \asymp \log \binom{K}{m}$. The lower bound above gives $\eta \geq c/(mK\sqrt{T}\Lambda) \geq c/N^2$ for a universal $c > 0$. Increasing C_b ensures $N \geq c^{-1/2}$, and hence $\eta \geq N^{-4}$, as required by the finite-arithmetic bounds in Section 9.

Whole-row concentration. Let $\mathcal{G}_0$ be the event that Algorithm 10 publishes a complete table whose masks and arm arrays satisfy Lemma 29 and Lemma 33. Their failure bounds give $\Pr(\mathcal{G}_0) \geq 1 - CN^{-40}$. Fix masks and arrays in $\mathcal{G}_0$, and let $R_1, \dots, R_M$ be the independent finite control rows sampled by the constructor.

For a denominator- D reward matrix r and $j \in [M]$, let $F_{j,r}(R_j) \in [-mT, mT]$ be the conditional expected regret under reference execution, where the expectation is taken over private bootstrap randomness conditional on $J = j$ and on the realized row R_j . Put

$$\mathcal{E} := E_{\text{false}} \cup E_{\text{amb}} \cup E_{\text{stab}}, \quad E_{j,r}(R_j) := \Pr(\mathcal{E} \mid J = j, R_j) \in [0, 1],$$

where E_{false} and E_{amb} are the decoder events of Appendix A.2, and E_{stab} is the weight-stability failure of Lemma 5. The reference uses the corrected bootstrap state and rejects a decoded word other than the current encoded word. Its correction precedes every control access, so $F_{j,r}$ and $E_{j,r}$ depend on R_j alone as j varies.

Write

$$\mu_F(r) := \frac{1}{M} \sum_{j=1}^M \mathbb{E} F_{j,r}(R_j), \quad \mu_E(r) := \frac{1}{M} \sum_{j=1}^M \mathbb{E} E_{j,r}(R_j),$$

and

$$\bar{F}_r := \frac{1}{M} \sum_j F_{j,r}(R_j), \quad \bar{E}_r := \frac{1}{M} \sum_j E_{j,r}(R_j).$$

For fixed r , independence across j and Hoeffding's inequality [Hoe63] yield

$$\begin{aligned} \Pr \{ \bar{F}_r > \mu_F(r) + V \} &\leq e^{-MV^2/(2m^2T^2)}, \\ \Pr \left\{ \bar{E}_r > \mu_E(r) + \frac{V}{mT} \right\} &\leq e^{-2MV^2/(m^2T^2)}. \end{aligned}$$

There are $(D+1)^{KT}$ denominator- D matrices, and

$$\frac{MV^2}{2m^2T^2} \geq 2K^7T\Lambda^4 > KT \log(D+1) + 40\Lambda + \log 2.$$

Let G_{row} be the event that the following inequalities hold simultaneously

$$\bar{F}_r \leq \mu_F(r) + V, \tag{52}$$

$$\bar{E}_r \leq \mu_E(r) + \frac{V}{mT}. \tag{53}$$

Thus a union bound, conditional on $\mathcal{G}_0$, gives an event $\mathcal{G}_{\text{row}}$ with $\Pr(\mathcal{G}_{\text{row}}^c \mid \mathcal{G}_0) \leq CN^{-40}$. Hence $\mathcal{G}_{\text{pub}} := \mathcal{G}_0 \cap \mathcal{G}_{\text{row}}$ has probability at least $1 - CN^{-32}$.

Regret decomposition. Fix a public output in $\mathcal{G}_{\text{pub}}$ and a rounded reward matrix r . On $\mathcal{E}_{\text{boot}} \cap \mathcal{E}^c$, Proposition 3 couples initialization, the decoder rule couples subsequent messages, and Lemma 27 excludes a local domain error. If $\mathfrak{R}_T^{\text{act}}$ and $\mathfrak{R}_T^{\text{ref}}$ are the two complete realized regrets, then

$$|\mathfrak{R}_T^{\text{act}} - \mathfrak{R}_T^{\text{ref}}| \leq 2mT \mathbf{1}\{\mathcal{E}_{\text{boot}}^c \cup \mathcal{E}\}.$$

Taking expectations conditional on the published output gives

$$R_T(r) \leq \bar{F}_r + 2mT(\Delta_{\text{boot}} + \bar{E}_r). \tag{54}$$

Recall the two product-stream estimates needed for the reference. Write the auxiliary-learning bound and the stopped transport bound, with the failure-tail allowance established below, as

$$\begin{aligned} \mathcal{L}_\eta &:= \frac{2 \log \binom{K}{m}}{\eta} + \left(2AmK^2 + \frac{A}{4}(K-m+1)L_{\text{stab}} \right) \eta T + \frac{mT}{\eta D} + CmTN^{-8}, \\ \mathcal{T}_\eta &:= C_{\text{tr}}[(4AmK + 32)\eta T + 2] + 4mT\delta. \end{aligned}$$

Let $\tau_L \in [T+1]$ be the reference launch round, with $\tau_L = T+1$ if no launch occurs. For every comparator $S \in \mathcal{S}$, $r_t(S) \leq g_t$, so Proposition 3(ii) gives

$$\sum_{t < \tau_L} r_t(S) \leq G(\tau_L - 1) \leq CQ_{\text{boot}}.$$

Lemma 33(i) bounds the reserved-round charge by $m(2pT + C_0\Lambda)$.

Here is the pathwise accounting that connects the two estimates. Fix $S \in \mathcal{S}$ and put $c_t = r_t(S)$, $\ell_t = r_t(U_t) + r_t(j_t)$, and $y_t = \sum_i Y_{i,t}$ on enabled rounds. Let τ_{bad} and J_{stop} be as in Proposition 22. On an enabled collection or handover round through τ_{bad} ,

$$c_t - y_t = (c_t - \ell_t) + (\ell_t - y_t) \leq (c_t - \ell_t) + \bar{g}_t,$$

because $\ell_t \leq \bar{g}_t = \min\{m, g_t\}$ and $y_t \geq 0$. On a frozen collection round, the auxiliary learner has no summand and $c_t - y_t \leq \bar{g}_t$. On a release round, the coupling of Lemma 20 identifies $j_t = X_t$, so its remaining discrepancy $\ell_t - y_t$ is included in J_{stop} .

Retain the full auxiliary sum over $\mathcal{A}$ defined in Section 5. Its tail after τ_{bad} is at least $-mT\mathbf{1}\{\tau_{\text{bad}} \leq T\}$, and the actual-regret tail is at most $mT\mathbf{1}\{\tau_{\text{bad}} \leq T\}$. Consequently,

$$\begin{aligned} \mathfrak{R}_T^{\text{ref}}(S) &\leq CQ_{\text{boot}} + m(2pT + C_0\Lambda) + \sum_{t \in \mathcal{A}} (c_t - \ell_t) + J_{\text{stop}} \\ &\quad + 2mT\mathbf{1}\{\tau_{\text{bad}} \leq T\}. \end{aligned} \tag{55}$$

The failure round itself is included in the valid comparison since its rewards precede the failure-detecting update. Taking expectations uses the full auxiliary bound of Proposition 7, the stopped envelope bound, and $\Pr(\tau_{\text{bad}} \leq T) \leq 2\delta$. In particular, it does not condition the learner on successful decoding or stop its estimate filtration at a decoder failure.

Lemma 25 transfers both complete-reference regret and the event probability from ideal product rows to finite product rows with allowance ϵ_{samp} . The accounting above and the decoder and stability bounds give

$$\mu_F(r) \leq CQ_{\text{boot}} + m(2pT + C_0\Lambda) + \mathcal{L}_\eta + \mathcal{T}_\eta + 2mT\epsilon_{\text{samp}}, \tag{56}$$

$$\mu_E(r) \leq \delta/N^4 + N^{-8} + \epsilon_{\text{samp}}. \tag{57}$$

Combining (52)–(57) with (54) gives

$$\begin{aligned} R_T(r) &\leq CQ_{\text{boot}} + m(2pT + C_0\Lambda) + \mathcal{L}_\eta + \mathcal{T}_\eta + CV \\ &\quad + CmT(\Delta_{\text{boot}} + \delta + N^{-8} + \epsilon_{\text{samp}}). \end{aligned} \tag{58}$$

The choice of C_1 makes $W = C_1 m^2 K^2 b_N$ dominate the coefficients of ηT in $\mathcal{L}_\eta + \mathcal{T}_\eta$. Indeed, $C_{\text{tr}} = O(mK\Lambda)$, $L_{\text{stab}} = O(\Lambda)$, and $b_N \asymp \Lambda$. Since $\log \binom{K}{m} \leq (\log 2)\mathcal{J}$, the combined inequality becomes

$$\begin{aligned} R_T(r) &\leq C \left[Q_{\text{boot}} + mpT + mK\Lambda + V + \frac{\mathcal{J}}{\eta} + W\eta T + \frac{mT}{\eta D} \right. \\ &\quad \left. + mT(\Delta_{\text{boot}} + \delta + N^{-8} + \epsilon_{\text{samp}}) \right]. \end{aligned} \tag{59}$$

Bounds for the terms. Appendix B.9 gives $d\ell_{\text{col}}/S_{\text{col}} \leq \epsilon_{\text{arr}}$ and $L/A_{\text{key}} \leq \epsilon_{\text{arr}}/(8e)$. Using $B_{\text{blk}} \leq T/n$ in (13), then $\epsilon_{\text{arr}} = c\epsilon K^3 b_N^2/(m^3 T)$, gives

$$mT\Delta_{\text{boot}} \leq C \frac{m^3 T^2 \epsilon_{\text{arr}}}{n} + O(1) = O\left(\frac{K^3 T b_N^2}{n}\right) + O(1) = O(V).$$

The remaining terms of (13) contribute $O(1)$ because $mT/N = O(1)$ and $m^2 T^2 e^{-C_0\Lambda} = O(1)$. Also $L_{\text{bit}} \geq 64 \log_2 N$ and $\delta = N^{-8}$ imply

$$mT(\delta + N^{-8} + \epsilon_{\text{samp}}) \leq CmT(N^{-8} + (K^2 + b_N)TN^{-64}) = O(1).$$

The dyadic choice of η in (51) gives

$$\frac{\mathcal{J}}{\eta} + W\eta T \leq C\sqrt{\mathcal{J}WT} \leq CmK\sqrt{T\Lambda \log\left(\frac{K}{m}\right)} \leq CV.$$

Since $D \geq N^2$,

$$\frac{mT}{\eta D} \leq \frac{2mT}{D} \sqrt{\frac{WT}{\mathcal{J}}} \leq \frac{C}{K} \sqrt{\frac{\Lambda}{\mathcal{J}T}} = O(V).$$

Finally, $pT \leq \sqrt{T}$ and $1/p \leq 4 + 2\sqrt{T}$ yield $Q_{\text{boot}} = O(V)$ and $mpT + mK\Lambda = O(V)$.

Substitution in (59) proves $R_T(r) = O(V)$ for every rounded matrix on the fixed public output.

For an arbitrary original reward matrix r^{orig} , rounding down changes each comparator reward by at most m/D per round, whereas the actual unrounded reward is at least the rounded reward. Consequently,

$$R_T(r^{\text{orig}}) \leq R_T(r) + \frac{mT}{D} = O(V).$$

All bounds hold simultaneously for every rounded matrix on the same output in $\mathcal{G}_{\text{pub}}$, and the rounding comparison is pathwise. This proves the required quantifier over all oblivious reward matrices.

On the complementary parameter branch use a fixed assignment to m distinct arms. For two m -sets S, S' , $|S \setminus S'| \leq k_\star$, so $r_t(S) - r_t(S') \leq k_\star$. The resulting regret is at most $k_\star T \leq C_b \widehat{V} = O(V)$. This proves (3) for every horizon. The computational bounds follow from Section 9 below. $\square$

9 Finite computation and sampling

We bound the computations performed by the preceding protocol. At each commitment, Algorithm 7 draws an assignment using Algorithm 9 below: the initial and post-grid draws use one product distribution, and a within-episode draw after collection uses three signed product terms. The two-term refresh in Algorithm 3 belongs to the logical execution; the physical coordinator implements its accumulated effect at commitment. Algorithm 10 constructs the finite public rows used for these draws and for control. During play, Algorithm 12 calls the bootstrap decoder Algorithm 11 at complete block boundaries, whereas Algorithm 8 calls the control decoder Algorithm 4 on enabled DOWN and UP channels.

Arithmetic model. We use a random-access bit-cost model with numerical inputs K, m, T . For a reward observation $Y \in [0, 1]$, the feedback interface supplies $\lfloor DY \rfloor$, including D when $Y = 1$. Thus processing an arbitrary real reward requires only its prescribed quantization. All subsequent operations use integers, rational numbers, and random bits.

9.1 A direct sampler for the assignment law

For $1 \leq r \leq 3$, let $v^{(1)}, \dots, v^{(r)} \in \mathbb{Q}_{>0}^K$ be weight vectors and let $c_1, \dots, c_r \in \mathbb{Q}$ be coefficients. The coefficients may have either sign. Define the mass of an m -set by

$$f(S) = \sum_{\ell=1}^r c_\ell \prod_{i \in S} v_i^{(\ell)}. \quad (60)$$

For each component, define suffix sums

$$E_\ell(i, k) = \sum_{\substack{A \subseteq \{i, \dots, K\} \\ |A|=k}} \prod_{j \in A} v_j^{(\ell)}.$$

An empty product is one; an impossible subset size gives zero. In particular, $E_\ell(K+1, 0) = 1$ and $E_\ell(K+1, k) = 0$ for $k \neq 0$. The recurrence is

$$E_\ell(i, k) = E_\ell(i+1, k) + v_i^{(\ell)} E_\ell(i+1, k-1). \quad (61)$$

Algorithm 9 Direct subset sampling and orientation

Require: Weight vectors and coefficients in (60), with $f(S) \geq 0$ and $\sum_{|S|=m} f(S) = 1$; anchor $\bar{w}$.

- 1: Compute $E_\ell(i, k)$ by (61) for $i = K, \dots, 1$ and $k = 0, \dots, m$.
 - 2: Set $S \leftarrow \emptyset$, $k \leftarrow m$, and $a_\ell \leftarrow c_\ell$ for $\ell = 1, \dots, r$.
 - 3: **for** $i = 1, \dots, K$ **do**
 - 4: Compute $W = \sum_{\ell=1}^r a_\ell E_\ell(i, k)$ and $W_{\text{in}} = \sum_{\ell=1}^r a_\ell v_i^{(\ell)} E_\ell(i+1, k-1)$.
 - 5: **if** $W \leq 0$ or $W_{\text{in}} \notin [0, W]$ **then**
 - 6: Enter the local error state of Appendix A.1 and stop.
 - 7: **end if**
 - 8: **if** an independent uniform coordinate is less than W_{in}/W **then**
 - 9: Set $S \leftarrow S \cup \{i\}$ and $k \leftarrow k-1$.
 - 10: Set $a_\ell \leftarrow a_\ell v_i^{(\ell)}$ for every ℓ .
 - 11: **end if**
 - 12: **end for**
 - 13: Draw $j \in S$ with probabilities $\alpha_j(S)$ from (16).
 - 14: Return $U = S \setminus \{j\}$.
-

Lemma 23 (Sampling nonnegative sums of products). *Under its stated hypotheses, Algorithm 9 samples S with law f . Its output satisfies*

$$\Pr(U) = \sum_{j \notin U} f(U \cup \{j\}) \alpha_j(U \cup \{j\}).$$

It uses $O(rKm)$ rational operations and at most $K+1$ uniform coordinates. Every reached prefix has $W > 0$, so the local error test never fires under exact sampling.

Proof. Fix $i \in [K]$ and a prefix $A \subseteq \{1, \dots, i-1\}$ reached before arm i is processed. With $k = m - |A|$, its compatible completions are

$$C(A, i) = \{A \cup B : B \subseteq \{i, \dots, K\}, |B| = k\}.$$

The algorithm has $a_\ell = c_\ell \prod_{j \in A} v_j^{(\ell)}$. Expanding the suffix sums in (61) gives

$$\begin{aligned} W &= \sum_{S \in C(A, i)} f(S), \\ W_{\text{in}} &= \sum_{\substack{S \in C(A, i) \\ i \in S}} f(S). \end{aligned}$$

In particular, $0 \leq W_{\text{in}} \leq W$.

We prove by induction on i that the probability of reaching a prefix A equals its completion mass W . At $i = 1$ this mass is $\sum_{|S|=m} f(S) = 1$. If a reached prefix has mass $W > 0$, the two branches in Algorithm 9 have probabilities

$$\Pr(\text{reach } A \cup \{i\}) = W \frac{W_{\text{in}}}{W} = W_{\text{in}},$$

and

$$\Pr(\text{reach } A \text{ after rejecting } i) = W \left(1 - \frac{W_{\text{in}}}{W}\right) = W - W_{\text{in}}.$$

Both equal the completion masses of their respective prefixes. A zero-mass branch is reached with probability zero, so every evaluated denominator is positive. At $i = K + 1$, the completion mass of a reached m -set S is $f(S)$.

