

# Linearly bounded liars, adaptive covering codes, and deterministic random walks

Joshua N. Cooper

University of South Carolina, Columbia, South Carolina  
cooper@math.sc.edu

Robert B. Ellis\*

Illinois Institute of Technology, Chicago, IL  
rellis@math.iit.edu

December 17, 2010

## Abstract

We analyze a deterministic form of the random walk on the integer line called the *liar machine*, similar to the rotor-router model, finding asymptotically tight pointwise and interval discrepancy bounds versus random walk. This provides an improvement in the best-known winning strategies in the binary symmetric pathological liar game with a linear fraction of responses allowed to be lies. Equivalently, this proves the existence of adaptive binary block covering codes with block length  $n$ , covering radius  $\leq fn$  for  $f \in (0, 1/2)$ , and cardinality  $O(\sqrt{\log \log n}/(1-2f))$  times the sphere bound  $2^n / \binom{n}{\lfloor fn \rfloor}$ .

## 1 Introduction

In this paper we employ machinery of deterministic random walks to produce an improved strategy in the pathological liar game with a linearly bounded liar. We also provide discrepancy bounds of independent interest for a discretized random walk which we call the “liar machine.” Liar games, introduced by Rényi and Ulam [10, 13], are played by a questioner and responder, whom we can Paul and Carole, respectively, according to tradition; they model search in the presence of error. The original variant is like “twenty questions” to identify a distinguished element of the search space, except with lies; while in the pathological variant, Carole lies as much possible, and Paul tries to preserve at least one element of the search space. Winning strategies in liar games correspond to adaptive codes, introduced by Berlekamp [1]. A primary objective in developing winning strategies for liar

---

\*Project sponsored by the National Security Agency under Grant Number #H98230-07-1-0029. The United States Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation herein.

games is to optimize the size of a search space that can be processed given the number of questions Paul can ask and a constraint on how Carole may lie. Translated into coding theory language, this objective is to optimize the size of a message set that can be handled given the number of bits to be transmitted and a constraint on how noise can corrupt the transmission. Berlekamp’s codes in [1] are adaptive packing codes for error correction, corresponding to the original liar game, whereas the pathological liar game, introduced by the second author and Yan [6], corresponds to adaptive covering codes. When both liar game variants have the same parameters, except for the variant-dependent winning condition for Paul, these parameters determine the generalization of the Hamming ball in the discrete hypercube that simultaneously defines the packing object for the corresponding adaptive packing codes, and the covering object for the corresponding adaptive covering codes (see, for example, Theorem 19 of [8]). For both the liar game and adaptive coding viewpoints there is a theoretical size limit on the search space, called the sphere bound, that provides the target for optimization, often in terms of a multiple of the sphere bound.

We combine two ideas to improve the best-known winning strategy for the pathological liar game with Yes-No questions and a linearly bounded liar. The first idea is to reduce the pathological liar game to a chip-moving machine on the integer line, which we call the liar machine, introduced implicitly by Spencer and Winkler for the original liar game [12]. The second is to adapt the analysis of deterministic random walks on the integers, developed by the first author, Doerr, Spencer, and Tardos [3], to the time-evolution of the liar machine, and confirm a winning strategy in the pathological liar game. Our main results are pointwise and interval discrepancy bounds on the time-evolution of the liar machine as compared to random walks on the integers, in Theorems 2 and 3; and an improved upper bound on the size of the search space for which Paul can win the pathological liar game with Yes-No questions and a linearly bounded liar, in Theorem 4.

## 2 Definitions and main results

### 2.1 The liar game and pathological variant

The Rényi-Ulam liar game is an  $n$ -round 2-person question-and-answer game on a search space  $[M] := \{1, \dots, M\}$ . A fixed integer parameter  $e \geq 0$  is the maximum number of *lies* an element of the search space can accumulate before being disqualified, and the game begins with an initial function  $\ell : \{1, \dots, M\} \rightarrow \{0, 1, \dots, e\}$ , representing the initial assignment of up to  $e$  lies to each  $y \in [M]$ . As elements of  $M$  are distinguished only by their number of lies, we may ignore element labels and consider instead the initial state vector  $x_0 = (x_0(0), x_0(1), \dots, x_0(e))$ , where  $x_0(i) = |\{y \in [M] : \ell(y) = i\}|$  is the number of elements of  $[M]$  initialized with  $i$  lies. Most often we set  $x_0 = (M, 0, \dots, 0)$ . Paul and Carole play an  $n$ -round game in which Paul attempts to discover a distinguished element  $z \in [M]$  of the search space. To start each round, Paul weakly partitions  $[M]$  into two parts by choosing a *question*  $(A_0, A_1)$  such that  $[M] = A_0 \cup A_1$ , where  $\cup$  denotes disjoint union. We interpret this choice as the question, “Is  $z \in A_0$ ?” Carole completes the round

by responding with her *answer*, an index  $j \in \{0, 1\}$ . For each  $y \in [M]$ , if  $y \in A_j$ , no additional lie is assigned to  $y$ , but if  $y \in A_{1-j}$ , one additional lie is assigned to  $y$ . Any  $y \in [M]$  accumulating  $e + 1$  lies is *disqualified*. We interpret Carole's answer of  $j = 0$  as "Yes" and of  $j = 1$  as "No." Analogous to the definition of  $x_0$ , for each  $s = 1, \dots, n$ , let the state vector  $x_s = (x_s(0), \dots, x_s(e))$  record the number of elements  $x_s(i)$  that have  $i$  lies at the end of round  $s$ . Paul's question  $(A_0, A_1)$  in round  $s$  corresponds to a question vector  $a_s = (a_s(0), \dots, a_s(e))$  with  $0 \leq a_s(i) \leq x_{s-1}(i)$  for all  $0 \leq i \leq e$ , by letting  $a_s(i)$  count the number of elements in  $A_0$  that have  $i$  lies at the end of round  $s - 1$ . Define the right-shift operator  $R$  on any vector  $x = (x(0), \dots, x(e))$  by  $R(x) = (0, x(0), \dots, x(e - 1))$ . Given  $x_{s-1}$  and  $a_s$ , define

$$\begin{aligned} Y(x_{s-1}, a_s) &:= a_s + R(x_{s-1} - a_s), \\ N(x_{s-1}, a_s) &:= x_{s-1} - a_s + R(a_s); \end{aligned}$$

and for each  $s = 1, \dots, n$ , set  $x_s = Y(x_{s-1}, a_s)$  if Carole responds  $j = 0$  ("Yes") in round  $s$ , and otherwise  $x_s = N(x_{s-1}, a_s)$  if Carole responds  $j = 1$  ("No") in round  $s$ . Elements  $y \in [M]$  that accumulate  $e + 1$  lies are shifted out to the right and may be ignored for the rest of the game. Paul wins the original liar game if  $\sum_{i=0}^e x_n(i) \leq 1$ , that is, if all but at most one element are disqualified after  $n$  rounds; he wins the pathological liar game if  $\sum_{i=0}^e x_n(i) \geq 1$ , that is, if at least one element survives after  $n$  rounds. We are primarily interested in the pathological variant, which may be interpreted as having a capricious Carole lying to eliminate elements as quickly as possible, while Paul forms questions to prevent all elements from being disqualified. The type of question that we will use to connect the pathological liar game with the liar machine (defined next in Section 2.2) is the *alternating question*: Paul enumerates the elements  $1, \dots, M$  with the restriction that elements with more lies get higher labels, and he puts the even-labeled chips in  $A_0$  and the odd-labeled chips in  $A_1$ . We summarize the pathological liar game as follows.

**Definition 1.** Let  $n, M, e \geq 0$  be integers, and let  $x = (x(0), x(1), \dots, x(e))$  be a non-negative integer vector with  $\sum_{i=0}^e x(i) = M$ . Define the  $(x, n, e)_2^*$ -game to be the  $n$ -round pathological liar game with Yes-No questions, initial configuration  $x$ , and  $e$  lies. We say that Paul can win the  $(x, n, e)_2^*$ -game provided there exists a winning strategy for Paul regardless of Carole's responses.

In the notation  $(x, n, e)_2^*$ , use of the asterisk indicates the pathological variant of the liar game rather than the original. The subscript 2 means that questions are binary and symmetric with respect to replacing  $a_s$  with  $x_{s-1} - a_s$  while preserving the same two vectors as candidates for  $x_s$ . This corresponds in coding theory to the binary symmetric channel assumption; see [5] for a much broader class of channel assumptions.

## 2.2 The liar machine and the linear machine

We define the "liar machine" as follows. Start with some configuration of chips on the even or odd integers (but not both). At each time step number the chips  $c_1, c_2, \dots$ , from left

to right. At each location with, say,  $k$  chips, send  $\lfloor k/2 \rfloor$  of the chips one step left, and  $\lfloor k/2 \rfloor$  one step right. If one chip remains (because  $k$  is odd) we break the tie by sending the highest-indexed  $c_j$  one step left if  $j$  is even or one step right if  $j$  is odd. In other words, odd piles of chips alternate sending their extra chip left or right, with the odd pile at the lowest integer sending it to the right.

Formally, define the “starting configuration” to be a map  $f_0 : \mathbb{Z} \rightarrow \mathbb{N}$  with finite support on  $2\mathbb{Z}$  or  $2\mathbb{Z} + 1$ . Then, given  $f_t : \mathbb{Z} \rightarrow \mathbb{N}$ , define  $\chi_t : \mathbb{Z} \rightarrow \{-1, 0, 1\}$  by

$$\chi_t(j) = \begin{cases} 0 & \text{if } f_j \equiv 0 \pmod{2} \\ (-1)^{\sum_{i < j} \chi_t(i)} & \text{if } f_j \equiv 1 \pmod{2}. \end{cases} \quad (1)$$

Then we define

$$f_{t+1}(j) = \frac{f_t(j-1) + f_t(j+1) + \chi_t(j-1) - \chi_t(j+1)}{2}.$$

Now, we define the “linear machine” by taking  $g_0 : \mathbb{Z} \rightarrow \mathbb{R}$  to be any function. Let the operator  $\mathcal{L} : \mathbb{R}^{\mathbb{Z}} \rightarrow \mathbb{R}^{\mathbb{Z}}$  be defined by

$$\mathcal{L}g(j) = \frac{g(j-1)}{2} + \frac{g(j+1)}{2},$$

and define  $g_{t+1} = \mathcal{L}g_t$ . Then  $g_t(j)$  is just the expected number of chips at location  $j$  after a simple random walk on  $\mathbb{Z}$  starting from the configuration  $g_0$ . In particular, we expect  $g_t$  and  $f_t$  to be relatively close to one another if  $g_0 \equiv f_0$ . Also, define the operator  $\Delta : \mathbb{R}^{\mathbb{Z}} \rightarrow \mathbb{R}^{\mathbb{Z}}$  by

$$\Delta f(j) = f(j-1).$$

It is easy to see that  $\mathcal{L}$  and  $\Delta$  are linear, and they commute with each other. We write  $\delta_j \in \mathbb{Z}^{\mathbb{Z}}$  for the function which is 1 at  $j$  and 0 elsewhere. In order to consider intervals in a configuration, for a set  $S \subset \mathbb{Z}$  and a function  $h : \mathbb{Z} \rightarrow \mathbb{R}$ , define  $h(S) = \sum_{i \in S} h(i)$ .

### 2.3 Main results

Our first two main results are a pointwise and an interval discrepancy bound in the time-evolution of the liar machine versus the linear machine starting with the same initial configuration; these bounds are absolute, independent of the initial configuration. All logarithms, here and throughout, are natural logarithms.

**Theorem 2.** Let  $f_0 \equiv g_0$ , and define  $f_t$  and  $g_t$  according to the evolution of the liar machine and linear machine, respectively, as described above. Then

$$|f_t(j) - g_t(j)| < 12 \log t$$

for all  $t \geq 2$ ,  $j \in \mathbb{Z}$ .

**Theorem 3.** Let  $I = [a, b] \subset \mathbb{Z}$  and  $f_0 \equiv g_0$ , and define  $f_t(I)$  and  $g_t(I)$  according to the evolution of the liar machine and linear machine, respectively, as described above. Then

$$|f_t(I) - g_t(I)| \leq c' \cdot \begin{cases} \sqrt{t} & \text{if } B > \sqrt{t}/2 \\ B \log(t/B^2) & \text{if } B \leq \sqrt{t}/2, \end{cases}$$

where  $B = b - a$  and  $c'$  is an absolute constant.

In Corollary 8 we prove that Theorems 2 and 3 are tight up to a constant multiple for a general initial configuration  $f_0$ . Corollary 25 allows extraction of a winning strategy for the pathological liar game from the time-evolution of the liar machine, yielding the following improved bound for the pathological liar game.

**Theorem 4.** Let  $M = \frac{2^n}{\binom{n}{\lfloor fn \rfloor}} (4/(1 - 2f))c' \sqrt{\log \log n} (1 + o(1))$ , where  $c'$  is the constant from Theorem 3. Then for  $n$  sufficiently large, Paul can win the  $((M, 0, \dots, 0), n, \lfloor fn \rfloor)_2^*$ -pathological liar game with  $M$  elements and  $\lfloor fn \rfloor$  lies on the binary symmetric channel.

We now discuss the improvement provided by Theorem 4. The previous best known bound on  $M$  for  $f \in (0, 1/2)$  is Theorem 1 of [4], which in our language bounds the smallest  $M$  for which Paul can win the  $((M, 0, \dots, 0), n, \lfloor fn \rfloor)_2^*$ -game with a restricted strategy (called “non-adaptive” in the literature) of selecting all questions before any responses from Carole are available.