Conditional on S , the independent orientation coordinate removes $j \in S$ with probability $\alpha_j(S)$. Summing over j proves the stated law of U .

There are $r(K + 1)(m + 1)$ suffix entries, each using at most one addition and one multiplication. The loop uses $O(rK)$ further operations and at most K coordinates; orientation uses $O(m)$ operations and one coordinate. The totals are $O(rKm)$ exact rational operations and at most $K + 1$ coordinates. $\square$

Lemma 24 (Direct sampling after collection). *Suppose an assignment ends at cutoff u , and collection finishes at commitment t before the next grid boundary and before stability failure. Let $n_{u,t}$ count the nonreserved rounds in $[u, t)$, set $\kappa = (1 - h)^{n_{u,t}+1}/h$, and define*

$$f_t(S) = Q_{w_t}(S) + \kappa(Q_{w_u}(S) - Q_{w_{u-1}}(S)). \quad (62)$$

Then f_t is a probability distribution. Algorithm 9 with the three components

$$(v^{(1)}, c_1) = \left(w_t, \frac{1}{\mathcal{Z}(w_t)}\right), \quad (v^{(2)}, c_2) = \left(w_u, \frac{\kappa}{\mathcal{Z}(w_u)}\right), \quad (v^{(3)}, c_3) = \left(w_{u-1}, \frac{-\kappa}{\mathcal{Z}(w_{u-1})}\right)$$

returns the within-episode commitment law v_t used by Algorithm 7. With one component $(w_t, 1/\mathcal{Z}(w_t))$, its output law is p_{w_t} , the initial and post-grid commitment law in that algorithm. For comparison, the logical refresh law ρ_s in (23) is sampled using the two components

$$(v^{(1)}, c_1) = \left(w_{s+1}, \frac{1}{h\mathcal{Z}(w_{s+1})}\right), \quad (v^{(2)}, c_2) = \left(w_s, -\frac{1-h}{h\mathcal{Z}(w_s)}\right).$$

Proof. For every permitted update before stability failure, Lemma 5 gives

$$Q_{w_{s+1}}(S) \geq \frac{Q_{w_s}(S)}{1+h} \geq (1-h)Q_{w_s}(S).$$

In particular, the update on round $u - 1$ satisfies $Q_{w_u}(S) \geq (1-h)Q_{w_{u-1}}(S)$. Scores do not change on reserved rounds, so iteration over the $n_{u,t}$ nonreserved rounds of $[u, t)$ gives

$$Q_{w_t}(S) \geq (1-h)^{n_{u,t}}Q_{w_u}(S).$$

Substitute $\kappa = (1-h)^{n_{u,t}+1}/h$ into (62). The preceding two inequalities imply

$$\begin{aligned} f_t(S) &= Q_{w_t}(S) + \kappa(Q_{w_u}(S) - Q_{w_{u-1}}(S)) \\ &\geq \frac{(1-h)^{n_{u,t}}}{h}(Q_{w_u}(S) - (1-h)Q_{w_{u-1}}(S)) \\ &\geq 0. \end{aligned}$$

Since each Q has total mass one, $\sum_{|S|=m} f_t(S) = 1$. Thus f_t is a probability distribution.

The anchor $\bar{w}$ is fixed during the episode, so the orientation probabilities $\alpha_j(S)$ are identical in all three terms of f_t . Applying (16) term by term gives

$$\begin{aligned} \sum_{j \notin U} f_t(U \cup \{j\}) \alpha_j(U \cup \{j\}) &= p_{w_t}(U) + \kappa(p_{w_u}(U) - p_{w_{u-1}}(U)) \\ &= v_t(U). \end{aligned}$$

The last equality is (45), and Lemma 23 now gives the three-component sampler. With one component the same lemma samples Q_{w_t} and then p_{w_t} after orientation.

For the two-component logical refresh at round s , the set mass is

$$\frac{Q_{w_{s+1}}(S) - (1-h)Q_{w_s}(S)}{h} \geq 0.$$

Its total mass is $(1 - (1-h))/h = 1$. The orientation identity and (23) give the law ρ_s . $\square$

The point of Lemmas 23, 24 and Algorithm 9 is computational. Directly tabulating the commitment law over all m -subsets would require $\Theta(\binom{K}{m})$ probabilities, which is exponential in general. Lemma 24 represents the required law using three product components, and Lemma 23 samples from this representation in $O(Km)$ rational operations without enumerating the state space.

Lemma 25 (Finite sampler coupling). *For every fixed reward matrix and reserved row, we can couple the finite and ideal reference executions as follows. Define their projected behavioral trace to contain actions, observations, comparison and decoder outcomes, encoded and accepted words, and all score, clock, and local-state registers. Exclude raw sampler coordinates. In a saved record (s, j, y, v) , replace the raw field v by its vector of all admissible future contributions*

$$(\psi_k(y, v))_{k \in \mathbb{Z} \cap [0, 4b_N]},$$

where ψ_k is defined in (41). This projection is for analysis only; the implementation still stores v . Let Tr_{ideal} and $\text{Tr}_{\text{finite}}$ denote these projected traces. Then

$$\Pr(\text{Tr}_{\text{ideal}} \neq \text{Tr}_{\text{finite}}) \leq C(K^2 + b_N)T2^{-L_{\text{bit}}} = \epsilon_{\text{samp}}.$$

For any functional of Tr with range in $[-mT, mT]$, the two expectations differ by at most $2mT\epsilon_{\text{samp}}$; for any event determined by Tr , the two probabilities differ by at most ϵ_{samp} .

Proof. Put $\epsilon = 2^{-L_{\text{bit}}}$ and $C := C_{\text{row}}$, the fixed coordinate namespace defined in Subsection B.4. For the selected row J , abbreviate $V_{s,c} := V_{J,s,c}$. For $(s, c) \in [T] \times C$, couple $V_{s,c} \sim \text{Unif}([0, 1])$ to $V_{s,c}^{(L)} = \epsilon \lfloor V_{s,c} / \epsilon \rfloor$. Couple the masks, certified code, bootstrap arrays, and affine-hash bits identically in the two executions. Their bounded-sampler exhaustion is accounted for separately in the public-construction analysis. The row is sampled before play, but its coordinates are exposed in the order the protocol reads them. If the same threshold $a \in [0, 1]$ is applied to a paired coordinate, then unequal Boolean results imply

$$V_{s,c} \in [a, a + \epsilon]. \tag{63}$$

A draw with q outcomes has at most $q - 1$ cumulative thresholds.

Algorithm 10 reserves independent coordinates for each potential call site in a control row, even when its branch is not taken. The injective tuple $(R_0, \dots, R_d)$ of Section 6 can be generated by successive uniform

draws from sets of sizes $K, K-1, \dots, K-m+1$. Its r th draw, $0 \leq r < m$, has $K-r-1$ cumulative boundaries. On DATA, Algorithm 5 uses the m independent pilot arms $P_0, \dots, P_d$, each with $K-1$ boundaries; its coordinator fast-arm draw has $|[K] \setminus U| - 1 = K-m$ boundaries.

The row's three-way slot choice has two boundaries. Its selected follower and DATA recorder-source choices have $d-1$ and $m-1$; the proposal bit of Algorithm 3 and the immediate thinning test of Algorithm 6 have one each. In Algorithm 9, the loop over $i \in [K]$ makes at most K binary inclusion comparisons, and the final orientation among m arms has at most $m-1$ boundaries. Therefore, with B_{row} denoting the boundary budget for comparisons made at that round and for its fixed proposal bit,

$$\begin{aligned} B_{\text{row}} &:= \sum_{r=0}^{m-1} (K-r-1) + m(K-1) + (K-m) \\ &\quad + [2 + (d-1) + (m-1) + 1 + 1] + K + (m-1) \\ &= 2mK + 2K - \binom{m}{2} \leq 2K^2 + 2K = O(K^2). \end{aligned}$$

There are at most $2m + K + 7 = O(K)$ coordinate call sites in this allocation.

For each potential call (s, c) , let $\mathcal{H}_{s,c}$ be the sigma-field generated by the masks, code, bootstrap arrays, affine-hash bits, J , private bootstrap randomness, and the protocol history and scalar row coordinates exposed before this call, but not $V_{s,c}$. This is an exposure sigma-field: the protocol has not used the remaining published coordinates to determine the current threshold. Let $A_{s,c}$ be the set of cumulative thresholds determined by $\mathcal{H}_{s,c}$, with $A_{s,c} = \emptyset$ if the call is skipped. The independence of J and the control rows (Proposition 3) gives

$$V_{s,c} \perp\!\!\!\perp \mathcal{H}_{s,c}.$$

Thus the other construction randomness may affect $A_{s,c}$ but does not affect the following conditional estimate. Define

$$E_{\text{on}} := \bigcup_{s=1}^T \bigcup_{c \in C} \bigcup_{a \in A_{s,c}} \{V_{s,c} \in [a, a + \varepsilon)\}.$$

Conditioning before each coordinate and using (63) gives

$$\Pr(E_{\text{on}}) \leq \sum_{s,c} \mathbb{E} \left[\Pr \left\{ V_{s,c} \in \bigcup_{a \in A_{s,c}} [a, a + \varepsilon) \middle| \mathcal{H}_{s,c} \right\} \right] \leq \varepsilon \sum_{s,c} \mathbb{E}|A_{s,c}| \leq TB_{\text{row}}\varepsilon \leq CK^2T\varepsilon. \quad (64)$$

A thinning coordinate saved by an unoccupied follower is tested only after a DOWN acceptance. Its namespace is $c_{\text{thin}} = \text{thin}$, distinct from the independent recorder-source coordinate rec . Write $V_s := V_{s,c_{\text{thin}}}$ and $V_s^{(L)} := V_{s,c_{\text{thin}}}^{(L)}$. If the parsed rate exponent is k , its thinning threshold in (41) is $a_k = 4dK2^{-k}/\zeta$, where $k \in \mathbb{Z} \cap [0, 4b_N]$ by the DOWN parsing range. Define the fixed set and event

$$\begin{aligned} \mathcal{D} &:= \{a_k : k \in \mathbb{Z} \cap [0, 4b_N]\} \cap [0, 1], \\ E_{\text{rec}} &:= \bigcup_{s=1}^T \bigcup_{a \in \mathcal{D}} \{\mathbf{1}\{V_s \leq a\} \neq \mathbf{1}\{V_s^{(L)} \leq a\}\}. \end{aligned}$$

The event includes every recorder position, whether or not it is eventually saved. Since $|\mathcal{D}| \leq 4b_N + 1$, (63) and a union bound give

$$\Pr(E_{\text{rec}}) \leq T(4b_N + 1)\varepsilon. \quad (65)$$

This bound is uniform over the exponent selected by any later parsed DOWN word, including an incorrectly accepted one. If $a_k > 1$, both thinning indicators equal one, so it needs no entry in $\mathcal{D}$. Consequently, by (41), on E_{rec}^c we have

$$\psi_k(y, V_s) = \psi_k(y, V_s^{(L)}) \quad \text{for all } s \in [T], k \in \mathbb{Z} \cap [0, 4b_N], y \in [0, 1]. \quad (66)$$

On $(E_{\text{on}} \cup E_{\text{rec}})^c$, induction over local operations gives identical projected traces. Online comparisons agree until the first possible difference, and (66) makes every admissible later use of paired saved records agree. The raw fields V_s and $V_s^{(L)}$ need not be numerically equal and are not part of the projection. Equations (64) and (65) yield the claimed probability bound.

Let $F, F_L \in [-mT, mT]$ be the values of the same functional of $\text{Tr}_{\text{ideal}}, \text{Tr}_{\text{finite}}$. They agree when the traces agree, so

$$|\mathbb{E}F - \mathbb{E}F_L| \leq \mathbb{E}|F - F_L| \leq 2mT \Pr(F \neq F_L) \leq 2mT \epsilon_{\text{samp}}.$$

Applying the same inequality to event indicators gives the event bound. $\square$

The complete reference regret and the decoder and stability events are determined by this projected trace. Thus replacing the ideal coordinates by their L_{bit} -bit implementations changes those expectations by at most $2mT \epsilon_{\text{samp}}$ and those event probabilities by at most ϵ_{samp} , as required in the main proof.

9.2 Storage and bit complexity

Proposition 26 (Finite implementation complexity). *On the construction branch, the public description has $O(N^4 \log N)$ bits and can be constructed using $\tilde{O}(N^4)$ bit operations. The total bootstrap-decoding cost is $\tilde{O}(N^{5/2})$, and the total post-initialization communication-decoding cost is $O(mT \log^3 N)$ on every execution. The latter improves to $O(T \log^3 N)$ when all active players use the same row. All remaining online computations have polynomial bit complexity in K, m, T .*

Moreover, the bounded integer-sampling implementation is total, and the probability that any capped constructor or private-bootstrap draw reaches its cap is at most N^{-40} .

Proof. We use the parameter estimates established in the proof of Lemma 29:

$$mB_+ = O(\sqrt{N}), \quad S_{\text{col}} = O(N^{3/2}), \quad Q_{\text{arr}} = O(N^2),$$

together with

$$mT \leq N, \quad K^2 T \leq N^2,$$

and the fact that every arm, message, column, and sampler coordinate has $O(\log N)$ bits.

Public construction. The raw arm arrays contain

$$O(mTS_{\text{col}})$$

entries, while the stored message permutations contain

$$O(mB_+ S_{\text{col}} Q_{\text{arr}}) = O(N^4)$$

labels. The reserved masks use $MT \leq N^3$ bits. By Lemma 25, the finite learning/control rows contain $O(K^2)$ $O(\log N)$ -bit coordinates per row and time, and hence use

$$O(MTK^2 \log N) = O(N^4 \log N)$$

bits. The certified code is only polylogarithmic in N . Thus the entire published description has $O(N^4 \log N)$ bits.

The same counts show that the constructor performs at most $O(N^4)$ finite sampling, permutation, and table-entry operations. Each such operation involves $O(\log N)$ -bit quantities and bounded rejection sampling with $O(\log N)$ attempts. Hence the total construction cost is $\tilde{O}(N^4)$ bit operations.

Decoding. At a complete bootstrap block boundary, checking one sender column requires a pass over at most n positions and Gaussian elimination in $O(\log N)$ binary unknowns. Its cost is therefore

$$O(n \log^2 N).$$

Each receiver has at most three neighboring senders, so the number of sender-column checks over all blocks is $O(mB_+S_{\text{col}})$. Since $B_+n \leq 2T$,

$$\begin{aligned} O(mB_+S_{\text{col}}n \log^2 N) &= O(mTS_{\text{col}} \log^2 N) \\ &= \tilde{O}(N^{5/2}). \end{aligned}$$

For the post-initialization affine channels,

$$B, Q_{\text{dec}} = O(\log N).$$

A full decoder decision solves at most Q_{dec} equations in B binary unknowns and hence costs $O(\log^3 N)$ bit operations. Each player makes at most one channel decision on a calendar round, so the unconditional total cost is

$$O(mT \log^3 N).$$

When all active players use the same row, its common slot and selected follower enable at most one receiver channel per round. This gives $O(T \log^3 N)$ for synchronized executions, including the reference.

Remaining online computation. It remains only to check that exact learner and scheduling arithmetic uses polynomial-size operands and polynomially many operations.

The quantization denominator satisfies $D = O(N^2)$ and every score lies in $[0, TD]$. Hence an exact weight

$$w_i = \left(\frac{D+1}{D} \right)^{z_i}$$

has numerator and denominator of polynomial bit length. Products, elementary-symmetric-sum dynamic programs, and the at most three rational components used by the commitment sampler therefore also have polynomial bit length. The proposal probability h has $O(\log N)$ bits, while

$$\kappa = \frac{(1-h)^{n_{u,t}+1}}{h}, \quad n_{u,t} \leq T,$$

has polynomial bit length as well. All message fields have $O(\log N)$ bits.

There are at most T commitments and $O(mT)$ local round updates. The loops appearing in the learner, recorder, scheduler, and direct sampler have lengths polynomial in K and m , and the sparse observation lists contain at most T records per follower. Consequently all remaining online computations use polynomially many arithmetic, comparison, hashing, and list operations on polynomial-bit operands. Their total bit complexity is therefore polynomial in K, m, T .

Finally, the constructor and private bootstrap together make $O(N^4)$ bounded uniform-integer draws, and hence at most N^5 such draws after enlarging the fixed branch constant if necessary. A single rejection attempt succeeds with probability at least $1/2$, so with

$$a_{\text{cap}} = \lceil 45 \log_2 N \rceil$$

a particular call reaches its cap with probability at most $2^{-a_{\text{cap}}}$. Therefore

$$\Pr\{\text{some bounded sampler reaches its cap}\} \leq N^5 2^{-a_{\text{cap}}} \leq N^{-40}.$$

The prescribed fallback rules define the execution even on this event, so the finite implementation is total. $\square$

A Exceptional executions

The decoder estimates apply through the first weight doubling. A decoder failure does not terminate or condition the logical learner in the reference execution.

A.1 A total local algorithm

The following convention defines every action even after an erroneous declaration or decode. It uses one absorbing error state, denoted $\dagger$, for each player. Before performing a local operation, check that its inputs belong to their stated finite domains. In particular, arm labels must belong to $[K]$, scores to $\mathbb{Z} \cap [0, TD]$, increments to $\mathbb{Z} \cap [0, D]$, and time fields to $[0, T + 1]$. Every required denominator must be positive. Each Bernoulli parameter, including a thinning or mixture parameter, must belong to $[0, 1]$. For the slow-set sampler, apply these checks to the scalar probabilities and normalizers that its successive sampling steps actually evaluate (Section 9.1). The local record must specify exactly one applicable action branch. A computed value is checked before it is stored or used in another operation.

Let Φ be one specified local operation, with input domain $\text{dom}(\Phi)$ and allowed output set $\mathcal{D}_\Phi$. Apply its total extension

$$\overline{\Phi}(x) = \begin{cases} \Phi(x), & x \neq \dagger, x \in \text{dom}(\Phi), \Phi(x) \in \mathcal{D}_\Phi, \\ \dagger, & \text{otherwise.} \end{cases} \quad (67)$$

The first branch evaluates the operation only after its input-domain checks. Once the state is $\dagger$, no learning or transport update is performed.

A decoder rejection is an ordinary operation and does not enter $\dagger$. In the error state the player plays arm $i + 1$ on active nonreserved rounds and continues its bootstrap-array action on declared reserved rounds. Since $0 \leq i \leq d = m - 1 < K$, this arm always exists. Bootstrap updates and activation retain their own specified finite-domain convention.

Lemma 27 (The error state is unused on valid executions). *Use the construction-branch rate choice of Section 8 and suppose the public constructor has published a table. If the bootstrap states and activation times agree with the reference execution, every accepted message is the message currently sent, and the feasibility and stability bounds of Section 5 hold through round t . Then no learning or transport operation enters $\dagger$ by round t .*

Proof. Order the learning and transport operations through round t by round, within-round step, and player label, and write them as $\mathfrak{o}_1, \dots, \mathfrak{o}_L$. We prove by induction on q that after $\mathfrak{o}_1, \dots, \mathfrak{o}_q$ no player is in $\dagger$, the physical and reference local registers agree, and the registers and calendar-round scores satisfy the following bounds. For $1 \leq s \leq t$, $0 \leq v \leq t + 1$, $j \in [K]$, and $i \in [d]$, they are

$$\begin{aligned} \Delta_s(j) &\in \mathbb{Z} \cap [0, D], & Z_j(v) &= \sum_{r < v} \Delta_r(j) \in \mathbb{Z} \cap [0, TD], \\ S_i, c_j &\in \mathbb{Z} \cap [0, TD], & \delta_i &\in \mathbb{Z} \cap [0, D], \\ \text{state}_i &\in \{0, 1, 2\}, & \mathcal{P} &\subseteq [d]. \end{aligned} \quad (68)$$

The assertion holds for $q = 0$: bootstrap states and activation times agree by hypothesis, $z = S = \delta = 0$, the UP placeholder is $(0, 0, 0)$, and the published table supplies symbols in their prescribed alphabets. Assume the assertion holds through $\mathfrak{o}_q$ for some $q < L$. Since every accepted word is the currently encoded word, Lemmas 17, 18, and 19 apply to this common prefix.

Increments and score registers. For each selected owner i and arm j , feasibility gives $N^{-4} \leq \xi_j$ and $\eta \leq \xi_j \leq 2/(AmK)$. Since $Y_{i,s} \in [0, 1]$, the recorder's increment satisfies

$$\Delta_s(j) = \begin{cases} \lfloor D\eta Y_{i,s}/\xi_j \rfloor, & I_s = j, \\ 0, & I_s \neq j, \end{cases} \quad 0 \leq \Delta_s(j) \leq D.$$

Every score and collection sum contains at most T such increments, so it lies in $[0, TD]$. Lemma 17 gives the same bound for a follower's sum after delayed DOWN acceptance.

At a commitment, either $\sigma = 0$ or $\mathcal{P} = \emptyset$ by Algorithm 7. In the first case the coordinator sets $z(t) = 0 = Z(t)$. In the second, Lemma 15 gives

$$\begin{aligned} z_{a_i}(u) &= z_{a_i}(\sigma) + S_i = Z_{a_i}(u), \\ z(t) &= z(u) + c(t) = Z(t), \\ Z(u-1) &= Z(u) - \Delta_{u-1}. \end{aligned}$$

Lemma 18 supplies the asserted value of S_i ; the induction hypothesis supplies $z(\sigma) = Z(\sigma)$. Hence the reconstructed scores lie in $[0, TD]$.

Sampling operations and probabilities. The parameter and stability bounds yield

$$\begin{aligned} w_j &= (1 + 1/D)^{z_j} > 0, \\ 0 < h &= 4AmK\eta \leq \frac{1}{4}, \quad 0 \leq \gamma \leq 1. \end{aligned}$$

Lemma 5 gives $\rho_s \geq 0$ and $\sum_U \rho_s(U) = 1$. For its direct sampler, the underlying m -set mass is

$$f_s^\rho(S) = \frac{Q_{w_{s+1}}(S) - (1-h)Q_{w_s}(S)}{h} \geq 0, \quad \sum_{|S|=m} f_s^\rho(S) = 1.$$

Lemmas 20 and 24 give the same properties for the within-episode mass f_t ; at an initial or grid commitment the mass is Q_{w_t} .

Write f for any of these nonnegative m -set laws. At a sampler prefix (A, i) , its total and inclusion masses are

$$\begin{aligned} W &= \sum_{S \in \mathcal{C}(A, i)} f(S), \\ W_{\text{in}} &= \sum_{\substack{S \in \mathcal{C}(A, i) \\ i \in S}} f(S), \quad 0 \leq W_{\text{in}} \leq W. \end{aligned}$$

Initially $W = 1$. If $W > 0$ and the coordinate $v \in [0, 1)$, the chosen branch has mass

$$W' = \begin{cases} W_{\text{in}}, & v < W_{\text{in}}/W, \\ W - W_{\text{in}}, & v \geq W_{\text{in}}/W \end{cases} > 0.$$

Indeed, the first branch requires $W_{\text{in}} > vW \geq 0$; the second requires $W_{\text{in}} \leq vW < W$. Thus $W > 0$ at every reached prefix, also for finite coordinates. At termination $|S| = m$ and $\sum_{k \in S} \bar{w}_k^{-1} > 0$, so orientation chooses $j \in S$. For $U = S \setminus \{j\}$, the fast-arm denominator satisfies

$$\sum_{k \notin U} \frac{w_k}{1 + s_U \bar{w}_k} > 0.$$

Moreover $|U| = d$ and $|[K] \setminus U| = K - m + 1 \geq 2$.

For each selected owner i and arm j , feasibility and (35) also give

$$0 \leq \frac{\xi_j}{(1/2)s_i\pi_i(j)\zeta} \leq 1.$$

The denominator is positive because $s_i\pi_i(j) > 0$ on the owner event and $\zeta > 0$. The arm rule chooses an element of $[K]$, and the recorder chooses at most one $I_s \in [K] \cup \{\perp\}$.

Message fields and local transitions. At a commitment the DOWN word and, at a cutoff, the ENDPOINT word are

$$M_i^{\text{DOWN}} = (\sigma, a_i, e, -\log_2 \xi_{a_i}), \quad M_i^{\text{END}} = (\sigma, S_i, \delta_i).$$

The bounds above and Lemma 16 put these words in their parsing alphabets. The decoder output belongs to $\{\perp\} \cup \{0, 1\}^B$; a nonparsing word is rejected, while an accepted word is current by hypothesis. Lemmas 18 and 19 give $\text{state}_i \in \{0, 1, 2\}$ and $\mathcal{P} \subseteq [d]$. At the first cutoff, $c = 0$ and $\mathcal{P} = [d]$. The cutoff lies in $[T + 1]$, and the coefficient remains defined since

$$\kappa \in \left\{0, \frac{1-h}{h}(1-h)^q : q \in \{0, \dots, T\}\right\}, \quad h > 0.$$

The local record, published row, and current observation determine exactly one action branch. Thus $\mathbf{o}_{q+1}$ has a valid input, produces an allowed output, and makes the same update in the physical and reference executions. It does not enter $\dagger$ and preserves (68), proving the induction step.

Taking $q = L$ proves the claim through round t . □

A.2 Decoder error bound

Proof of Lemma 8. Work in the product-stream reference execution, through the first round of a stability or feasibility failure; stop each batch thereafter or at the horizon. Put $Q = Q_{\text{dec}}$. For one completed batch on a directed channel, let $r_1 < \dots < r_Q$ be its positive-test rounds, and let $M_r \in \mathbb{F}_2^B \cup \{\perp\}$ be the word encoded on that channel at round r .

An admissible wrong word was not sent in its batch. Suppose a candidate M passes the acceptance test at a decision time t but $M \neq M_t$. Then

$$M_{r_q} \neq M \quad (q \in [Q]). \tag{69}$$

Indeed, suppose $M_{r_q} = M$ for some q . On a DOWN channel, M is a fixed assignment record with start field $\sigma = M.\sigma$. Let $\mathcal{P}_t^-$ be the value of $\mathcal{P}$ at the arm-choice step of round t , and let $\sigma_i(t^-)$ be follower i 's last accepted DOWN start before its decoder decision at t .

Suppose first that the coordinator still retains the assignment starting at σ . On a DOWN round selecting i , it encodes M whenever $i \in \mathcal{P}_t^-$ by Algorithm 5. Hence $M_t \neq M$ implies $i \notin \mathcal{P}_t^-$. If t precedes this assignment's cutoff, i was removed from $\mathcal{P}$ by a DOWN receipt in an earlier round. Lemma 11 implies that i was occupied when that receipt occurred. At the start of this assignment i was unoccupied, and in the reference execution it can become occupied only by accepting M ; thus i accepted M before round t . If t is at or after the cutoff, $\mathcal{P}$ was reset to $[d]$ there, so $i \notin \mathcal{P}_t^-$ implies that the coordinator accepted i 's ENDPOINT before round t . By Lemma 19, that ENDPOINT is offered only after i accepts M .

Suppose instead that a successor assignment has been committed by round t . By Lemma 19, the coordinator accepted i 's ENDPOINT for the σ -assignment before that commitment. Hence i accepted M in both cases. Since accepted DOWN start fields strictly increase, every case gives $\sigma_i(t^-) \geq M.\sigma$, contradicting the DOWN acceptance condition $\sigma_i(t^-) < M.\sigma$ in (40).

On an UP channel, acceptance requires $M.\sigma = \sigma > 0$ and $i \in \mathcal{P}$, so M is not the initial placeholder. By Lemma 19, this assignment's ENDPOINT remains fixed until a newer DOWN record is applied. A successor assignment cannot be committed before this ENDPOINT is accepted. Hence, if $M_t \neq M$, either $i \notin \mathcal{P}$ or the current assignment has start time greater than $M.\sigma$. Both contradict UP acceptance. This proves (69).

One-batch estimate. Fix a channel (UP or DOWN), a batch start α , and a deterministic candidate $M \in \mathbb{F}_2^B$. Recall from Section 6.1 that this channel's round- r hash is $H_r(x) = c_r \cdot x \oplus b_r$, with a fresh pair (c_r, b_r) . Let τ_q be the round of the q -th positive test in this batch before stopping, with $\tau_q = \infty$ if that test does not occur. Set

$$D_q := \{\tau_q < \infty, M_{\tau_j} \neq M, H_{\tau_j}(M) = 1 \text{ for all } j \leq q\}.$$

Conditional on the history $\mathcal{H}_\alpha$ at the batch start, we claim

$$\Pr(D_Q \mid \mathcal{H}_\alpha) \leq 2^{-Q}. \quad (70)$$

To prove this, let $\mathcal{H}_r^-$ contain the information available before (c_r, b_r) is read. Let A_r indicate that round r adds a positive test to this batch. If $M_r \notin \{M, \perp\}$, the affine hash values $(H_r(M_r), H_r(M))$ are independent fair bits conditional on $\mathcal{H}_r^-$. If $M_r = \perp$, the channel action and A_r do not use this hash, including before activation. Consequently,

$$\mathbb{E}[A_r \mathbf{1}\{M_r \neq M, H_r(M) = 1\} \mid \mathcal{H}_r^-] = \frac{1}{2} \mathbb{E}[A_r \mathbf{1}\{M_r \neq M\} \mid \mathcal{H}_r^-]. \quad (71)$$

Write N_{r-1} for the number of tests already in the batch before round r , and let $I_{q,r}$ indicate that the batch is active, $N_{r-1} = q - 1$, and D_{q-1} holds. This indicator is $\mathcal{H}_r^-$ -measurable. Summing (71) over the possible round of the q -th positive test gives

$$\Pr(D_q \mid \mathcal{H}_\alpha) = \frac{1}{2} \mathbb{E} \left[\sum_{r=\alpha}^T I_{q,r} A_r \mathbf{1}\{M_r \neq M\} \mid \mathcal{H}_\alpha \right] \leq \frac{1}{2} \Pr(D_{q-1} \mid \mathcal{H}_\alpha).$$

The inequality holds because the q -th test can occur only once. Iteration for $q = 1, \dots, Q$ proves (70).

Union bound and decoding time. Let E_{false} be the event that a completed batch has a unique solution $M \neq M_t$ passing the acceptance test. Such an M solves every retained equation, so $H_{r_q}(M) = 1$; (69) supplies $M_{r_q} \neq M$. There are at most $2d$ channels (UP or DOWN for each follower), and each has $(T + 1)$ possible batch-start rounds $\alpha \in [T + 1]$. Thus there are at most $2d(T + 1)$ batches and 2^B candidate words per batch. Since $N = 2Km(T + 1)$, (26) and (70) give

$$\Pr(E_{\text{false}}) \leq 2d(T + 1)2^{B-Q} \leq N^8 2^{B-Q} \leq \frac{\delta}{2N^4}.$$

Let E_{amb} be the event that a batch consisting of positive tests of one fixed awaited word M^* has more than one solution. The transmitted M^* solves every equation. Any other solution M satisfies $M_{r_q} = M^* \neq M$ and $H_{r_q}(M) = 1$ for all q . The same conditional estimate and union bound yield

$$\Pr(E_{\text{amb}}) \leq \frac{\delta}{2N^4}.$$

The bound is uniform when M^* is chosen from the batch-start history. The union bound gives

$$\Pr(E_{\text{false}} \cup E_{\text{amb}}) \leq \delta/N^4.$$

On $(E_{\text{false}} \cup E_{\text{amb}})^c$, no completed batch has a unique solution different from the word currently encoded on its channel that passes (40), and no completed batch formed entirely from one fixed awaited word has multiple solutions.

Suppose M^* becomes awaited when its receiver's batch contains $n \in \{0, \dots, Q - 1\}$ equations. After $Q - n$ further positive receiver observations, the decoder decides that batch and clears it. The batch may contain equations from a preceding word. If M^* is still awaited, the next Q positive observations form a new batch whose transmitted word is M^* throughout. Hence M^* solves every equation in the new batch; on E_{amb}^c it is the unique solution and passes (40). Thus, if service continues for $(Q - n) + Q \leq 2Q$ further positive observations, M^* is accepted.

Finally, solving Q binary equations in B unknowns costs $O(QB^2) = O(\log^3 N)$ bit operations by Gaussian elimination; parsing costs $O(B)$. This proves all assertions. $\square$

B Public construction and bootstrap communication

This appendix specifies the public table and proves its simultaneous guarantees. The mask and arm-array properties hold for all reward supports and actual bootstrap messages; the finite-row bound holds for all rounded reward matrices. The constructor samples once and does not test these universal properties of its output. As throughout the paper, $c, C > 0$ denote universal constants whose values may change between appearances. All probability arguments in this appendix use the exact sampling laws displayed in the algorithms. The bounded implementation is compared to those laws by the rejection-stream coupling of Section 4, with cap exhaustion charged separately.

B.1 Public construction and reserved masks

Recall from Section 4 that $M = N^2$ is the number of published learning/control rows. Recall also the construction branch (11), the fixed concentration constant C_0 , the reserved-round probability p , the bootstrap block length n , and the finite-coordinate precision $L_{\text{bit}} = \lceil 64 \log_2 N \rceil$. These are precisely the parameters needed by the public constructor below.