**Theorem 5** (Delsarte and Piret). Let  $f \in (0, 1/2)$ . The minimum  $M$  for which Paul can win the  $((M, 0, \dots, 0), n, \lfloor fn \rfloor)_2^*$ -game with the restriction that all  $n$  questions must be formed before any responses from Carole are available is bounded by

$$M \leq \left\lceil \frac{2^n}{\binom{n}{\lfloor fn \rfloor}} n \log 2 \right\rceil.$$

The quantity  $2^n / \binom{n}{\lfloor fn \rfloor}$  is called the sphere bound, and so Theorem 4 provides an improved *density* in the best-known minimum  $M$ , from a linear to sub-logarithmic factor in  $n$  times the sphere bound. The sphere bound is an immediate lower bound on  $M$ ; this can be seen by defining an appropriate weight function on the liar game state which Carole greedily minimizes (cf. [7, Lemma 3]). In Theorem 5, the “spheres” are Hamming balls of radius  $\lfloor fn \rfloor$  that are used to cover the binary discrete hypercube (Hamming space) of dimension  $n$ . The equivalence of winning strategies in the pathological liar game to coverings of Hamming space by objects of size  $\binom{n}{\lfloor fn \rfloor}$  is proved in Theorem 3.7 of [5].

We conclude the section by outlining the rest of the paper. Section 3 contains the proofs of the liar machine discrepancy bounds: Theorems 2 and 3, and Corollary 25. Section 4 proves several technical distributional facts about the binomial and hypergeometric distributions needed to bound the distribution of chips in the liar machine (via discrepancy from the linear machine). Section 5 reduces a strategy for Paul in the pathological liar game to the liar machine and blends the preceding results into Theorem 4. Section 6 contains open questions and closing remarks.

### 3 Proofs of liar machine discrepancy bounds

The proofs of Theorems 2 and 3 flow directly from the definitions in Section 2.2, and resemble the arguments in [3]. These proofs require a bound on, and the bimodality in space of, a term that tracks the discrepancy between the liar machine and the linear machine; we state these technical properties as Lemma 6 but defer their proof until after those of Theorems 2 and 3. Next, Lemma 7 shows that the parity of the number of chips in the liar machine can be pre-selected for an arbitrary product of intervals in space and time, by choosing an appropriate initial configuration. This leads to a complementary lower bound in Corollary 8 on discrepancy for a general initial configuration. We adopt the convention, here and throughout, that  $\binom{a}{b}$  is zero unless  $a$  and  $b$  are nonnegative integers and  $b \leq a$ .

**Lemma 6.** There exist constants  $c_1, c_2, c_3,$  and  $c_4$  so that the following holds for even  $B > 0$ . Define

$$h_B(j) = 2^{-s} \left( \binom{s}{(s+j-B)/2} - \binom{s}{(s+j)/2} \right).$$

Then, when  $B \geq \sqrt{s}$ ,

$$\frac{c_1}{\sqrt{s}} \leq \max_j |h_B(j)| \leq \frac{c_2}{\sqrt{s}}, \quad (2)$$

and when  $B \leq \sqrt{s}$ ,

$$\frac{c_3 B}{s} \leq \max_j |h_B(j)| \leq \frac{c_4 B}{s};$$

these inequalities hold for all  $s \geq 1$  except for the left-hand inequality in (2), which holds for all  $s \geq S$ , some absolute constant. Furthermore,  $h_B(j)$  is bimodal on its support.

*Proof of Theorem 2.* Evidently,

$$f_{t+1} = \mathcal{L}f_t + \frac{1}{2}(\Delta - \Delta^{-1})\chi_t.$$

Therefore, by the linearity of  $\mathcal{L}$  and the fact that it commutes with  $\Delta$ ,

$$f_t = \mathcal{L}^t f_0 + \frac{1}{2} \sum_{s=0}^{t-1} (\Delta - \Delta^{-1}) \mathcal{L}^s \chi_{t-1-s}.$$

Since  $g_t = \mathcal{L}^t g_0 = \mathcal{L}^t f_0$ ,

$$\begin{aligned} 2|f_t - g_t| &= \left| \sum_{s=0}^{t-1} (\Delta - \Delta^{-1}) \mathcal{L}^s \chi_{t-1-s} \right| \\ &\leq 2 + \sum_{s=1}^{t-1} |(\Delta - \Delta^{-1}) \mathcal{L}^s \chi_{t-1-s}|. \end{aligned}$$

Consider a fixed  $s$ . Denote by  $z_i$  the  $i^{\text{th}}$  element of the support of  $\chi_{t-1-s}$ , with  $z_0$  its minimal element and  $z_{i+1} > z_i$  for each  $i$ . Note that the  $z_i$  all have the same parity, by our assumption that the chips occupy only even or only odd integers. Then

$$\begin{aligned}
(\Delta - \Delta^{-1})\mathcal{L}^s \chi_{t-1-s} &= (\Delta - \Delta^{-1})\mathcal{L}^s \sum_i (-1)^i \delta_{z_i} \\
&= \sum_i (-1)^i (\Delta - \Delta^{-1})\mathcal{L}^s \delta_{z_i} \\
&= \sum_i (-1)^i (\Delta - \Delta^{-1})\mathcal{L}^s \Delta^{z_i} \delta_0 \\
&= \sum_i (-1)^i \Delta^{z_i} (\Delta - \Delta^{-1})\mathcal{L}^s \delta_0.
\end{aligned} \tag{3}$$

Note that

$$(\Delta - \Delta^{-1})\mathcal{L}^s \delta_0(j) = 2^{-s} \left( \binom{s}{(s+j-1)/2} - \binom{s}{(s+j+1)/2} \right).$$

Therefore, by Lemma 6,  $(\Delta - \Delta^{-1})\mathcal{L}^s \delta_0$  is bimodal on its support. This means that the alternating sum  $\sum_i (-1)^i \Delta^{z_i} (\Delta - \Delta^{-1})\mathcal{L}^s \delta_0$  is bounded by at most four times the maximum (in absolute value) of the quantity  $(\Delta - \Delta^{-1})\mathcal{L}^s \delta_0$ , since the  $z_i$  all have the same parity. This maximum, by Lemma 6, is at most  $3/s$ . Therefore,

$$\begin{aligned}
|f_t - g_t| &\leq \frac{1}{2} \left( 12 \sum_{s=1}^{t-1} \frac{1}{s} + 2 \right) \\
&\leq 12 \log t.
\end{aligned}$$

□

*Proof of Theorem 3.* Without loss of generality,  $I = \{1, \dots, B\}$ . We may also assume that  $B$  is even. Evidently,

$$f_{t+1}(I) = \left( \mathcal{L}f_t(I) + \frac{1}{2}(\Delta - \Delta^{-1})\chi_t(I) \right).$$

Therefore, by the linearity of  $\mathcal{L}$  and the fact that it commutes with  $\Delta$ ,

$$f_t(I) = \mathcal{L}^t f_0(I) + \frac{1}{2} \sum_{s=0}^{t-1} (\Delta - \Delta^{-1})\mathcal{L}^s \chi_{t-1-s}(I).$$