Algorithm 10 samples the masks, certified code, bootstrap arrays, and finite learning/control rows in that order, using fresh randomness at each stage.

Algorithm 10 Construct and publish the public table

Require: K, m, T .

- 1: **if** $k_\star T \leq C_b \widehat{V}$ **then**
 - 2: Publish the fixed-arm fallback flag; stop.
 - 3: **end if**
 - 4: Sample $C_{j,t} \sim \text{Bernoulli}(p)$ independently for $j \in [M]$ and $t \in [T]$.
 - 5: Using randomness independent of the masks, construct the certified code G of Subsection B.3.
 - 6: **if** code certification fails **then**
 - 7: Publish the fixed-arm fallback flag; stop.
 - 8: **end if**
 - 9: Sample the label permutations and raw arm-array coordinates of Subsections B.2 and B.3.
 - 10: Independently sample M finite learning/control rows as defined in Subsection B.4, using L_{bit} bits per scalar coordinate and independent fair affine-hash bits.
 - 11: Publish all masks, the certified code G , every label permutation $\pi_{s,b,w}$, all raw arm-array coordinates, and all learning/control rows.
-

On successful construction, the published table has the following components. The index ranges and sampling laws for the bootstrap columns are specified in the cited subsections.

Component	Stored data	Detailed construction
Masks	$C = (C_{q,t})_{q \in [M], t \in [T]}$, with $C_{q,t} \in \{0, 1\}$.	Algorithm 10.
Code	$G \in \mathbb{F}_2^{H_{\text{arr}} \times B_{\text{arr}}}$, shared by all bootstrap columns.	Subsection B.3 and Lemma 28.
Bootstrap arrays	For each player s , block b , and column w , the published permutation $\pi_{s,b,w}$; for each position t , the variables $\lambda_{s,b,w,t}$, $b_{s,b,w}(t)$, $(a_{0,s,b,w,t}, a_{1,s,b,w,t})$, $j_{s,b,w,t}$, and $z_{s,b,w,t}$, determining the arm maps $c_{s,b,w,t}(\nu)$ and $d_{s,b,w,t}(\nu)$.	Subsections B.2 and B.3; Figure 2.
Control rows	$(R_q)_{q \in [M]}$, with the finite coordinate and hash namespaces specified in Subsection B.4.	Subsection B.4.

Table 1: Components of the public table on the construction branch.

The decoder uses the published rows of G in (84). The maps $c_{s,b,w,t}$ and $d_{s,b,w,t}$ are determined by G and the published raw coordinates and need not be sampled separately. The code constructor does not inspect C . Thus G is independent of C , and the primitive bootstrap-column draws have their stated laws conditional on every fixed mask realization. The fixed-mask condition in Lemma 29 is used when proving its reserved-position property A7.

Algorithm 10 specifies exact sampling laws. Its finite realization replaces each finite-set uniform draw by the bounded sampler of (6) and publishes the fixed-arm fallback flag immediately if a public call exhausts its cap. On the event of no public exhaustion, its output agrees with the exact-sampling construction under the rejection-stream coupling.

For the rest of this appendix abbreviate $A := A_{\text{key}}$, the authentication-key alphabet size defined in Section 4. Use the player labels $0, \dots, m-1$ of Section 5, with root 0. Each nonroot i has parent $\lfloor (i-1)/2 \rfloor$ and ordered children $2i+1, 2i+2$ whenever these labels are below m . Write $d_i \leq 2$ for its number of children.

B.2 Scalar messages

Independently of the root's uniform $J \in [M]$, each nonroot player draws a uniform authentication key $U_i \in [A]$. The root has no parent and, since $m \geq 2$, is an informed internal node; its rule never uses a key. Encode $J-1 = J_0 + AJ_1$, with $J_0, J_1 \in \{0, \dots, A-1\}$. Let $\perp$ denote the distinguished absent/unknown symbol, which is not an element of $[A]$. For $u \in [A]$ and $0 \leq c < d_i$, $k \in \{0, 1\}$, $0 \leq v < A$, and $e \in [A] \cup \{\perp\}$, define

$$\text{DISC}(u) := u - 1,$$

and

$$\text{ROOT}(c, k, v, e) := A + ((2c + k)A + v)(A + 1) + \begin{cases} 0, & e = \perp, \\ e, & e \in [A]. \end{cases}$$

As these arguments range over their stated domains, the DISC and ROOT labels give a bijection onto $\{0, \dots, A + 2d_i A(A + 1) - 1\}$. Denote this native alphabet by $\mathcal{M}_i$; thus $|\mathcal{M}_i| = A + 2d_i A(A + 1) = O(A^2)$. Set $B_{\text{arr}} = \lceil \log_2 \max_i |\mathcal{M}_i| \rceil$, the raw-address width, and pad all alphabets to $Q_{\text{arr}} = 2^{B_{\text{arr}}}$ labels; unused labels are invalid.

Recall S_{col} from (12) and abbreviate $S_0 := S_{\text{col}}$. To index both complete blocks and the possible final incomplete block, set

$$B_+ := \left\lceil \frac{T}{n} \right\rceil, \quad I_b^+ := \{(b-1)n+1, \dots, \min\{bn, T\}\}, \quad b \in [B_+].$$

For every player $s \in \{0, \dots, m-1\}$, block $b \in [B_+]$, and column $w \in [S_0]$, store an independent uniform permutation $\pi_{s,b,w}$ of the Q_{arr} labels and its inverse. Raw addresses are $v \in \mathbb{F}_2^{B_{\text{arr}}}$. To send a native label μ in sender–block pair (s, b) , first fix the complete message, then choose a fresh uniform column w and use address $\pi_{s,b,w}^{-1}(\mu)$. When s and b are fixed, we suppress them and write simply π_w .

B.3 Certified code and raw arm arrays

Lemma 28 supplies a short binary linear code with this separation property and an exact polynomial-time certificate.

Lemma 28 (Certified binary code). *For every integer $B \geq 1$, there is a randomized algorithm which, on success, outputs an integer H' and a matrix $G \in \mathbb{F}_2^{H' \times B}$, which satisfies,*

$$H' \leq 1024B, \quad \text{rank}_{\mathbb{F}_2}(G) = B, \quad \min_{x \in \mathbb{F}_2^B \setminus \{0\}} \|Gx\|_1 \geq \frac{H'}{8}. \quad (72)$$

Moreover, each trial uses $\text{poly}(B)$ bit operations and has success probability at least $3/4$. Consequently, after $\lceil 80 \log_2 N \rceil$ independent trials, the probability that no code is produced is at most N^{-100} .

Proof. Set

$$r = \lceil \log_2(4(B+1)) \rceil, \quad q = 2^r, \quad k = \left\lceil \frac{B}{r} \right\rceil. \quad (73)$$

Then

$$4(B+1) \leq q < 8(B+1), \quad 2k \leq 2(B+1) \leq q. \quad (74)$$

We first construct an inner binary code of dimension r . Sample a matrix

$$A \sim \text{Unif}(\mathbb{F}_2^{128r \times r}),$$

meaning that all $128r^2$ entries of A are independent fair bits. For a fixed $u \in \mathbb{F}_2^r \setminus \{0\}$, the coordinates of Au are independent fair bits, and therefore

$$\|Au\|_1 \sim \text{Bin}(128r, 1/2), \quad \mathbb{E}\|Au\|_1 = 64r.$$

The lower-tail Chernoff bound [BLM13, Chapter 2] gives

$$\Pr\{\|Au\|_1 < 32r\} \leq \exp\left(-\frac{(1/2)^2}{2} 64r\right) = e^{-8r}. \quad (75)$$

Hence, by a union bound over the $2^r - 1$ nonzero inputs,

$$\Pr\left\{\min_{u \neq 0} \|Au\|_1 < 32r\right\} \leq (2^r - 1)e^{-8r} < e^{-(8-\log 2)r} < \frac{1}{4}. \quad (76)$$

The trial accepts A precisely when

$$\|Au\|_1 \geq 32r \quad \text{for every } u \in \mathbb{F}_2^r \setminus \{0\}. \quad (77)$$

Because $2^r = q = O(B)$ by (74), this condition is checked exactly by enumerating all nonzero u , using polynomially many bit operations.

Next construct the outer evaluation code. Enumerate monic binary polynomials of degree r until an irreducible polynomial h is found, and identify

$$\mathbb{F}_q = \mathbb{F}_2[Z]/(h(Z)).$$

There are $2^r = O(B)$ monic degree- r candidates, and irreducibility is checked by exact polynomial arithmetic, so this step also takes $\text{poly}(B)$ bit operations. Let

$$\phi : \mathbb{F}_q \longrightarrow \mathbb{F}_2^r$$

be the coordinate isomorphism in the polynomial basis $1, Z, \dots, Z^{r-1}$, and define the certified inner encoder

$$E(a) = A\phi(a) \in \mathbb{F}_2^{128r}.$$

Choose $2k$ distinct elements $\beta_1, \dots, \beta_{2k} \in \mathbb{F}_q$, which is possible by (74).

For $x \in \mathbb{F}_2^B$, append $kr - B$ zero bits and split the resulting kr -bit vector into k consecutive blocks in $\mathbb{F}_2^r$. Via ϕ^{-1} , denote the corresponding field elements by $a_0(x), \dots, a_{k-1}(x) \in \mathbb{F}_q$, and set

$$p_x(X) = \sum_{s=0}^{k-1} a_s(x) X^s \in \mathbb{F}_q[X]. \quad (78)$$

Define

$$C(x) = (E(p_x(\beta_1)), \dots, E(p_x(\beta_{2k}))) \in \mathbb{F}_2^{H'}, \quad H' = 2k(128r) = 256kr. \quad (79)$$

Every map appearing in (79) is $\mathbb{F}_2$ -linear, so there is a binary matrix $G \in \mathbb{F}_2^{H' \times B}$ such that $C(x) = Gx$.

If $x \neq 0$, then p_x is a nonzero polynomial of degree at most $k-1$. It therefore has at most $k-1$ roots in $\mathbb{F}_q$. Among the $2k$ distinct evaluation points,

$$\#\{j \in [2k] : p_x(\beta_j) \neq 0\} \geq 2k - (k-1) = k+1. \quad (80)$$

For every such j , $\phi(p_x(\beta_j)) \neq 0$, so the exact inner certificate (77) implies

$$\|E(p_x(\beta_j))\|_1 \geq 32r.$$

Summing the ℓ_1 norm of the $2k$ disjoint blocks in (79) gives

$$\|Gx\|_1 = \|C(x)\|_1 \geq 32r(k+1) \geq 32rk = \frac{H'}{8}. \quad (81)$$

Thus $Gx \neq 0$ for every $x \neq 0$, so $\text{rank}_{\mathbb{F}_2}(G) = B$ as well.

It remains to bound the length and the failure probability. Since $k = \lceil B/r \rceil$, then $kr \leq B + r$. For $B \geq 1$, $4(B+1) \leq 2^{3B}$, hence $r \leq 3B$ and therefore

$$H' = 256kr \leq 256(B+r) \leq 1024B. \quad (82)$$

The matrix G can be materialized by evaluating C on the B standard basis vectors, again in $\text{poly}(B)$ bit operations. Finally, by (76), one trial fails with probability at most $1/4$, and therefore

$$(1/4)^{\lceil 80 \log_2 N \rceil} \leq 2^{-160 \log_2 N} = N^{-160} \leq N^{-100}.$$

This proves the lemma. □

Apply Lemma 28 with $B = B_{\text{arr}}$ and denote its returned matrix by $G \in \mathbb{F}_2^{H_{\text{arr}} \times B_{\text{arr}}}$. Write its rows as $G_j \in \mathbb{F}_2^{B_{\text{arr}}}$, $j \in [H_{\text{arr}}]$.

We now define one raw arm array formally. Recall that α_0 is the listening probability. For every player $s \in \{0, \dots, m-1\}$, block $b \in [B_+]$, column $w \in [S_0]$, and position $t \in I_b^+$, independently sample

$$\begin{aligned} \lambda_{s,b,w,t} &\sim \text{Bernoulli}(\alpha_0), & b_{s,b,w}(t) &\sim \text{Unif}([K]), \\ (a_{0,s,b,w,t}, a_{1,s,b,w,t}) &\sim \text{Unif}\{(a_0, a_1) \in [K]^2 : a_0 \neq a_1\}, \\ j_{s,b,w,t} &\sim \text{Unif}([H_{\text{arr}}]), & z_{s,b,w,t} &\sim \text{Unif}(\mathbb{F}_2). \end{aligned}$$

Thus $j_{s,b,w,t}$ is the sampled code-row index and $z_{s,b,w,t} \in \{0, 1\}$ is the fair binary offset. Define the listening set of column (s, b, w) by

$$\mathcal{L}_{s,b,w} := \{t \in I_b^+ : \lambda_{s,b,w,t} = 1\}.$$

When the player s and block b are fixed, we suppress those indices and write $\mathcal{L}_w$, $b_w(t)$, (a_0, a_1) , $j_{w,t}$, and $z_{w,t}$. Set

$$\begin{aligned} c_{v,w}(t) &= a_{G_{j_{w,t}} \cdot v \oplus z_{w,t}}, \\ d_{v,w}(t) &= \begin{cases} b_w(t), & t \in \mathcal{L}_w, \\ c_{v,w}(t), & t \notin \mathcal{L}_w. \end{cases} \end{aligned} \tag{83}$$

With all indices restored, the address-dependent arm maps in Table 1 are

$$c_{s,b,w,t}(v) := c_{v,w}(t), \quad d_{s,b,w,t}(v) := d_{v,w}(t) \quad (v \in \mathbb{F}_2^{B_{\text{arr}}}, t \in I_b^+).$$

The fresh offset makes any fixed candidate's arm uniform independently of its row index. Symbols are independent across positions and columns; addresses in the same column are correlated. The permutations are therefore retained explicitly and are not replaced by identity labels.

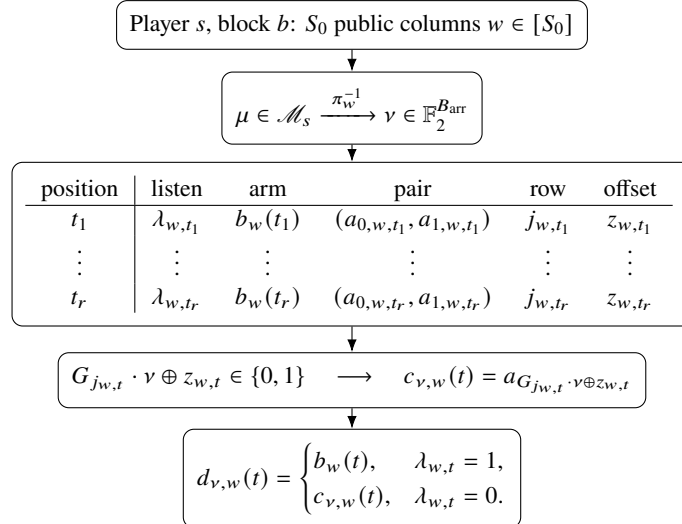


Figure 2: One public arm-array column, where $I_b^+ = \{t_1 < \dots < t_r\}$. The code matrix G is shared across columns; a_0, a_1 in the candidate formula are the pair stored at position t . Player and block indices are suppressed inside the column.

Thus a column stores π_w and $(\lambda_{w,t}, b_w(t), (a_{0,w,t}, a_{1,w,t}), j_{w,t}, z_{w,t})_{t \in I_b^+}$. The functions $c_{v,w}$ and $d_{v,w}$ are computed from these data and the shared matrix G .

B.4 Finite learning/control rows

Let $B = 4\lceil \log_2(TD + 1) \rceil$ be the message width from Section 6. The fixed scalar-coordinate namespace is

$$\begin{aligned} C_{\text{row}} = & \{\text{slot}, \text{sel}, \text{rec}, \text{fast}, \text{thin}, \text{prop}\} \\ & \cup \{\text{tuple}_r, \text{pilot}_r : 0 \leq r < m\} \cup \{\text{commit}_r : 1 \leq r \leq K + 1\}. \end{aligned}$$

It has $2m + K + 7$ elements. For each $(q, t, c) \in [M] \times [T] \times C_{\text{row}}$, independently draw an integer $Z_{q,t,c}$ uniformly from $\{0, \dots, 2^{L_{\text{bit}}} - 1\}$ using L_{bit} fair bits, and publish

$$V_{q,t,c}^{(L)} = 2^{-L_{\text{bit}}} Z_{q,t,c}.$$

For analysis, the coupled ideal row instead uses independent $V_{q,t,c} \sim \text{Unif}([0, 1))$, with $V_{q,t,c}^{(L)} = 2^{-L_{\text{bit}}} \lfloor 2^{L_{\text{bit}}} V_{q,t,c} \rfloor$. The continuous coordinates are not published.