Since  $g_t(I) = \mathcal{L}^t g_0(I) = \mathcal{L}^t f_0(I)$ ,

$$2|f_t(I) - g_t(I)| = \left| \sum_{s=0}^{t-1} (\Delta - \Delta^{-1})\mathcal{L}^s \chi_{t-1-s}(I) \right|$$

$$\leq 2 + \sum_{s=1}^{t-1} |(\Delta - \Delta^{-1})\mathcal{L}^s \chi_{t-1-s}(I)|.$$

Denote by  $z_i$  the  $i^{\text{th}}$  element of the support of  $\chi_{t-1-s}$ , with  $z_0$  its minimal element and  $z_{i+1} > z_i$  for each  $i$ . Then

$$\begin{aligned} (\Delta - \Delta^{-1})\mathcal{L}^s \chi_{t-1-s}(I) &= (\Delta - \Delta^{-1})\mathcal{L}^s \sum_i (-1)^i \delta_{z_i}(I) \\ &= \sum_i (-1)^i (\Delta - \Delta^{-1})\mathcal{L}^s \delta_{z_i}(I) \\ &= \sum_i (-1)^i (\Delta - \Delta^{-1})\mathcal{L}^s \Delta^{z_i} \delta_0(I) \\ &= \sum_i (-1)^i \Delta^{z_i} (\Delta - \Delta^{-1})\mathcal{L}^s \sum_{k=1}^B \Delta^k \delta_0(0) \\ &= \sum_i (-1)^i \Delta^{z_i} \sum_{k=1}^B \Delta^k (\Delta - \Delta^{-1})\mathcal{L}^s \delta_0(0) \\ &= \sum_i (-1)^i \Delta^{z_i} (\Delta^{B+1} + \Delta^B - \Delta - 1)\mathcal{L}^s \delta_0(0) \\ &= \sum_i (-1)^i (\Delta + 1)(\Delta^B - 1)\mathcal{L}^s \delta_0(-z_i). \end{aligned} \tag{4}$$

Note that  $\mathcal{L}^s \delta_0(j) = 2^{-s} \binom{s}{(s+j)/2}$ , so that

$$(\Delta^B - 1)\mathcal{L}^s \delta_0(j) = 2^{-s} \left( \binom{s}{(s+j-B)/2} - \binom{s}{(s+j)/2} \right).$$

By Lemma 6, when  $B = \Omega(\sqrt{s})$ , the maximum possible value of the right-hand side is  $\Theta(1/\sqrt{s})$ ; when  $B = o(\sqrt{s})$ , it is of order

$$2^{-s} B \cdot \max_j \left( \binom{s}{(s+j-1)/2} - \binom{s}{(s+j+1)/2} \right) = \Theta(B/s).$$

Since an alternating sum over a bimodal function like  $(\Delta + 1)(\Delta^B - 1)\mathcal{L}^s \delta_0$  is bounded by four times the maximum absolute value of that function,

$$\begin{aligned} |f_t(I) - g_t(I)| &\leq c \sum_{s=1}^{t-1} \frac{\min(\sqrt{s}, B)}{s+1} \\ &\leq c' \cdot \begin{cases} \sqrt{t} & \text{if } B > \sqrt{t}/2 \\ B \log(t/B^2) & \text{if } B \leq \sqrt{t}/2, \end{cases} \end{aligned}$$

for some absolute constants  $c$  and  $c'$ . □



*Proof of Lemma 6.* It is easy to see that

$$h_2(j) = 2^{-s} \frac{j-1}{s+1} \binom{s+1}{(s+j)/2}.$$

Therefore,

$$\frac{h_2(j)}{\Delta^2 h_2(j)} = \frac{(j-1)(s-j+4)}{(j-3)(s+j)},$$

which equals one when  $j^2 - 4j - (s-2) = 0$ , i.e.,  $j = 2 \pm \sqrt{s+2}$ . Then the maximum of  $|h_2(j)|$  can be bounded by

$$\begin{aligned} 2^{-s} \frac{|j_{\max}| - 1}{s+1} \max_j \binom{s+1}{(s+j)/2} &\leq \frac{1 + \sqrt{s+2}}{s+1} \cdot \frac{1}{\sqrt{3(s+1)/2}} \\ &\leq \frac{1+2}{(s+1)\sqrt{2}} < \frac{3}{s}. \end{aligned}$$

Since  $h_B(j) = \sum_{i=0}^{B/2-1} h_2(j-2i)$ , it immediately follows that

$$\max_j |h_B(j)| \leq \frac{B}{2} \max_j |h_2(j)| < \frac{3B}{2s},$$

so we may take  $c_4 = 3/2$ . Now, it is clear that

$$\begin{aligned} \max_j |h_B(j)| &< \max_j 2^{-s} \cdot \binom{s}{(s+j)/2} \\ &< \frac{1}{\sqrt{s}}, \end{aligned}$$

so we may take  $c_2 = 1$ .

On the other hand, by a version of the Local Central Limit Theorem (see, e.g., [9, Thm. 1.2.1]),  $2^{-s} \binom{s}{(s+j)/2} = \sqrt{\frac{2}{\pi s}} \cdot e^{-j^2/2s} + O(s^{-3/2})$ , so that we have

$$h_B(j) = \sqrt{\frac{2}{\pi s}} \cdot e^{-(j-B)^2/2s} - \sqrt{\frac{2}{\pi s}} \cdot e^{-j^2/2s} + O(s^{-3/2}).$$

Hence, when  $B \geq \sqrt{s}$ ,

$$\begin{aligned} \max_j |h_B(j)| &\geq |h_B(0)| \\ &= \left| \sqrt{\frac{2}{\pi s}} \cdot e^{-B^2/2s} - \sqrt{\frac{2}{\pi s}} + O(s^{-3/2}) \right| \\ &\geq \sqrt{\frac{2}{\pi s}} \left| e^{-1/2} - 1 \right| + O(s^{-3/2}) \end{aligned}$$

$$> \frac{1 + o(1)}{4\sqrt{s}},$$

so we may take  $c_1 = 1/4$  and  $S$  sufficiently large. Note that the error term  $O(s^{-3/2})$  is uniform in  $j$  and therefore the  $o(1)$  does not depend on  $B$ .