The names slot, sel, rec choose the slot, selected follower, and DATA recorder source. The tuple_r and pilot_r coordinates choose the injective arm tuple and DATA pilots; fast chooses the fast arm, while thin and prop supply their binary tests. The $K + 1$ commitment coordinates supply Algorithm 9.

For every $(q, t, \sigma, i) \in [M] \times [T] \times \{\text{DOWN}, \text{UP}\} \times [d]$, independently sample

$$(c_{q,t,\sigma,i}, b_{q,t,\sigma,i}) \sim \text{Unif}(\mathbb{F}_2^B \times \mathbb{F}_2),$$

independently of all scalar coordinates. The published finite row is

$$R_q := ((V_{q,t,c}^{(L)})_{t,c}, (c_{q,t,\sigma,i}, b_{q,t,\sigma,i})_{t,\sigma,i}).$$

Rows are independent across q and independent of the masks and bootstrap arrays. On a selected DOWN or UP channel, the affine hash is $u \mapsto c_{q,t,\sigma,i} \cdot u \oplus b_{q,t,\sigma,i}$.

B.5 Bootstrap decoding and outward protocol

Fix a block b and a receiver i . Let w_i be the column chosen by i for this block, and abbreviate $\mathcal{L}_i := \mathcal{L}_{i,b,w_i}$ and $b_i(t) := b_{i,b,w_i}(t)$. Let $A_{i,b} \subseteq I_b^+$ be the positions at which i follows its bootstrap array action. Its positive listening set is

$$P_{i,b} := \{t \in \mathcal{L}_i \cap A_{i,b} : Y_{i,t} > 0\},$$

where $Y_{i,t}$ is its observed reward. Before activation, $A_{i,b} = I_b^+$; after activation, it consists of the declared reserved positions. Write $P = P_{i,b}$ when i, b are fixed, and call $|P| \geq e_*$ the positive-observation quota.

The receiver knows the labels of its parent and children from the fixed tree, but a positive observation carries no sender label. To test one neighbor j , the receiver compares the same set P with every public column (j, b, w) , $w \in [S_0]$. Here j is a *candidate sender*: it selects an array family and the native alphabet $\mathcal{M}_j$; it is not inferred from an observation.

For this test abbreviate $\mathcal{L}_w := \mathcal{L}_{j,b,w}$, $a_{k,w,t} := a_{k,j,b,w,t}$, $r_w(t) := j_{j,b,w,t}$, and $z_{w,t} := z_{j,b,w,t}$. The index $r_w(t) \in [H_{\text{arr}}]$ selects a row of G , whereas j labels the candidate sender. If $t \in P \setminus \mathcal{L}_w$ and $b_i(t) = a_{k,w,t}$ for $k \in \{0, 1\}$, a candidate address ν can explain the positive observation only if its candidate arm avoids $b_i(t)$. By (83), this condition is the exclusion equation

$$G_{r_w(t)} \cdot \nu = (1 - k) \oplus z_{w,t}. \quad (84)$$

The two stored arms are distinct, so k is unique whenever such an equation exists. Coverage requires that each row $h \in [H_{\text{arr}}]$ occurs in at least

$$\lceil |P| / (64KH_{\text{arr}}) \rceil \quad (85)$$

exclusion equations. A solution ν also has to pass

$$\sum_{t \in P \cap \mathcal{L}_w} \mathbf{1}\{c_{j,b,w,t}(\nu) = b_i(t)\} \leq (64\alpha_0/K)|P|. \quad (86)$$

Algorithm 11 performs this test separately for each possible column of the specified candidate sender. Its output may contain several pairs and does not assert which player caused any individual positive observation.

Algorithm 11 Decode the public columns of candidate sender j

Require: Complete block b , receiver i , candidate neighbor j , $P_{i,b}$, and receiver listening arms $(b_i(t))_{t \in P_{i,b}}$.

```

1:  $D_{i \leftarrow j, b} \leftarrow \emptyset$ .
2: if  $|P_{i,b}| < e_*$  then
3:   return  $D_{i \leftarrow j, b}$ .
4: end if
5: for each public column  $w \in [S_0]$  of player  $j$  in block  $b$  do
6:   For each  $t \in P_{i,b} \setminus \mathcal{L}_{j,b,w}$  with  $b_i(t) = a_{k,w,t}$  for some  $k \in \{0, 1\}$ , add equation (84).
7:   if each row  $h \in [H_{\text{arr}}]$  occurs at least  $\lceil |P_{i,b}|/(64KH_{\text{arr}}) \rceil$  times then
8:     Solve the equations over  $\mathbb{F}_2$ .
9:     if they have a unique solution  $\nu$  and  $\nu$  satisfies (86) then
10:       $\mu \leftarrow \pi_{j,b,w}(\nu)$ .
11:      if  $\mu \in \mathcal{M}_j$  then
12:        Append  $(w, \mu)$  to  $D_{i \leftarrow j, b}$ .
13:      end if
14:    end if
15:  end if
16: end for
17: return  $D_{i \leftarrow j, b}$  in increasing column order.
```

Define the set of retained native labels by

$$\text{Ret}_{i \leftarrow j, b} := \{\mu \in \mathcal{M}_j : (w, \mu) \in D_{i \leftarrow j, b} \text{ for some } w \in [S_0]\}.$$

The actual message of player j in block b is denoted $\mu_{j,b}$. The decoder does not receive $\mu_{j,b}$; its possible retention is analyzed below.

Player i stores two digits $\widehat{J}_{i,0}, \widehat{J}_{i,1}$ and, for each child j , a key list $\mathcal{H}_{ij}$. The root starts with digits (J_0, J_1) ; every other digit and every child list starts at $\perp$. Each is filled at most once. The row declared by i is

$$\widehat{J}_i = \begin{cases} 1 + \widehat{J}_{i,0} + A\widehat{J}_{i,1}, & \widehat{J}_{i,0}, \widehat{J}_{i,1} \neq \perp, 1 + \widehat{J}_{i,0} + A\widehat{J}_{i,1} \in [M], \\ \perp, & \text{otherwise.} \end{cases} \quad (87)$$

Thus i is informed exactly when $\widehat{J}_i \neq \perp$. It parses a completed block if $\widehat{J}_i = \perp$ or some child list is $\perp$. Since $d_0 \geq 1$ and $\widehat{J}_0 = J$, the root never sends DISC.

Algorithm 12 constructs one message per block. At its end, the receiver calls Algorithm 11 once for each neighbor whose message is needed, always using its own $P_{i,b}$. The child's DISC field supplies a possible key; a parent's ROOT label supplies a digit only when its destination and echo equal the receiver's child index and private key.

Algorithm 12 Outward bootstrap at player i

```
1: for each block  $b$  in calendar order do
2:   if  $\widehat{J}_i = \perp$  or  $d_i = 0$  then
3:      $\mu \leftarrow \text{DISC}(U_i)$ .
4:   else
5:     Draw  $c \sim \text{Unif}(\{0, \dots, d_i - 1\})$  and  $k \sim \text{Unif}(\{0, 1\})$  independently; set  $j \leftarrow 2i + 1 + c$ .
6:     Set  $e \leftarrow \perp$  if  $\mathcal{H}_{ij} = \perp$ ; otherwise draw  $e \sim \text{Unif}(\mathcal{H}_{ij})$ .
7:      $\mu \leftarrow \text{ROOT}(c, k, \widehat{J}_{i,k}, e)$ .
8:   end if
9:   Draw a fresh  $w_i \sim \text{Unif}([S_0])$  independently; set  $v \leftarrow \pi_{i,b,w_i}^{-1}(\mu)$ .
10:  At each position  $t \in I_b^+$ , play  $d_{i,b,w_i,t}(v)$  while inactive and on declared reserved rounds while active;
    otherwise use the learning protocol.
11:  Form  $P_{i,b}$  from positive observations at positions  $t \in \mathcal{L}_{i,b,w_i}$  where the bootstrap arm was played.
12:  if block  $b$  is complete and parsing is required then
13:    for each child  $j$  with  $\mathcal{H}_{ij} = \perp$  do
14:      Run Algorithm 11 for  $(i, j, b)$  and set  $H \leftarrow \{u \in [A] : \text{DISC}(u) \in \text{Ret}_{i \leftarrow j, b}\}$ .
15:      if  $H \neq \emptyset$  then
16:         $\mathcal{H}_{ij} \leftarrow H$ .
17:      end if
18:    end for
19:    if  $i \neq 0$  and some  $\widehat{J}_{i,k} = \perp$  then
20:      Set  $p_i \leftarrow \lfloor (i-1)/2 \rfloor$  and  $c_i \leftarrow i - (2p_i + 1)$ ; run the decoder for  $(i, p_i, b)$ .
21:      for  $k \in \{0, 1\}$  with  $\widehat{J}_{i,k} = \perp$  do
22:         $V_{i,k} \leftarrow \{v \in \{0, \dots, A-1\} : \text{ROOT}(c_i, k, v, U_i) \in \text{Ret}_{i \leftarrow p_i, b}\}$ .
23:        if  $V_{i,k} \neq \emptyset$  then
24:           $\widehat{J}_{i,k} \leftarrow \min V_{i,k}$ .
25:        end if
26:      end for
27:    end if
28:  end if
29:  At a complete boundary, apply Algorithm 1 after these updates.
30: end for
```

B.6 Uniform array properties

Set

$$\kappa_{\text{arr}} := 2^{18}, \quad h_{\text{exc}} := \frac{e_*}{128\kappa_{\text{arr}}K}.$$

The constant κ_{arr} separates the two classes of raw candidates below, while h_{exc} is the allowed number of exceptional positions in the uniform array properties. Recall that ϵ_{arr} from (12) is the allowed exceptional-column fraction. For a complete block I_b , fix a support $\chi \in \{0, 1\}^{I_b \times [K]}$ and set $G_\chi := \sum_{t \in I_b} \sum_{a \in [K]} \chi_t(a)$. Fix arbitrary receiver listening choices $(\mathcal{L}_i, b_i)$, with $\mathcal{L}_i \subseteq I_b$ and $b_i : I_b \rightarrow [K]$, and put

$$E := \{t \in \mathcal{L}_i : \chi_t(b_i(t)) = 1\}, \quad s := |E|.$$

Call a pair $y = (v, w)$ a raw entry and abbreviate $c_y(t) := c_{v,w}(t)$ and $d_y(t) := d_{v,w}(t)$. For a tuple $x = (x_j)_{j \neq i}$ of other players' raw entries, let $d_{x_j}(t) = d_{j,b,w_j,t}(v_j)$ when $x_j = (v_j, w_j)$, and define

$$P_x := \{t \in E : d_{x_j}(t) \neq b_i(t) \text{ for all } j \neq i\}. \quad (88)$$

For a raw entry $y = (v, w)$ put $I_y := \{t \in E \setminus \mathcal{L}_w : c_y(t) = b_i(t)\}$ and $J_y := \{t \in E \cap \mathcal{L}_w : c_y(t) = b_i(t)\}$. An entry is class 1 if $\max\{|I_y|, |J_y|\} < s/(\kappa_{\text{arr}}K)$, and class 2 otherwise. These classes depend on raw arrays and receiver choices, not on transmitted labels or private history. Recall that, for the positive listening set P_x , quota means $|P_x| \geq e_*$. Recall also that a native label is retained when it is returned by Algorithm 11; its class is the class of the raw entry that produced it.

Before stating the uniform properties, recall the array-scale parameters introduced in Section 4:

$$d = m - 1, \quad L = \lceil C_0 K b_N \rceil, \quad e_* = \lceil C_0^4 K^2 b_N^2 \rceil, \quad \ell_{\text{col}} = \lceil L/K \rceil,$$

$$\epsilon_{\text{arr}} = c_\epsilon \frac{K^3 b_N^2}{m^3 T}, \quad S_0 = S_{\text{col}}.$$

Here L is the candidate-list scale, e_* is the required number of positive listening observations before raw decoding, ℓ_{col} is the disjoint-column scale used in the array arguments, ϵ_{arr} is the allowed exceptional-column fraction, and S_0 is the number of public array columns. Recall also that $\alpha_0 = 2^{-40}$ is the listening probability and that n is the bootstrap block length from (7).

For a complete block b and a positive set $P \subseteq I_b$, write $D_{i \leftarrow j, b}(P)$ for the output of Algorithm 11 when receiver i tests player j using P . For (A5) and (A8), fix a complete block b , players $i \neq j$, a support χ , receiver choices $(\mathcal{L}_i, b_i)$, and a message $\mu_j \in \mathcal{M}_j$. For each target column w , put

$$y_w := (\pi_{j, b, w}^{-1}(\mu_j), w), \quad P_w(x_{-i, -j}) := P_x \quad \text{with } x_j = y_w,$$

where $x_{-i, -j} = (x_k)_{k \notin \{i, j\}}$ consists of fixed raw entries. Write $\text{Cov}_{i, j, b, w}(P)$ if the exclusion equations for column (j, b, w) satisfy (85).

Lemma 29 (Uniform sampled arm arrays). *On the construction branch (11), fix any reserved mask matrix $(C_{q, t})$, with probability at least $1 - C_{\text{arr}} N^{-40}$ the code is certified and the following hold simultaneously for every complete block and all specified inputs. Here C_{arr} is universal; the probability is over the remaining public-construction randomness.*

(A1) *For any L raw entries from distinct columns of one player, outside at most h_{exc} positions every arm occurs among their $c_y(t)$ at least $L/(2K)$ times. We call this property A1'.*

(A2) *Fix a receiver and ℓ_{col} tuples, each containing one raw entry of every other player. If each player's columns are distinct across the tuples, then outside one set of at most h_{exc} positions, for every arm a at least $\ell_{\text{col}}/16$ tuples have all their played arms different from a .*

(A3) *For every player s , native label $\mu \in \mathcal{M}_s$, set $E' \subseteq I_b$ with $|E'| \geq e_*$, and map $\beta : I_b \rightarrow [K]$,*

$$\left| \left\{ w \in [S_0] : \sum_{t \in E' \cap \mathcal{L}_{s, b, w}} \mathbf{1}\{c_{s, b, w, t}(\pi_{s, b, w}^{-1}(\mu)) = \beta(t)\} > \frac{2\alpha_0 |E'|}{K} \right\} \right| \leq \epsilon_{\text{arr}} S_0.$$

(A4) *For every support $\chi \in \{0, 1\}^{I_b \times [K]}$ with $G_\chi \geq 2Ke_*/\alpha_0$ and every receiver i ,*

$$\left| \left\{ w \in [S_0] : \sum_{t \in \mathcal{L}_{i, b, w}} \chi_t(b_{i, b, w}(t)) < \frac{\alpha_0 G_\chi}{2K} \right\} \right| \leq \epsilon_{\text{arr}} S_0.$$

(A5) *Fix the common data above with $d_j \geq 1$, messages $(\mu_k)_{k \notin \{i, j\}}$, a child destination $c_{\text{dest}} \in \{0, \dots, d_j - 1\}$, and an echo $u \in [A]$. Conditional on the public table, if every nonreceiver player chooses an independent uniform column and transmits its fixed message, the probability that quota holds and the decoder retains an incorrect class-2 ROOT label with destination c_{dest} and echo u is at most ϵ_{arr} .*

(A6) For every player, position, and arm a , the fraction of columns listening on a lies in $[\alpha_0/(2K), 2\alpha_0/K]$. For each native message, the fraction whose played arm is a is at most $(1 + 1/m)/K$.

(A7) For every mask row $q \in [M]$, player s , and column $w \in [S_0]$, set $\mathcal{R}_{q,b} := \{t \in I_b : C_{q,t} = 1\}$. Then

$$|\mathcal{L}_{s,b,w} \cap \mathcal{R}_{q,b}| \leq 2\alpha_0 |\mathcal{R}_{q,b}| + C_0 \Lambda.$$

(A8) Fix the common data above and $x_{-i,-j}$. At most $\epsilon_{\text{arr}} S_0$ target columns w satisfy $|P_w(x_{-i,-j})| \geq e_*$ but fail $\text{Cov}_{i,j,b,w}(P_w(x_{-i,-j}))$.

B.7 List bounds, authentication and recovery

The results of this subsection are stated for the all-bootstrap auxiliary execution with exact-uniform private draws. In Proposition 3 we couple this execution to the physical initialization. Since the two executions agree until the first activation, and the support bound below implies that recovery occurs before any activation on the good event, the recovery guarantees transfer to the physical initialization.

Recall that h_{exc} is the allowed exceptional-position count and ϵ_{arr} is the exceptional-column fraction. For the fixed support and receiver listening choices used above, recall

$$E = \{t \in \mathcal{L}_i : \chi_t(b_i(t)) = 1\}, \quad s = |E|,$$

and, for a tuple x of the other senders' raw entries, $P_x \subseteq E$ is the resulting positive listening set defined in (88).