When  $B < \sqrt{s}$ ,

$$\begin{aligned} 2^s h_B(j) &= \binom{s}{(s+j-B)/2} - \binom{s}{(s+j)/2} \\ &= \binom{s}{(s+j)/2} \left( \prod_{i=1}^{B/2} \frac{s+j-B+2i}{s-j+2i} - 1 \right) \\ &= \binom{s}{(s+j)/2} \left( \prod_{i=1}^{B/2} \left( 1 + \frac{2j-B}{s-j+2i} \right) - 1 \right), \end{aligned}$$

so we have

$$\begin{aligned} \max_j h_B(j) &\geq h_B(\sqrt{s}) \\ &= 2^{-s} \binom{s}{(s+\sqrt{s})/2} \left( \prod_{i=1}^{B/2} \left( 1 + \frac{2\sqrt{s}-B}{s-\sqrt{s}+2i} \right) - 1 \right) \\ &\geq 2^{-s} \binom{s}{(s+\sqrt{s})/2} \left( \left( 1 + \frac{\sqrt{s}}{s} \right)^{B/2} - 1 \right) \\ &\geq \frac{c_0}{\sqrt{s}} \cdot \frac{B}{2\sqrt{s}} = \frac{c_0 B}{2s}, \end{aligned}$$

so we can take  $c_3 = c_0/2$ .

Finally, we have

$$\begin{aligned} 2^s (h_B(j-2) - h_B(j)) &= \binom{s}{(s+j-B)/2-1} - \binom{s}{(s+j-B)/2} \\ &\quad - \binom{s}{(s+j)/2-1} + \binom{s}{(s+j)/2} \\ &= 2^s (h_2(j-B) - h_2(j)) \\ &= \frac{j-B-1}{s+1} \binom{s+1}{(s+j-B)/2} - \frac{j-1}{s+1} \binom{s+1}{(s+j)/2}. \end{aligned}$$

This quantity is positive when

$$(j-B-1) \binom{s+1}{(s+j-B)/2} > (j-1) \binom{s+1}{(s+j)/2},$$

i.e.,

$$(j - B - 1) \prod_{i=1}^{B/2} (s + j - 2i + 2) > (j - 1) \prod_{i=1}^{B/2} (s - j + 2i + 2).$$

(We may assume that each term of both products is nonnegative.) When  $1 \leq j \leq B + 1$ , this inequality cannot be satisfied, since the left-hand side is nonpositive and the right-hand side is nonnegative. When  $j > B + 1$ , the inequality is the same as

$$\left(1 - \frac{B}{j - 1}\right) \prod_{i=1}^{B/2} (s + j - 2i + 2) > \prod_{i=1}^{B/2} (s - j + 2i + 2).$$

The left-hand side is nondecreasing in  $j$  and the right-hand side is nonincreasing in  $j$ , so  $h_B(j - 2) - h_B(j)$  has at most one change of sign in this regime. When  $j < 1$ , we have the condition

$$\prod_{i=1}^{B/2} (s + j - 2i + 2) < \left(1 + \frac{B}{j - B - 1}\right) \prod_{i=1}^{B/2} (s - j + 2i + 2),$$

where again the left-hand side is nondecreasing in  $j$  and the right-hand side is nonincreasing in  $j$ , so  $h_B(j - 2) - h_B(j)$  has at most one more change of sign. Therefore,  $h_B(j)$  is bimodal on its support. □

**Lemma 7.** For each function  $g : \{0, \dots, N - 1\} \times \{0, \dots, T - 1\} \rightarrow \{0, 1\}$ , there exists a chip-assignment function  $f_0 : \mathbb{Z} \rightarrow \mathbb{N}$  so that, for all  $0 \leq n < N$  and  $0 \leq t < T$ ,

$$f_t(n) \equiv g(n, t) \pmod{2},$$

where  $f_t$  is the state of the liar machine at time  $t$  if  $f_0$  is its initial state (i.e., at time  $t = 0$ ).

*Proof.* We proceed by induction. For  $T = 1$ , the result is immediate: we simply set  $f_0 \equiv g(\cdot, 0)$ . Suppose that the claim holds for  $T$ , i.e., there exists an  $f_0$  so that  $f_t$  agrees with  $g(\cdot, t)$  in parity for each  $t \in \{0, \dots, T - 1\}$ . Now we perform a second induction (on  $n$ ) to show the following claim:

**Claim.** For each  $n \in \{0, \dots, N - 1\}$ , there exists a chip-assignment function  $f_0^{(n)} : \mathbb{Z} \rightarrow \mathbb{N}$  so that, for all pairs  $(n', t)$  with  $0 \leq n' < N$  and  $0 \leq t < T$  or  $0 \leq n' < n$  and  $t = T$ ,

$$f_t^{(n)}(n') \equiv g(n', t) \pmod{2},$$

where  $f_t^{(n)}$  is the state of the liar machine at time  $t$  if  $f_0^{(n)}$  is its initial state.

Again, the claim is immediate for  $n = 0$  (given the inductive hypothesis), since we can just let  $f_0^{(0)} = f_0$  from the top-level induction. Suppose it holds for  $n$ . If  $f_T^{(n)}(n) \equiv g(n, T)$

(mod 2), then setting  $f_0^{(n+1)} = f_0^{(n)}$  clearly suffices to prove the claim for  $n+1$ . If, however,  $f_T^{(n)}(n) \not\equiv g(n, T) \pmod{2}$ , then define  $f_0^{(n+1)}$  by

$$f_0^{(n+1)}(k) = \begin{cases} f_0^{(n)}(k) & \text{if } k \neq n+T \\ f_0^{(n)}(k) + 2^T & \text{if } k = n+T. \end{cases}$$

Then  $f_t^{(n+1)}(k) \equiv f_t^{(n)}(k) \pmod{2}$  for  $t < T$  and  $0 \leq k < N$ , since the “new”  $2^T$  chips placed at site  $n+T$  at time  $t=0$  are split exactly in half at each time  $t < T$ , so that  $2|2^{T-t}|f_t^{(n+1)}(k) - f_t^{(n)}(k)$  for all  $t < T$ . For  $t=T$  and  $k < n$ ,  $f_t^{(n+1)}(k) = f_t^{(n)}(k)$ , since the “new” chips can only occupy sites in  $[n+T-t, n+T+t]$  at time  $t$ , which for  $T=t$  is the interval  $[n, n+2T]$  not containing  $k$ . Finally, there is one chip added to site  $n$  at time  $T$ , i.e.,  $f_T^{(n+1)}(n) = f_T^{(n)}(n) + 1$ , because exactly one of the  $2^T$  “new” chips makes it to site  $n$  after  $T$  steps. This means, in particular, that  $f_T^{(n+1)}(n) \equiv g(n, T) \pmod{2}$ , completing the induction.  $\square$

This “parity forcing” lemma implies that it is possible to set the function  $\chi_t(j)$  for any finite space-time interval to whatever we wish. We may now conclude that Theorems 2 and 3 are tight up to asymptotic order of magnitude.