Lemma 30 (Candidate retention). *For a fixed sender, receiver, and block, let C be the set of raw entries (v, w) that pass the coverage, unique-solution, and overlap tests of Algorithm 11. Then $|C| < L$.*

Let (μ^, w^*) be the sender's actual native message and column, and let $v^* = \pi_{w^*}^{-1}(\mu^*)$, $y^* = (v^*, w^*)$. Conditional on the precolumn history, private keys, receiver column, and current messages,*

$$\Pr(|P_x| \geq e_* \text{ and } y^* \notin C) \leq 2\epsilon_{\text{arr}} + \frac{d\ell_{\text{col}}}{S_0}.$$

Proof. Suppose L candidates survive for positive set P_x , and denote them by $y_1, \dots, y_L$. The decoder gives at most one candidate per column. Property A1' therefore applies, and gives

$$\begin{aligned} \sum_{k=1}^L \sum_{t \in P_x} \mathbf{1}\{c_{y_k}(t) = b_i(t)\} &\geq \frac{L}{2K} (|P_x| - h_{\text{exc}}) \\ &\geq \frac{L|P_x|}{4K}. \end{aligned}$$

The second inequality uses $|P_x| \geq e_*$ and $h_{\text{exc}} \leq e_*/2$. Each surviving candidate satisfies the exclusion equations, so every counted match lies in its own listening set. Summing (86) consequently gives

$$\sum_{k=1}^L \sum_{t \in P_x} \mathbf{1}\{c_{y_k}(t) = b_i(t)\} \leq \frac{64\alpha_0 L |P_x|}{K}.$$

This contradicts $64\alpha_0 < 1/4$, and proves the list bound.

Fix the conditioning in the statement. Then E , s , and each sender's native message are fixed, while the d sender columns remain independent and uniform. If $s < e_*$, then $|P_x| \leq s < e_*$, so the event $\{|P_x| \geq e_*\}$

is impossible. Assume $s \geq e_*$ and call a column tuple deficient when $|P_x| < s/32$. If ℓ_{col} deficient tuples $x^{(1)}, \dots, x^{(\ell_{\text{col}})}$ used distinct columns within each sender, A2 would imply

$$\begin{aligned} \sum_{k=1}^{\ell_{\text{col}}} |P_{x^{(k)}}| &\geq \frac{\ell_{\text{col}}}{16} (s - h_{\text{exc}}) \\ &> \frac{\ell_{\text{col}} s}{32}, \end{aligned}$$

contrary to their deficiency. Choose a maximal family disjoint in every sender's column. It has fewer than ℓ_{col} members, and every deficient tuple shares some sender-column pair with this family. There are at most $d\ell_{\text{col}}$ such pairs; each has probability $1/S_0$. The union bound proves

$$\Pr\{|P_x| < s/32\} \leq d\ell_{\text{col}}/S_0. \quad (89)$$

The true address always satisfies the exclusion equations. Property A3 fails for its column with probability at most ϵ_{arr} . Outside that event and deficiency, its overlap count satisfies

$$\begin{aligned} \sum_{t \in P_x \cap \mathcal{L}_w} \mathbf{1}\{c_{\pi_w^{-1}(\mu), w}(t) = b_i(t)\} &\leq \frac{2\alpha_0 s}{K} \\ &\leq \frac{64\alpha_0 |P_x|}{K}. \end{aligned}$$

For coverage, first fix the other senders' actual columns. A8 bounds by ϵ_{arr} the conditional probability that $|P_x| \geq e_*$ while coverage fails for the target's independent uniform column. Averaging over the fixed columns preserves this bound. Under coverage, injectivity makes the true address the unique solution. Hence the event

$$\{|P_x| \geq e_*\} \cap \{y^\star \notin C\}$$

is contained in the union of the A3 exception, deficiency, and the A8 exception, whose total probability is at most $2\epsilon_{\text{arr}} + d\ell_{\text{col}}/S_0$. $\square$

For a nonroot player i , put

$$p(i) := \left\lfloor \frac{i-1}{2} \right\rfloor, \quad c_i^{\text{loc}} := i - (2p(i) + 1) \in \{0, 1\},$$

so that $p(i)$ is the parent of i and c_i^{loc} is the local child index used in a ROOT label sent from $p(i)$ to i . Define the false authenticated-ROOT event

$$F_{i,b} := \left\{ \begin{array}{l} \exists k \in \{0, 1\}, v \in \{0, \dots, A-1\} : \\ \text{ROOT}(c_i^{\text{loc}}, k, v, U_i) \in \text{Ret}_{i \leftarrow p(i), b}, \\ \text{ROOT}(c_i^{\text{loc}}, k, v, U_i) \neq \mu_{p(i), b} \end{array} \right\}. \quad (90)$$

Lemma 31 (Authentication of a root digit). *Fix a complete block b and a nonroot player i . Then*

$$\Pr(F_{i,b}) \leq \epsilon_{\text{arr}} + \frac{d\ell_{\text{col}}}{S_0} + \frac{L}{A}.$$

Proof. Fix b and i , and suppress these indices in the notation for the fixed receiver. Thus E is its potential positive listening set, $s = |E|$, and $P = P_{i,b}$ is its actual positive listening set. If $s < e_*$, then $|P| \leq s < e_*$, so Algorithm 11 returns the empty list and $F_{i,b}$ is impossible. Assume henceforth that $s \geq e_*$.

We first record two deterministic consequences of A1' and the code distance. There are fewer than L columns containing a class-1 raw address. Indeed, if $y_1, \dots, y_L$ were class-1 addresses in distinct columns, then their class definition and A1' would give

$$\frac{2Ls}{\kappa_{\text{arr}}K} > \sum_{q=1}^L (|I_{y_q}| + |J_{y_q}|) \geq \frac{L(s - h_{\text{exc}})}{2K} \geq \frac{Ls}{4K},$$

contradicting $\kappa_{\text{arr}} > 8$.

Next consider a column w for which there exists a set $P' \subseteq E$ with $|P'| \geq s/32$ such that the exclusion equations generated by P' satisfy the coverage condition (85). Call such a P' a *witness* for w . A witness column contains at most one class-1 raw address. To prove this, suppose that $y = (v, w)$ and $y' = (v', w)$ were distinct class-1 addresses in the same witness column. Lemma 28 gives at least $H_{\text{arr}}/8$ code rows on which Gv and Gv' differ. For each such row, coverage supplies at least $\lceil |P'|/(64KH_{\text{arr}}) \rceil$ usable occurrences. At every such occurrence exactly one of $c_y(t)$ and $c_{y'}(t)$ equals the receiver arm $b_i(t)$. Hence

$$|I_y| + |I_{y'}| \geq \frac{H_{\text{arr}}}{8} \left\lceil \frac{|P'|}{64KH_{\text{arr}}} \right\rceil \geq \frac{s}{16384K}.$$

On the other hand, since both addresses are class 1,

$$|I_y| + |I_{y'}| < \frac{2s}{\kappa_{\text{arr}}K} = \frac{s}{131072K},$$

a contradiction.

For each witness column w that contains a class-1 raw address, let (v_w, w) denote that unique address and put $\mu_w := \pi_w(v_w)$. Define

$$\mathcal{D}_{i,b} := \{e \in [A] : \exists w, k \in \{0, 1\}, v \in \{0, \dots, A-1\} \text{ with } \mu_w = \text{ROOT}(c_i^{\text{loc}}, k, v, e)\}.$$

Because fewer than L columns contain a class-1 address, $|\mathcal{D}_{i,b}| < L$. Moreover, $\mathcal{D}_{i,b}$ is determined by the public arrays, the fixed reward support, and player i 's fresh receiver column in block b ; that fresh column is independent of the private key U_i . Consequently

$$\Pr\{U_i \in \mathcal{D}_{i,b}\} = \mathbb{E} \left[\frac{|\mathcal{D}_{i,b}|}{A} \right] \leq \frac{L}{A}. \quad (91)$$

Let

$$\mathsf{D}_{i,b} := \{|P_{i,b}| < s/32\}$$

be the deficiency event. By (89),

$$\Pr(\mathsf{D}_{i,b}) \leq \frac{d\ell_{\text{col}}}{S_0}. \quad (92)$$

Suppose that $\mathsf{D}_{i,b}$ does not occur and that a false retained ROOT label contributing to $\mathsf{F}_{i,b}$ is produced by a class-1 raw address. Since the label is retained, its column satisfies coverage for the actual set $P_{i,b}$; since $|P_{i,b}| \geq s/32$, the set $P_{i,b}$ is a witness for that column. The retained label has echo U_i , so by the definition of $\mathcal{D}_{i,b}$ we must have $U_i \in \mathcal{D}_{i,b}$. Thus the class-1 part of $\mathsf{F}_{i,b}$ is contained in

$$\mathsf{D}_{i,b} \cup \{U_i \in \mathcal{D}_{i,b}\}.$$

Equations (91)–(92) therefore bound its probability by $d\ell_{\text{col}}/S_0 + L/A$.

It remains to control false retained ROOT labels produced by class-2 raw addresses. For each fixed value $u \in [A]$, apply A5 with destination c_i^{loc} and echo u . A retained label can occur only when $|P_{i,b}| \geq e_*$, so A5 gives probability at most ϵ_{arr} that an incorrect class-2 ROOT label with that destination and echo is retained. Averaging this bound over the uniform key U_i yields

$$\Pr\{F_{i,b} \text{ occurs through a class-2 raw address}\} \leq \epsilon_{\text{arr}}.$$

Combining the class-1 and class-2 cases proves the lemma. $\square$

Let $h_{\text{tree}} = \lfloor \log_2 m \rfloor$ be the tree height. Let

$$\mathcal{T} := \{(i, j) : j \in \{2i+1, 2i+2\}, j < m\}$$

be the set of parent-child edges, with the parent written first. For a complete block $b \in [B_{\text{blk}}]$, write $\chi^{(b)}$ for its support matrix and

$$G_b := \sum_{t,a} \chi_t^{(b)}(a).$$

Define the set of informative block indices by

$$\mathcal{I} := \left\{ b \in [B_{\text{blk}}] : G_b \geq \frac{128Ke_*}{\alpha_0} \right\}.$$

Write $R := |\mathcal{I}|$, and when $R \geq q$ let b_q denote the q th smallest element of $\mathcal{I}$. Recall from Subsection B.5 that $\hat{J}_{i,k}$ is player i 's stored estimate of digit J_k and $\mathcal{H}_{ij}$ is the child-key list stored by player i for child j . For $b = 0, 1, \dots, B_{\text{blk}}$, write $\hat{J}_{i,k}^{[b]}$, $\hat{J}_i^{[b]}$, and $\mathcal{H}_{ij}^{[b]}$ for the corresponding stored values immediately after the decoding/list updates of complete block b , with $b = 0$ denoting their initial values.

Proposition 32 (Outward recovery). *Fix $\delta_{\text{prop}} \in (0, 1)$ and set*

$$r_{\text{boot}} := 1 + \lceil 8L(2h_{\text{tree}} + \log(m/\delta_{\text{prop}})) \rceil.$$

There is an event $\mathcal{G}_{\text{rec}}$ such that

$$\Pr(\mathcal{G}_{\text{rec}}^c) \leq Cm^2 B_{\text{blk}} \left(\epsilon_{\text{arr}} + \frac{d\ell_{\text{col}}}{S_0} + \frac{L}{A} \right) + \delta_{\text{prop}}, \quad (93)$$

and on $\mathcal{G}_{\text{rec}}$ the following statements hold simultaneously:

(i) *For every $b \in \{0, \dots, B_{\text{blk}}\}$, every player i , and $k \in \{0, 1\}$,*

$$\hat{J}_{i,k}^{[b]} \neq \perp \implies \hat{J}_{i,k}^{[b]} = J_k.$$

(ii) *For every $b \in \{0, \dots, B_{\text{blk}}\}$ and every $(i, j) \in \mathcal{T}$,*

$$\mathcal{H}_{ij}^{[b]} \neq \perp \implies U_j \in \mathcal{H}_{ij}^{[b]} \text{ and } |\mathcal{H}_{ij}^{[b]}| < L.$$

(iii) *If $R \geq 1$, then every child list is initialized after the first informative block:*

$$\mathcal{H}_{ij}^{[b_1]} \neq \perp \quad \text{for every } (i, j) \in \mathcal{T}.$$

(iv) *If $R \geq r_{\text{boot}}$, then every player is informed after the r_{boot} th informative block:*

$$\hat{J}_i^{[b_{r_{\text{boot}}}] = J \quad \text{for every } i \in \{0, \dots, m-1\}.$$

Moreover, if $R < r_{\text{boot}}$, then the support in complete blocks obeys

$$\sum_{b=1}^{B_{\text{blk}}} G_b \leq \frac{128Ke_*T}{\alpha_0 n} + (r_{\text{boot}} - 1)Kn. \quad (94)$$

Proof. Let

$$\vec{\mathcal{T}} := \{(r, s) : (r, s) \in \mathcal{T} \text{ or } (s, r) \in \mathcal{T}\},$$

where the first coordinate is the receiver and the second is the sender. Define

$$\begin{aligned} \mathcal{B}_{\text{quota}} &:= \bigcup_{b \in \mathcal{I}} \bigcup_{i=0}^{m-1} \{|P_{i,b}| < e_*\}, \\ \mathcal{B}_{\text{rej}} &:= \bigcup_{b=1}^{B_{\text{blk}}} \bigcup_{(i,j) \in \vec{\mathcal{T}}} (\{|P_{i,b}| \geq e_*\} \cap \{\mu_{j,b} \notin \text{Ret}_{i \leftarrow j, b}\}), \\ \mathcal{B}_{\text{false}} &:= \bigcup_{b=1}^{B_{\text{blk}}} \bigcup_{i=1}^{m-1} F_{i,b}, \\ \mathcal{B} &:= \mathcal{B}_{\text{quota}} \cup \mathcal{B}_{\text{rej}} \cup \mathcal{B}_{\text{false}}. \end{aligned}$$

These three events are, respectively, failure to obtain the positive quota in an informative block, rejection of an actual neighboring message despite meeting the quota, and retention of a false authenticated ROOT label.

Fix $b \in \mathcal{I}$ and a receiver i . A4 gives

$$s_{i,b} \geq \frac{\alpha_0 G_b}{2K} \geq 64e_*$$

outside a set of receiver columns of probability at most ϵ_{arr} . Conditional on such a receiver column, (89) gives

$$|P_{i,b}| \geq s_{i,b}/32 \geq 2e_*$$

except with probability at most $d\ell_{\text{col}}/S_0$. Hence

$$\Pr\{|P_{i,b}| < e_*\} \leq \epsilon_{\text{arr}} + \frac{d\ell_{\text{col}}}{S_0}.$$

For each directed neighboring pair (i, j) and block b , Lemma 30 gives

$$\Pr(|P_{i,b}| \geq e_*, \mu_{j,b} \notin \text{Ret}_{i \leftarrow j, b}) \leq 2\epsilon_{\text{arr}} + \frac{d\ell_{\text{col}}}{S_0}.$$

Lemma 31 gives, for every nonroot i and block b ,

$$\Pr(F_{i,b}) \leq \epsilon_{\text{arr}} + \frac{d\ell_{\text{col}}}{S_0} + \frac{L}{A}.$$

There are at most mB_{blk} quota tests, m^2B_{blk} directed-neighbor rejection tests, and $(m-1)B_{\text{blk}}$ false-authentication tests. A union bound therefore gives

$$\Pr(\mathcal{B}) \leq Cm^2B_{\text{blk}} \left(\epsilon_{\text{arr}} + \frac{d\ell_{\text{col}}}{S_0} + \frac{L}{A} \right). \quad (95)$$

We next prove the first three conclusions on $\mathcal{B}^c$. For $b = 0, 1, \dots, B_{\text{blk}}$ let $\text{Inv}(b)$ be the conjunction of

$$\widehat{J}_{i,k}^{[b]} \neq \perp \implies \widehat{J}_{i,k}^{[b]} = J_k \quad (0 \leq i < m, k \in \{0, 1\}), \quad (96)$$

$$\mathcal{H}_{ij}^{[b]} \neq \perp \implies U_j \in \mathcal{H}_{ij}^{[b]}, \quad |\mathcal{H}_{ij}^{[b]}| < L \quad ((i, j) \in \mathcal{T}). \quad (97)$$

Initially, the root stores J_0, J_1 , every nonroot digit is $\perp$, and every child list is $\perp$, so $\text{Inv}(0)$ holds.

Assume $\text{Inv}(b-1)$ and consider the updates at the end of complete block b . First fix a nonroot player i and a digit $k \in \{0, 1\}$ that is $\perp$ before the update. If the digit remains $\perp$, there is nothing to prove. Otherwise Algorithm 12 stores a value v from a retained label

$$\text{ROOT}(c_i^{\text{loc}}, k, v, U_i) \in \text{Ret}_{i \leftarrow p(i), b}.$$

Since $\mathcal{B}_{\text{false}}$ does not occur, this retained label equals the actual parent message $\mu_{p(i), b}$. Hence the parent was informed when the block message was chosen and sent

$$\mu_{p(i), b} = \text{ROOT}(c_i^{\text{loc}}, k, \widehat{J}_{p(i), k}^{[b-1]}, U_i).$$

By (96), $\widehat{J}_{p(i), k}^{[b-1]} = J_k$, so the newly stored value is $v = J_k$. Thus (96) is preserved.

Now fix a parent-child edge $(i, j) \in \mathcal{T}$. If $\mathcal{H}_{ij}^{[b-1]} \neq \perp$, the list is immutable and (97) is preserved. Suppose instead that $\mathcal{H}_{ij}^{[b-1]} = \perp$. Since child lists are immutable, this list was $\perp$ in every earlier block. We claim that $\widehat{J}_j^{[b-1]} = \perp$. Otherwise player j must previously have stored at least one digit from a retained label $\text{ROOT}(c_j^{\text{loc}}, k, v, U_j)$. But while $\mathcal{H}_{ij} = \perp$, every actual message of parent i addressed to j has echo $\perp$; messages not addressed to j have a different local-child field, and an uninformed parent sends a DISC label. Thus such a retained authenticated ROOT label could not equal the actual parent message, so some event $F_{j, b'}$ would have occurred in an earlier block $b' < b$, contradicting $\mathcal{B}_{\text{false}}^c$. Hence $\widehat{J}_j^{[b-1]} = \perp$, and player j sends $\mu_{j, b} = \text{DISC}(U_j)$ in block b .

If $\mathcal{H}_{ij}$ remains $\perp$ after decoding block b , then (97) is vacuous. If it is initialized, the child-decoder output used by player i is nonempty, so Algorithm 11 necessarily has $|P_{i, b}| \geq e_*$. Since $\mathcal{B}_{\text{rej}}$ does not occur, the actual child message $\text{DISC}(U_j)$ belongs to $\text{Ret}_{i \leftarrow j, b}$. Therefore the newly stored list contains U_j . Lemma 30 gives fewer than L raw candidates and hence fewer than L retained native labels, so the stored DISC-key list has size below L . This proves (97) and completes the induction.

If $R \geq 1$, consider the first informative block b_1 . For any edge (i, j) whose child list is still $\perp$ at the start of this block, the preceding argument shows that j sends $\text{DISC}(U_j)$. Since $\mathcal{B}_{\text{quota}}$ does not occur, $|P_{i, b_1}| \geq e_*$, and since $\mathcal{B}_{\text{rej}}$ does not occur, this actual DISC label is retained. Thus every such list is initialized in block b_1 . This proves conclusions (i)–(iii) on $\mathcal{B}^c$.

It remains to prove the finite informative-block recovery time. Fix a root-to-node path

$$v_0 = 0, v_1, \dots, v_h, \quad h \leq h_{\text{tree}}.$$

Assume $R \geq r_{\text{boot}}$. The first informative block b_1 has already initialized all child lists. For $q = 1, \dots, r_{\text{boot}} - 1$, regard b_{q+1} as trial q , and let $\mathcal{F}_{q-1}$ contain all private choices revealed before the message choices of that block.

On a history on which no event in $\mathcal{B}$ has yet occurred and the path is not yet fully informed, let v_{ℓ_q} be the first uninformed vertex on the path and choose deterministically one missing digit $k_q \in \{0, 1\}$ of that vertex. Its parent v_{ℓ_q-1} is informed. By conclusions (i)–(iii), its list for child v_{ℓ_q} contains $U_{v_{\ell_q}}$ and has size below L . Let Z_q be the indicator that in block b_{q+1} this parent chooses that child, digit k_q , and echo $U_{v_{\ell_q}}$. Conditional on $\mathcal{F}_{q-1}$,

$$\Pr(Z_q = 1 \mid \mathcal{F}_{q-1}) \geq \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{L} =: p_0 = \frac{1}{4L}.$$

If a bad event in $\mathcal{B}$ has already occurred or the path has already become fully informed, extend the probability space by defining Z_q to be an independent Bernoulli(p_0) variable. With this extension, for every q ,

$$\mathbb{E}[e^{-Z_q} \mid \mathcal{F}_{q-1}] \leq 1 - (1 - e^{-1})p_0,$$

and iteration gives

$$\mathbb{E} \exp \left(- \sum_{q=1}^r Z_q \right) \leq \exp \{ -r(1 - e^{-1})p_0 \} \quad (r \geq 1). \quad (98)$$

On $\mathcal{B}^c$, whenever $Z_q = 1$, the actual parent message is

$$\text{ROOT}(c_{v_{\ell_q}}^{\text{loc}}, k_q, J_{k_q}, U_{v_{\ell_q}}).$$

Because block b_{q+1} is informative, $\mathcal{B}_{\text{quota}}^c$ gives quota at the child, and $\mathcal{B}_{\text{rej}}^c$ ensures that the actual message is retained. Finally $\mathcal{B}_{\text{false}}^c$ ensures that no different retained ROOT label with the same destination and authentication key can be chosen instead. Hence player v_{ℓ_q} stores the previously missing digit J_{k_q} . Thus each success $Z_q = 1$ delivers one new digit on the path. Since at most two digits are needed on each of the h edges, the event that $\mathcal{B}^c$ holds and the path is still unfinished after r trials implies

$$\sum_{q=1}^r Z_q < 2h \leq 2h_{\text{tree}}.$$

By Markov's inequality and (98),

$$\begin{aligned} & \Pr\{\mathcal{B}^c \text{ and the path is unfinished after } r \text{ trials}\} \\ & \leq \Pr \left\{ e^{-\sum_{q=1}^r Z_q} > e^{-2h_{\text{tree}}} \right\} \leq \exp \{ 2h_{\text{tree}} - r(1 - e^{-1})p_0 \}. \end{aligned}$$

For $r = r_{\text{boot}} - 1$,

$$r(1 - e^{-1})p_0 \geq 2(1 - e^{-1}) \left(2h_{\text{tree}} + \log \frac{m}{\delta_{\text{prop}}} \right) \geq 2h_{\text{tree}} + \log \frac{m}{\delta_{\text{prop}}},$$

because $2(1 - e^{-1}) > 1$. Hence the preceding probability is at most δ_{prop}/m . A union bound over the at most m root-to-node paths gives

$$\Pr \left(\mathcal{B}^c \cap \{R \geq r_{\text{boot}}\} \cap \{\exists i : \hat{J}_i^{\lceil b_{r_{\text{boot}}} \rceil} \neq J\} \right) \leq \delta_{\text{prop}}. \quad (99)$$

Combining (95) and (99), and letting $\mathcal{G}_{\text{rec}}$ be the event on which conclusions (i)–(iv) hold, proves (93).

Finally suppose $R < r_{\text{boot}}$. Each informative complete block has support at most Kn , so their total support is at most $(r_{\text{boot}} - 1)Kn$. Every noninformative complete block satisfies $G_b < 128Ke_*/\alpha_0$, and there are at most $B_{\text{blk}} \leq T/n$ such blocks. Therefore

$$\sum_{b=1}^{B_{\text{blk}}} G_b \leq (r_{\text{boot}} - 1)Kn + \frac{128Ke_*T}{\alpha_0 n},$$

which is (94). □

B.8 Reserved masks and bootstrap clocks

For a support matrix $\chi \in \{0, 1\}^{T \times K}$, write $g_t = \sum_a \chi_t(a)$ and $G(t) = \sum_{u \leq t} g_u$.

Lemma 33 (Uniform reserved masks). *With probability at least $1 - CN^{-40}$, the sampled masks satisfy both properties below.*

- (i) *Every row has at most $2pT + C_0\Lambda$ reserved positions, and every complete block in every row has at most $2pn + C_0\Lambda$.*
- (ii) *For every support matrix χ , all but at most M/N rows satisfy, at every complete-block boundary t ,*

$$\frac{1}{2}pG(t) - KC_0\Lambda \leq \sum_{u \leq t} C_{j,u}g_u \leq 2pG(t) + KC_0\Lambda. \quad (100)$$

Proof. Fix χ , a row j , and a complete-block boundary t . Put $X = \sum_{u \leq t} C_{j,u}g_u/K$ and $\mu = pG(t)/K$. For $a \in [0, 1]$, convexity gives $e^a - 1 \leq (e - 1)a$ and $e^{-a} - 1 \leq -(1 - e^{-1})a$. Independence implies

$$\begin{aligned} \mathbb{E}e^X &= \prod_{u \leq t} (1 + p(e^{g_u/K} - 1)) \leq e^{(e-1)\mu}, \\ \mathbb{E}e^{-X} &= \prod_{u \leq t} (1 + p(e^{-g_u/K} - 1)) \leq e^{-(1-e^{-1})\mu}. \end{aligned}$$

Since $e - 1 < 2$ and $1 - e^{-1} > 1/2$, Markov's inequality yields

$$\begin{aligned} \Pr\{X > 2\mu + C_0\Lambda\} &\leq e^{-C_0\Lambda}, \\ \Pr\{X < \mu/2 - C_0\Lambda\} &\leq e^{-C_0\Lambda}. \end{aligned}$$

Let a row be bad for χ if either inequality fails at some complete-block boundary. There are at most T such boundaries, so its probability is at most $2Te^{-C_0\Lambda} \leq N^{-30}$. The bad-row indicators are independent across j . As $M = N^2$,

$$\Pr\{\text{more than } M/N \text{ rows fail}\} \leq (eN^{-29})^N.$$

There are 2^{KT} support matrices and $KT \leq N/4$. Consequently the probability that (ii) fails is at most

$$\exp\{KT \log 2 + N(1 - 29 \log N)\} \leq N^{-40}.$$

For (i), apply the upper exponential-moment bound with every weight equal to one, first on $[T]$ and then on each complete block. Each count exceeds twice its mean plus $C_0\Lambda$ with probability at most $e^{-C_0\Lambda}$. The union over at most $M(T + 1)$ counts is at most N^{-40} after increasing C_0 . This proves both claims. $\square$

Lemma 34 (Bootstrap clock concentration). *Fix a deterministic reward matrix, the players' private keys, and the root index J . Assume that row J satisfies both reserved-mask properties of Lemma 33 for the reward support in (5), and use the public arrays of Lemma 29. In each block, choose the bootstrap messages before drawing fresh columns independently and uniformly from $[S_{\text{col}}]$, conditional on the preceding history. All players use the corresponding communication sequences on reserved rounds; their actions on other rounds may be arbitrary.*

Let $\mathcal{L}_{i,b} \subseteq I_b$ be player i 's listening positions in block b , expressed in calendar time. Write $C_t = C_{J,t}$ for the selected row's reserved indicator and $G_C(t) = \sum_{u \leq t} C_u g_u$ for its cumulative reserved support. Define the true-row counts

$$\begin{aligned} X_{i,b} &= \sum_{t \in I_b} \mathbf{1}\{t \in \mathcal{L}_{i,b}\} C_t \mathbf{1}\{Y_{i,t} > 0\}, \\ C_i(t) &= \frac{K}{\alpha_0 p} \sum_{b: bn \leq t} X_{i,b}. \end{aligned}$$

These counts are defined mathematically even before a player recovers the true table row. Recall the activation threshold Q_{boot} from (9). With probability at least $1 - CmTe^{-C_0\Lambda}$, simultaneously for every player and complete-block boundary t ,

$$\frac{G(t)}{8e} - \frac{Q_{\text{boot}}}{2} \leq C_i(t) \leq 8G(t) + \frac{Q_{\text{boot}}}{2}. \quad (101)$$

Here C is universal.

Proof. Let $\mathcal{F}_{b-1}$ contain the public arrays, fixed private keys, root index, and all choices before block b 's fresh columns. In particular, each player's block message is $\mathcal{F}_{b-1}$ -measurable. For $t \in I_b$ and $a \in [K]$ with $C_t = 1$ and $r_t(a) > 0$, property (A6) gives probability between $\alpha_0/(2K)$ and $2\alpha_0/K$ that player i listens on arm a . For each other player, the probability of playing that arm is at most $(1 + 1/m)/K \leq 1/m$, since $K \geq m + 1$. Let $L_{i,t,a}$ be the event that player i listens on arm a at t , and let $A_{j,t}$ denote player j 's chosen arm. Conditional independence of columns across players gives the exact factorization

$$\Pr\left(L_{i,t,a} \cap \bigcap_{j \neq i} \{A_{j,t} \neq a\} \mid \mathcal{F}_{b-1}\right) = \Pr(L_{i,t,a} \mid \mathcal{F}_{b-1}) \prod_{j \neq i} \Pr(A_{j,t} \neq a \mid \mathcal{F}_{b-1}).$$

Moreover,

$$\Pr\{\text{all other players avoid } a \mid \mathcal{F}_{b-1}\} \geq (1 - 1/m)^{m-1} \geq e^{-1}.$$

The last inequality follows from $\log(1 - x) \geq -x/(1 - x)$ at $x = 1/m$. The event in the factorization is precisely a positive listening observation on arm a . Put $G_{C,b} = \sum_{t \in I_b} C_t g_t$. Summing over supported reserved pairs gives

$$\frac{\alpha_0}{2eK} G_{C,b} \leq \mathbb{E}[X_{i,b} \mid \mathcal{F}_{b-1}] \leq \frac{2\alpha_0}{K} G_{C,b}. \quad (102)$$

Property (A7) and Lemma 33(i) give the pathwise increment bound

$$0 \leq X_{i,b} \leq 2\alpha_0(2pn + C_0\Lambda) + C_0\Lambda \leq 4\alpha_0pn + 3C_0\Lambda =: b_*.$$

For each player and complete-block prefix, the exponential-supermartingale Chernoff bound of Howard, Ramdas, McAuliffe, and Sekhon [HRMS20], applied with (102) and $0 \leq X_{i,b} \leq b_*$, gives, with failure probability at most $2e^{-C_0\Lambda}$,

$$\frac{\alpha_0}{4eK} G_C(t) - b_*C_0\Lambda \leq \sum_{b:bn \leq t} X_{i,b} \leq \frac{4\alpha_0}{K} G_C(t) + b_*C_0\Lambda.$$

A union bound over at most mT player-boundary pairs makes these inequalities simultaneous. Multiply by $K/(\alpha_0 p)$ and use (100). The resulting slopes are $1/(8e)$ and 8 . Each additive error is bounded by

$$\frac{4C_0K\Lambda}{p} + 4C_0Kn\Lambda + \frac{3C_0^2K\Lambda^2}{\alpha_0 p}.$$

To compare with the threshold, use $n \leq 2C_0^3\sqrt{KT}$, $\Lambda \leq b_N$, and $\widehat{V} \geq K^{5/2}\sqrt{T}\Lambda^2$. Then the preceding error is at most

$$8C_0^4\widehat{V} + \left(4C_0 + \frac{3C_0^2}{\alpha_0}\right) \frac{Kb_N^2}{p}.$$

Choose the fixed universal C_0 so that $8C_0^4 \leq C_0^8/2$ and $4C_0 + 3C_0^2/\alpha_0 \leq C_0^8/2$. The error is then at most $Q_{\text{boot}}/2$, which proves (101). $\square$

B.9 Proof of the public-array lemma

Proof of Lemma 29. Condition first on an arbitrary certified output G of Lemma 28. All estimates below are uniform in these choices. At the end we add the code-certification failure probability.

Parameter bounds. On the construction branch, enlarging C_b gives

$$T > C_b^2 K^3 \Lambda^4, \quad n \leq T,$$

$$0 < \epsilon_{\text{arr}} < 1, \quad \Gamma \asymp Kn.$$

Moreover $n \leq 2C_0^3 \sqrt{KT}$ and $L \leq 2C_0 K b_N$. By the definitions of A and ϵ_{arr} , choosing $C_A \geq 32eC_0/c_\epsilon$ yields

$$\frac{\epsilon_{\text{arr}} A}{2eL} \geq \frac{c_\epsilon C_A}{2eC_0} \frac{K^3 b_N}{m^2} \frac{T+1}{T} \geq 4K\Lambda.$$

We shall also use

$$\begin{aligned} \frac{L}{A} &\leq \frac{\epsilon_{\text{arr}}}{8e}, \quad \epsilon_{\text{arr}} \geq \frac{c_\epsilon}{N}. \\ \epsilon_{\text{arr}} S_0 &\geq C_S d \ell_{\text{col}} \Gamma. \end{aligned}$$

The first term in (12) dominates the second on this branch, so $S_0 = O(N^{3/2})$. The definitions give $A = O(N)$ and $Q_{\text{arr}} = O(N^2)$; hence every alphabet and column-count logarithm is $O(\Lambda)$.