**Corollary 8.** Fix  $T$ , a nonnegative integer, and  $N$ , an integer. There exists an  $f_0 : \mathbb{Z} \rightarrow \mathbb{N}$  so that, letting  $g_0 \equiv f_0$ , and defining  $f_t$  and  $g_t$  according to the evolution of the liar machine and linear machine, respectively, we have

$$|f_T(N) - g_T(N)| = \Omega(\log T).$$

Fix an interval  $I$  of any given length  $B$ . Then there exists an  $f'_0 : \mathbb{Z} \rightarrow \mathbb{N}$  so that, letting  $g'_0 \equiv f'_0$ , and defining  $f'_t$  and  $g'_t$  according to the evolution of the liar machine and linear machine, respectively, we have

$$|f'_T(I) - g'_T(I)| = \Omega \left( \begin{cases} \sqrt{T} & \text{if } B > \sqrt{T}/2 \\ B \log(T/B^2) & \text{if } B \leq \sqrt{T}/2 \end{cases} \right).$$

*Proof.* The same argument applies for both claims: we can set the number of chips at each location and time so that the sums in the proof of Theorems 2 and 3 are maximized, in view of the lower bounds given by Lemma 6. In the first case, let  $\chi : \{N-T, \dots, N+T\} \times \{0, \dots, T-1\} \rightarrow \{-1, 0, 1\}$  be chosen to maximize the sum (3); in the second case, let  $\chi : \{\min(I) - T, \dots, \max(I) + T\} \times \{0, \dots, T-1\} \rightarrow \{-1, 0, 1\}$  be chosen to maximize the sum (4). Note that this requires that  $\chi$  alternate in sign on the support of its first argument. Let  $m_t$  be the minimum element of the support of  $\chi(\cdot, t)$ . Define

$$g(k, t) = \begin{cases} \frac{1 - \chi(m_t, t)}{2} & \text{if } k = N - T - 1 \\ |\chi(k, t)| & \text{if } N - T \leq k \leq N + T \\ 0 & \text{otherwise,} \end{cases}$$

in the first case, or else

$$g(k, t) = \begin{cases} \frac{1-\chi(m_t, t)}{2} & \text{if } k = \min(I) - T - 1 \\ |\chi(k, t)| & \text{if } \min(I) - T \leq k \leq \max(I) + T \\ 0 & \text{otherwise,} \end{cases}$$

in the second case. Then, we may obtain the desired  $f_0$  by applying the preceding lemma to  $g$ . Since the (possible) chip at  $k = N - T - 1$  or  $k = \min(I) - T - 1$  can never even reach the site  $N$  or any of  $I$  before time  $T$ , the relevant sums are unaffected by this small modification. However, the presence of such a chip when appropriate ensures that  $\chi_t(j) = \chi(j, t)$  for each  $(j, t) \in \{N - T, \dots, N + T\} \times \{0, \dots, T - 1\}$  or  $(j, t) \in \{\min(I) - T, \dots, \max(I) + T\} \times \{0, \dots, T - 1\}$  (where  $\chi_t(j)$  is as defined in (1)).  $\square$

## 4 Liar machine distributional bound

We need several technical facts to obtain lower bounds for the configuration of chips in the time-evolution of the liar machine. Lemma 9 shows that the cumulative distribution of the binomial random variable drops off sharply just below where it is evaluated. Lemma 10 shows that the ratio of the evaluations at the same relative position of the cumulative distributions of binomial random variables with a similar number of trials is not too small. This is needed to bound the left tail of the liar machine from below. Because for Theorem 26 we will run  $n$  steps of the liar machine in two stages of  $n_1$  and  $n_2$  steps, respectively, terms of a hypergeometric distribution arise. Theorem 12 quotes a result on the closeness of the median to the mean of a generalized hypergeometric distribution from [11], specialized to the hypergeometric distribution in Corollary 13. Then in Proposition 14 we show that for  $r$  sufficiently close to but below the mean  $\mu$ , asymptotically almost half of the hypergeometric distribution lies below  $r$ . This allows transferring from a partial sum of hypergeometric distributions in  $n_1$  and  $n_2$  to that of the binomial distribution in  $n$ , in Proposition 15. This last result is critical for Theorem 26 in negotiating a lower bound on the number of chips between two stages in the time-evolution of the liar machine, so that at least one chip survives in a prescribed interval after  $n$  rounds.

Throughout the section, we use the following notation. Let  $n \rightarrow \infty$ , fix  $f \in (0, 1/2)$ , and set

$$n_1 = n - \left\lfloor \frac{4}{(1-2f)^2} \log \log n \right\rfloor$$

and  $n_2 = n - n_1$ . The numbers of rounds in the first and second stages of the  $n$ -round liar machine, are  $n_1$  and  $n_2$ , respectively. Define  $F = \lfloor fn \rfloor$ ,  $F_1 = \lfloor fn_1 \rfloor$ , and  $F_2 = F - F_1$ .

**Lemma 9.** For any integer sequence  $n_3 = n_3(n) \rightarrow \infty$ , there is a function  $\epsilon(n, f)$  with  $\lim_{n \rightarrow \infty} \epsilon(n, f) = 0$  so that

$$\sum_{i=F-n_3}^F \frac{\binom{n}{i}}{\binom{n}{\leq F}} \geq 1 - \epsilon(n, f).$$

*Proof.* Note that

$$\begin{aligned} \frac{\binom{n}{F-t}}{\binom{n}{F}} &= \frac{F!(n-F)!}{(F-t)!(n-F+t)!} \\ &\leq \frac{F^t}{(n-F+1)^t} \\ &\leq \frac{(fn)^t}{((1-f)n)^t} = \left(\frac{f}{1-f}\right)^t. \end{aligned}$$

Therefore,

$$\begin{aligned} \sum_{i=0}^{F-n_3} \binom{n}{i} &\leq \sum_{i=0}^{F-n_3} \binom{n}{F} \left(\frac{f}{1-f}\right)^{F-i} \\ &\leq \binom{n}{F} \sum_{j=n_3}^{\infty} \left(\frac{f}{1-f}\right)^j \\ &\leq \binom{n}{\leq F} \left(\frac{f}{1-f}\right)^{n_3} \cdot \frac{1-f}{1-2f}. \end{aligned}$$

It follows that

$$\begin{aligned} \sum_{i=F-n_3}^F \frac{\binom{n}{i}}{\binom{n}{\leq F}} &= 1 - \sum_{i=0}^{F-n_3-1} \frac{\binom{n}{i}}{\binom{n}{\leq F}} \\ &\geq 1 - \left(\frac{f}{1-f}\right)^{n_3} \cdot \frac{1-f}{1-2f}, \end{aligned}$$

which clearly tends to 1 as  $n \rightarrow \infty$ , since  $f < 1/2$  implies  $f/(1-f) < 1$ .  $\square$

**Lemma 10.** There exists a function  $\delta(n, f)$  with  $\lim_{n \rightarrow \infty} \delta(n, f) = 0$  so that

$$\frac{2^n}{\binom{n}{\leq F}} \cdot \frac{\binom{n_1}{F_1}}{2^{n_1}} \geq (\log n)^{2-\delta(n, f)}.$$

*Proof.* First of all, note that

$$\frac{2^n}{\binom{n}{\leq F}} \cdot \frac{\binom{n_1}{F_1}}{2^{n_1}} = 2^{n-n_1} \frac{\binom{n_1}{F_1}}{\binom{n}{F}} \cdot \frac{\binom{n}{F}}{\binom{n}{\leq F}}.$$

Denote by  $A$ ,  $B$ , and  $C$  the three factors on the right-hand side. Since

$$n - n_1 = \left\lfloor \frac{4}{(1-2f)^2} \log \log n \right\rfloor \geq \frac{4}{(1-2f)^2} \log \log n - 1,$$

we have

$$A \geq \frac{1}{2} (\log n)^{4 \log 2 / (1-2f)^2}.$$

Then, applying the estimates from the proof of Lemma 9,

$$\begin{aligned} \binom{n}{\leq F} &= \binom{n}{F} \sum_{t=0}^F \frac{\binom{n}{F-t}}{\binom{n}{F}} \\ &\leq \binom{n}{F} \sum_{t=0}^{\infty} \left(\frac{f}{1-f}\right)^t = \binom{n}{F} \cdot \frac{1-f}{1-2f}, \end{aligned}$$

so that  $C \geq \frac{1-2f}{1-f}$ . Now, we use the fact that  $\binom{n}{\alpha n} = 2^{H(\alpha)n+O(1)}/\sqrt{n}$ , where  $H(x) = -x \log_2 x - (1-x) \log_2(1-x)$  is the entropy function. We may therefore write  $B$  as

$$\begin{aligned} \frac{\binom{n_1}{F_1}}{\binom{n}{F}} &= 2^{H(f)(n_1-n)+O(1)} \cdot \frac{\sqrt{n}}{\sqrt{n_1}} \\ &\geq \beta \left( 2^{\frac{-4H(f)}{(1-2f)^2} \log \log n} \right) \\ &= \beta (\log n)^{-\frac{4H(f) \log 2}{(1-2f)^2}}, \end{aligned}$$

where  $\beta > 0$  is an absolute constant. Combining these bounds, we have

$$ABC \geq \frac{\beta}{2} \frac{1-2f}{1-f} (\log n)^{\frac{4 \log 2}{(1-2f)^2} (1-H(f))}.$$

It is easy to check that  $\frac{4 \log 2}{(1-2f)^2} (1-H(f)) > 2$  for all  $f \in (0, 1/2)$ , from which the desired bound follows.  $\square$

**Lemma 11.**

$$\log \binom{n}{\leq \lfloor fn \rfloor} = \Theta(n),$$

where the implicit constant depends on  $f$ .

*Proof.* This follows immediately from the estimate

$$\binom{n}{\lfloor fn \rfloor} = 2^{H(f)n+O(\log n)}$$

as in the proof of Lemma 10.  $\square$

The following result appears in [11].

**Theorem 12.** Let an urn contain  $R$  red balls and  $B$  black balls. Suppose each red ball has weight  $w_{\circ}$  and each black has weight  $w_{\bullet}$ . Suppose that the balls are selected one-by-one without replacement where each as yet unselected ball is given a probability of being selected at the next round that equals its current fraction of the total weight of all unselected balls. Suppose  $r$  and  $b$  satisfy  $r = R(1 - e^{-w_{\circ}\rho})$  and  $b = B(1 - e^{-w_{\bullet}\rho})$ , for some fixed  $\rho > 0$ . Let  $r + b$  balls be drawn from the urn as prescribed. Let  $X_{\circ}$  be the number of red balls selected by this random process, and let  $X_{\bullet}$  be the number of black, so that  $X_{\circ} + X_{\bullet} = r + b$ . Then  $r' = \lceil r \rceil$  or  $\lfloor r \rfloor$  and  $b' = \lceil b \rceil$  or  $\lfloor b \rfloor$  are the medians of  $X_{\circ}$  and  $X_{\bullet}$ , respectively.

By taking  $w_o = w_\bullet$ , i.e.,  $r/b = R/B$ , this result gives the median of the hypergeometric distribution. If we let  $r + b = T$  be the total number of balls drawn, then this gives  $b = BT/(R + B)$ , i.e., the mean of  $X_o$ . Hence, we have the following Corollary.

**Corollary 13.** If  $\mu$  is the mean and  $m$  the median of a hypergeometric distribution, then  $m = \lceil \mu \rceil$  or  $m = \lfloor \mu \rfloor$ .

**Proposition 14.** Let  $0 \leq r \leq fn_2$ . Suppose that  $fn_1 + r$  elements are drawn uniformly at random (without replacement) from a set  $S = S_1 \cup S_2$  with  $|S_1| = n_1$ ,  $|S_2| = n_2$ , and  $n = n_1 + n_2$ . Let  $X$  denote the number of such elements in  $S_2$ . If  $n_1, n_2 \rightarrow \infty$  and  $n_2 = o(n)$ , there is some function  $h : \mathbb{N} \rightarrow \mathbb{R}$  with  $h = \omega(1)$  so that, for  $fn_2 \geq r \geq fn_2 - h(n)$ , we have

$$\Pr(X \leq r) = 1/2 - o(1).$$

*Proof.*  $X$  follows a hypergeometric distribution with parameters  $n = n_1 + n_2$ ,  $n_2$ , and  $R = fn_1 + r$ . Its expectation is therefore given by  $\mu = \frac{n_2}{n}(fn_1 + r) = n_2R/n$ . Writing  $p(k)$  for the probability that  $X = k$ , note that

$$p(k) = \frac{\binom{n_2}{k} \binom{n_1}{R-k}}{\binom{n}{R}}.$$

When  $k = \mu + \Delta$ , we have

$$\begin{aligned} \frac{p(k)}{p(k-1)} &= \frac{\binom{n_2}{k} \binom{n-n_2}{R-k}}{\binom{n_2}{k-1} \binom{n-n_2}{R-k+1}} \\ &= \frac{(k-1)!(n_2-k+1)!(R-k+1)!(n-n_2-R+k-1)!}{k!(n_2-k)!(R-k)!(n-n_2-R+k)!} \\ &= \frac{(n_2-k+1)(R-k+1)}{k(n-n_2-R+k)} \\ &= \frac{(n_2 - \frac{Rn_2}{n} - \Delta + 1)(R - \frac{Rn_2}{n} - \Delta + 1)}{(\frac{Rn_2}{n} + \Delta)(n-n_2-R + \frac{Rn_2}{n} + \Delta)}. \end{aligned}$$

Then, since  $R = fn_1 + r = fn_1 + fn_2 - o(n) = fn(1 + o(1))$  and  $n_2 = o(n)$ ,

$$\begin{aligned} \left| \frac{p(k)}{p(k-1)} - 1 \right| &= \left| \frac{nR - 2n_2R - n^2\Delta + 2n\Delta + nn_2 + n}{(Rn_2 + \Delta n)(n - n_2 - R + Rn_2/n + \Delta)} \right| \\ &= \left| \frac{fn^2(1 + o(1)) - n^2\Delta(1 + o(1))}{n(fn_2(1 + o(1)) + \Delta)(n(1 - f + o(1)) + \Delta)} \right| \\ &= O\left(\frac{n(1 + \Delta)}{(n_2 + \Delta)(n + \Delta)}\right) = o(1), \end{aligned}$$

as long as  $\Delta = o(n_2)$ . Therefore, as  $n \rightarrow \infty$ , the number of  $k$ 's so that  $p(k)$  is within  $1 + o(1)$  of  $p(\mu)$  grows without bound. This implies that  $p(k) = O(1/g(n))$  for some function