A1' and A2. Fix one family of entries and one position t . For A1', distinct columns give independent uniform candidate arms. For A2, the tuples are independent at t , and within each tuple the d played arms are independent and uniform. Since $d \leq K-2$,

$$(1 - 1/K)^d \geq e^{-d/(K-1)} \geq e^{-1}.$$

Chernoff's inequality [BLM13], followed by a union over $a \in [K]$, bounds the respective probabilities that t is bad by

$$q_1 := K e^{-L/(8K)}, \quad q_2 := K e^{-\ell_{\text{col}}/(8e)}.$$

For either property these bad-position indicators are independent over t . Since $L/K, \ell_{\text{col}} \geq C_0 b_N$, a sufficiently large C_0 makes $q := \max(q_1, q_2) \leq h_{\text{exc}}/(ne^2)$. If $h_{\text{exc}} \leq n$, the binomial upper tail gives

$$\Pr\{\text{more than } h_{\text{exc}} \text{ bad positions}\} \leq (enq/h_{\text{exc}})^{h_{\text{exc}}} \leq e^{-h_{\text{exc}}}.$$

For $h_{\text{exc}} > n$ both properties are automatic. The logarithm of the number of players, blocks, and admissible families is

$$O(L \log(Q_{\text{arr}} S_0) + \ell_{\text{col}} m \log(Q_{\text{arr}} S_0) + \Lambda) = O(C_0 K \Lambda^2).$$

Because $h_{\text{exc}} \geq C_0^4 K \Lambda^2 / (128 \kappa_{\text{arr}})$, the union bound makes the failure probability for A1'–A2 at most N^{-40} .

A3 and A4. For fixed (μ, E', β) , the count in A3 has law $\text{Bin}(|E'|, \alpha_0/K)$, even when the actual address of μ is fixed. Its upper-tail probability is at most $e^{-c \alpha_0 e^*/K}$. For A4, the count in one receiver column is a sum of independent indicators with mean $\alpha_0 G_X/K \geq 2e_*$; its lower-tail probability is at most $e^{-c e^*}$. Both bounds are at most N^{-30} for large C_0 .

For either fixed input, columns are independent. If X counts their failures, binomial domination and $\epsilon_{\text{arr}} \geq c_\epsilon/N$ give

$$\Pr\{X > \epsilon_{\text{arr}} S_0\} \leq \left(\frac{eN^{-30}}{\epsilon_{\text{arr}}} \right)^{\epsilon_{\text{arr}} S_0} \leq e^{-c \epsilon_{\text{arr}} S_0 \Lambda}. \quad (103)$$

The A3 inputs number at most $\text{poly}(N) 2^n K^n Q_{\text{arr}}$ and the A4 inputs at most $\text{poly}(N) 2^{Kn}$. Their logarithms are $O(\Gamma)$. Thus $\epsilon_{\text{arr}} S_0 \geq C_S d \ell_{\text{col}} \Gamma$ makes their combined failure at most $C N^{-40}$.

A5: raw candidates. Work on A1'–A2. Fix the data in A5, all nontarget permutations, and the target's actual address $\pi_{j,b,w}^{-1}(\mu_j)$ in each column w . For a tuple $\mathbf{w} = (w_k)_{k \neq i} \in [S_0]^d$, put $x_k(\mathbf{w}) := (\pi_{k,b,w_k}^{-1}(\mu_k), w_k)$ for $k \notin \{i, j\}$ and $P_{\mathbf{w}} := P_{w_j}((x_k(\mathbf{w}))_{k \notin \{i, j\}})$. Let $C_{\mathbf{w}}$ contain the target's raw entries passing quota, exclusion, coverage, uniqueness, and overlap for $P_{\mathbf{w}}$. These tests do not use the target's remaining permutation labels.

Each column contributes at most one entry to $C_{\mathbf{w}}$. If it contained L entries $y_1, \dots, y_L$, then A1' and $h_{\text{exc}} \leq e_*/2 \leq |P_{\mathbf{w}}|/2$ would give

$$\sum_{k=1}^L \sum_{t \in P_{\mathbf{w}}} \mathbf{1}_{\{c_{y_k}(t) = b_i(t)\}} \geq \frac{L}{2K}(|P_{\mathbf{w}}| - h_{\text{exc}}) \geq \frac{L|P_{\mathbf{w}}|}{4K}.$$

Exclusion places each match in the corresponding listening set, while overlap bounds the sum by $64\alpha_0 L|P_{\mathbf{w}}|/K < L|P_{\mathbf{w}}|/(4K)$. Thus $|C_{\mathbf{w}}| < L$.

For a class-2 raw entry y , set

$$W_y := |\{\mathbf{w} : y \in C_{\mathbf{w}}\}|, \quad D_0 := d\ell_{\text{col}}S_0^{d-1}.$$

We claim that

$$0 \leq W_y \leq D_0, \quad \sum_y W_y < LS_0^d. \quad (104)$$

The sum counts pairs $(\mathbf{w}, y)$ with $y \in C_{\mathbf{w}}$. To bound W_y , suppose ℓ_{col} counted tuples have distinct columns within each sender, and denote them by $\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(\ell_{\text{col}})}$. Quota implies $s \geq e_*$ and hence $h_{\text{exc}} \leq s/(128\kappa_{\text{arr}}K)$.

If $|I_y| \geq s/(\kappa_{\text{arr}}K)$, A2 gives

$$\sum_{r=1}^{\ell_{\text{col}}} |P_{\mathbf{w}^{(r)}} \cap I_y| \geq \frac{\ell_{\text{col}}}{16}(|I_y| - h_{\text{exc}}) > 0,$$

contrary to exclusion for y . Otherwise class 2 gives $|J_y| \geq s/(\kappa_{\text{arr}}K)$, and A2 gives

$$\sum_{r=1}^{\ell_{\text{col}}} |P_{\mathbf{w}^{(r)}} \cap J_y| \geq \frac{\ell_{\text{col}}s}{32\kappa_{\text{arr}}K} > \frac{64\alpha_0\ell_{\text{col}}s}{K}.$$

This contradicts overlap, since $|P_{\mathbf{w}^{(r)}}| \leq s$ and $\alpha_0 < 1/(2048\kappa_{\text{arr}})$. Therefore a maximal family of counted tuples with disjoint columns in every sender has fewer than ℓ_{col} members. Every counted tuple shares one of fewer than $d\ell_{\text{col}}$ sender-column pairs with that family; each pair lies in S_0^{d-1} tuples. This proves (104).

A5: permutation labels. Condition further on the raw arrays. The weights W_y are fixed. Actual target addresses cannot produce an incorrect label; conditional on their locations, the remaining labels are uniformly permuted within each column, independently between columns. At most $2A$ ROOT labels have the specified destination and echo. Since $d_j \geq 1$, $Q_{\text{arr}} - 1 \geq A^2$, so for k distinct remaining addresses in one column,

$$\Pr\{\text{all } k \text{ labels qualify}\} \leq \frac{(2A)_k}{(Q_{\text{arr}} - 1)_k} \leq \phi^k, \quad \phi := 2/A.$$

Independence between columns gives the bound $\phi^{|D|}$ for any set D of remaining addresses.

Let $\mathcal{H}$ be the sigma-field generated by the fixed A5 input, the nontarget permutations, the target actual-message locations, and all raw arm-array coordinates. Let ι_y indicate that a remaining class-2 address carries

such a label, and put $Z = \sum_y W_y t_y$. Every tuple counted as bad in A5 contributes at least one to Z . Expanding the product below and using (104) gives, for $\lambda > 0$,

$$\begin{aligned}\mathbb{E}[e^{\lambda Z} \mid \mathcal{H}] &= \mathbb{E}\left[\prod_y \{1 + (e^{\lambda W_y} - 1)t_y\} \mid \mathcal{H}\right] \\ &\leq \prod_y \{1 + \phi(e^{\lambda W_y} - 1)\} \\ &\leq \exp\left\{\frac{\phi L S_0^d}{D_0}(e^{\lambda D_0} - 1)\right\}.\end{aligned}$$

Here $e^{\lambda w} - 1 \leq w(e^{\lambda D_0} - 1)/D_0$ for $0 \leq w \leq D_0$. Taking $\lambda = D_0^{-1} \log(\epsilon_{\text{arr}}/(\phi L)) > 0$ and applying Markov's inequality gives

$$\Pr\{Z > \epsilon_{\text{arr}} S_0^d \mid \mathcal{H}\} \leq \exp\left\{-\frac{\epsilon_{\text{arr}} S_0}{d\ell_{\text{col}}} \log \frac{\epsilon_{\text{arr}} A}{2eL}\right\}. \quad (105)$$

The A5 probability is the number of bad tuples divided by S_0^d ; thus (105) bounds its failure. The number of external input choices is at most $\text{poly}(N)2^{Kn}(2K)^n Q_{\text{arr}}^m A$, whose logarithm is $O(\Gamma)$. The parameter bounds above and a union bound make the probability of A5 failure together with A1'–A2 success at most N^{-40} .

A6 and A7. For fixed player, block, position, and arm, the number of columns listening on that arm is $\text{Bin}(S_0, \alpha_0/K)$. For a fixed native message, the number whose played arm is that arm is $\text{Bin}(S_0, 1/K)$. Chernoff bounds for the two failures in A6 are, respectively,

$$2e^{-c\alpha_0 S_0/K} \quad \text{and} \quad e^{-cS_0/(m^2 K)}.$$

The second term of (12) makes their exponents arbitrarily large multiples of Λ by increasing C_S . There are $O(mTKQ_{\text{arr}}) = O(N^3)$ index choices, so A6 fails with probability at most CN^{-40} .

For A7, the masks are fixed. For each player, block, column, and mask row, let $X = |\mathcal{L}_w \cap \mathcal{R}_{q,b}|$ and $\mu = \alpha_0 |\mathcal{R}_{q,b}|$. Then X is binomial and

$$\mathbb{E}e^X \leq e^{(e-1)\mu}, \quad \Pr\{X > 2\mu + C_0\Lambda\} \leq e^{-C_0\Lambda}.$$

There are at most $mB_{\text{blk}}S_0M = O(N^{9/2})$ choices. Increasing C_0 makes their union probability at most N^{-40} .

A8. Fix its common data and $x_{-i,-j}$. Condition on the nontarget raw arrays and entries and on the target permutations. Then the target's address y_w is fixed in each column, whereas its raw arm coordinates remain independent over w . Put

$$F := \{t \in E : d_{x_k}(t) \neq b_i(t) \text{ for every } k \notin \{i, j\}\}.$$

For every target column w , $P_w(x_{-i,-j}) \subseteq F$. If $|F| < e_*$, A8 is immediate. Otherwise let $f_0 = |F|$. For $w \in [S_0]$ and code row $h \in [H_{\text{arr}}]$, count the positions $t \in F$ where the target does not listen, uses row h , and leaves $b_i(t)$ as its unplayed pair endpoint; call this count $X_{w,h}$. The fair offset and independent pair give

$$X_{w,h} \sim \text{Bin}\left(f_0, \frac{1 - \alpha_0}{KH_{\text{arr}}}\right).$$

Each counted position belongs to $P_w(x_{-i,-j})$ and supplies an exclusion equation in row h . For large C_0 , $e_* \geq 128KH_{\text{arr}}$; thus $f_0 \geq e_*$ implies

$$\left\lceil \frac{f_0}{64KH_{\text{arr}}} \right\rceil \leq \frac{(1 - \alpha_0)f_0}{2KH_{\text{arr}}}.$$

Chernoff's inequality and a union over h give

$$\Pr\left\{\min_h X_{w,h} < \left\lceil \frac{f_0}{64KH_{\text{arr}}} \right\rceil\right\} \leq H_{\text{arr}} e^{-ce_*/(KH_{\text{arr}})} \leq N^{-30}.$$

Since $|P_w| \leq f_0$, the complementary event implies $\text{Cov}_{i,j,b,w}(P_w)$. Target columns are independent under the stated conditioning. Apply (103), then union over the external inputs, whose number has logarithm $O(Kn + n \log(2K) + m \log(Q_{\text{arr}}S_0) + \Lambda) = O(\Gamma)$. The failure probability is at most N^{-40} , uniformly under the conditioning. This argument does not condition on P_w .

Finally, union the failures of A1'–A4 and A6–A8, the A5 failure on A1'–A2 success, and code-certification failure. Their probability is at most $C_{\text{arr}}N^{-40}$. $\square$

References

- [ALK20] Pragnya Alatur, Kfir Y. Levy, and Andreas Krause. Multi-player bandits: The adversarial case. *Journal of Machine Learning Research*, 21(77):1–23, 2020. <https://www.jmlr.org/papers/v21/19-912.html>.
- [AVW87] V. Anantharam, P. Varaiya, and J. Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays—part i: I.i.d. rewards. *IEEE Transactions on Automatic Control*, 32(11):968–976, 1987. <https://doi.org/10.1109/TAC.1987.1104491>.
- [BB20] Sébastien Bubeck and Thomas Budzinski. Coordination without communication: Optimal regret in two players multi-armed bandits. In *Proceedings of the 33rd Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 916–939, 2020. <https://proceedings.mlr.press/v125/bubeck20a.html>.
- [BBM⁺18] Rémi Bonnefoi, Lilian Besson, Christophe Moy, Emilie Kaufmann, and Jacques Palicot. Multi-armed bandit learning in IoT networks: Learning helps even in non-stationary settings. In *Cognitive Radio Oriented Wireless Networks*, pages 173–185. Springer International Publishing, 2018. https://doi.org/10.1007/978-3-319-76207-4_15.
- [BBS21] Sébastien Bubeck, Thomas Budzinski, and Mark Sellke. Cooperative and stochastic multi-player multi-armed bandit: Optimal regret with neither communication nor collisions. In *Proceedings of the 34th Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 821–822, 2021. <https://proceedings.mlr.press/v134/bubeck21b.html>.
- [BK18] Lilian Besson and Emilie Kaufmann. Multi-player bandits revisited. In *Proceedings of Algorithmic Learning Theory*, volume 83 of *Proceedings of Machine Learning Research*, pages 56–92, 2018. <https://proceedings.mlr.press/v83/besson18a.html>.
- [BLM13] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. <https://doi.org/10.1093/acprof:oso/9780199535255.001.0001>.
- [BLPS20] Sébastien Bubeck, Yuanzhi Li, Yuval Peres, and Mark Sellke. Non-stochastic multi-player multi-armed bandits: Optimal rate with collision information, sublinear without. In *Proceedings of the 33rd Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 961–987, 2020. <https://proceedings.mlr.press/v125/bubeck20c.html>.
- [BP19] Etienne Boursier and Vianney Perchet. SIC-MMAB: Synchronisation involves communication in multiplayer multi-armed bandits. In *Advances in Neural Information Processing Systems* 32, pages 12071–12080, 2019. https://proceedings.neurips.cc/paper_files/paper/2019/hash/c4127b9194fe8562c64dc0f5bf2c93bc-Abstract.html.
- [Dur64] Richard Durstenfeld. Algorithm 235: Random permutation. *Communications of the ACM*, 7(7):420, 1964.
- [HCT22] Wei Huang, Richard Combes, and Cindy Trinh. Towards optimal algorithms for multi-player bandits without collision sensing information. In *Proceedings of the 35th Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 1990–2012, 2022. <https://proceedings.mlr.press/v178/huang22a.html>.

- [HLP52] G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities*. Cambridge University Press, 2 edition, 1952.
- [Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. <https://doi.org/10.1080/01621459.1963.10500830>.
- [HRMS20] Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform chernoff bounds via nonnegative supermartingales. *Probability Surveys*, 17, 2020. <https://doi.org/10.1214/18-PS321>.
- [LM22] Gábor Lugosi and Abbas Mehrabian. Multiplayer bandits without observing collision information. *Mathematics of Operations Research*, 47(2):1247–1265, 2022. <https://doi.org/10.1287/moor.2021.1168>.
- [LZ10] Keqin Liu and Qing Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE Transactions on Signal Processing*, 58(11):5667–5681, 2010. <https://doi.org/10.1109/TSP.2010.2062509>.
- [RSS16] Jonathan Rosenski, Ohad Shamir, and Liran Szlak. Multi-player bandits—a musical chairs approach. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 155–163, 2016. <https://proceedings.mlr.press/v48/rosenski16.html>.
- [SS21] Chengshuai Shi and Cong Shen. On no-sensing adversarial multi-player multi-armed bandits with collision communications. *IEEE Journal on Selected Areas in Information Theory*, 2(2):515–533, 2021. <https://doi.org/10.1109/JSAIT.2021.3076027>.
- [SXS20] Chengshuai Shi, Wei Xiong, Cong Shen, and Jing Yang. Decentralized multi-player multi-armed bandits with no collision information. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1519–1528, 2020. <https://proceedings.mlr.press/v108/shi20a.html>.