$g : \mathbb{N} \rightarrow \mathbb{R}$  with  $g = \omega(1)$  and all  $k$ . If we let  $h(n) = \sqrt{g(n)}$ , the total probability that  $r \leq X \leq \mu$  is  $O(1/\sqrt{g(n)}) = o(1)$ . Since, by Corollary 13,  $\lceil \mu \rceil$  or  $\lfloor \mu \rfloor$  is the median  $m$  of the hypergeometric distribution, this implies that  $\Pr(X \leq r) \geq 1/2 + o(1)$ . On the other hand, since  $\mu \geq r$ ,

$$\Pr(X \leq r) \leq \Pr(X \leq \mu) = \Pr(X \leq m) + o(1) = 1/2 + o(1).$$

□

**Proposition 15.** For  $n$  tending to infinity and a fixed  $f \in (0, 1/2)$ ,

$$\sum_{k=F_1}^F \sum_{s=F_1}^k \binom{n_1}{s} \binom{n_2}{k-s} = \left(\frac{1}{2} - o(1)\right) \sum_{k=0}^F \binom{n}{k}$$

*Proof.* Let  $n_3 = \lceil h(n)/2 \rceil$ , where  $n = n_1 + n_2$  and  $h(n)$  is the function given by Proposition 14. Then we have

$$\frac{\sum_{s=F_1}^k \binom{n_1}{s} \binom{n_2}{k-s}}{\binom{n}{k}} = \frac{1}{2} - o(1),$$

for  $k \in [F - n_3, F]$  since the left-hand quantity represents the probability, if a set of  $k = F_1 + r$  elements is drawn uniformly at random,  $F_2 - n_3 \leq r \leq F_2$ , that at most  $r$  of the points will be taken from the last  $n_2$  of all  $n = n_1 + n_2$  elements. Therefore, by the above and then by Lemma 9,

$$\begin{aligned} \sum_{k=F_1}^F \sum_{s=F_1}^k \binom{n_1}{s} \binom{n_2}{k-s} &= \sum_{k=F-n_3}^F \sum_{s=F_1}^k \binom{n_1}{s} \binom{n_2}{k-s} \\ &\quad + \sum_{k=F_1}^{F-n_3-1} \sum_{s=F_1}^k \binom{n_1}{s} \binom{n_2}{k-s} \\ &= \left(\frac{1}{2} - o(1)\right) \sum_{k=F-n_3}^F \binom{n}{k} \\ &= \left(\frac{1}{2} - o(1)\right) (1 - o(1)) \sum_{k=0}^F \binom{n}{k} \\ &= \left(\frac{1}{2} - o(1)\right) \sum_{k=0}^F \binom{n}{k}. \end{aligned}$$

□

## 5 Reduction from liar machine to the pathological liar game

We now consider the alternating-question strategy for Paul, and show in Theorem 24 that Carole has no better response strategy than always assigning a lie to each of the odd-numbered chips. The time-evolution of the chips under these question-and-response strategies is equivalent to that of the liar machine, as described in the proof of Cor. 25. We then combine results of the previous sections to prove Theorem 4 on parameters for which Paul can win the  $((M, 0, \dots, 0), n, e)_2^*$ -game.

**Definition 16** (Position vector). Given the state vector  $x = (x(0), \dots, x(e))$  of a liar game with  $M$  elements, the *position vector*  $u = u(x) = (u(1), u(2), \dots, u(M))$  corresponding to  $x$  is defined by

$$u(j) := \min \left\{ k : \sum_{i=0}^k x(i) \geq j \right\}.$$

**Example 17.** The position vector of a state vector essentially labels the  $M$  elements tracked by the state vector from left to right, and records as  $u(j)$  the number of lies associated with the  $j$ th element.

$$\begin{aligned} \text{If } x &= (2, 0, 1, 3, 0), \text{ then} \\ u = u(x) &= (0, 0, 2, 3, 3, 3). \end{aligned}$$

Position vectors are monotonic increasing, and provided the maximum number of lies is available (from context, for example), the state vector can be recovered from the position vector. We analyze the round-by-round evolution of state vectors by comparing their corresponding position vectors under the weak majorization partial order, presented for analysis of the original liar game by [12].

**Definition 18** (Partial order on position vectors). Let  $M \in \mathbb{Z}^+$ , and let

$$U = \{(u(1), \dots, u(M)) \in \mathbb{N}^M : u(1) \leq \dots \leq u(M)\}$$

be the set of position vectors with  $M$  entries. For  $u, v \in U$ , we define the partial order  $u \leq v$  provided for all  $1 \leq k \leq M$ ,  $\sum_{j=1}^k u(j) \leq \sum_{j=1}^k v(j)$ .

**Example 19.** The partial order on position vectors gives  $(0, 2, 2) \leq (1, 1, 2) \leq (1, 2, 2) \leq (2, 2, 2)$ .

In order to analyze position vectors within the partial order, it will be convenient to continue tracking disqualified elements, with position at least  $e + 1$ , in the position vector. We do this with the understanding that disqualified elements are dropped when converting back to the state vector. The alternating question for Paul puts all elements tracked by an even (odd) index in the position vector  $u$  into  $A_0$  ( $A_1$ ). The number of lies associated with each element is easily read from the position vector. Carole's response either assigns an additional lie to the elements indexed by the odd positions, to obtain the new position vector  $\text{ODD}(u)$ , or assigns an additional lie to the elements indexed by the even positions, to obtain the new position vector  $\text{EVEN}(u)$ .

**Definition 20** ( $\text{ODD}(u)$  and  $\text{EVEN}(u)$ ). Given the position vector  $u = (u(1), \dots, u(M))$ , define the position vector  $\text{ODD}(u)$  to be the result of sorting  $(u(1)+1, u(2), u(3)+1, u(4), \dots, u(M)+(M \bmod 2))$  in nondecreasing order, and define the position vector  $\text{EVEN}(u)$  to be the result of sorting  $(u(1), u(2)+1, u(3), u(4)+1, \dots, u(M)+(M+1 \bmod 2))$  in nondecreasing order.

The following two properties appear in the proof of Lemma 2 of [12]. There is a minor error in the proof of the second property which we describe and correct after stating the lemma.

**Lemma 21.** Let  $u$  and  $v$  be position vectors of liar games with the same number of elements on the binary symmetric channel. Then

- (1)  $\text{EVEN}(u) \leq \text{ODD}(u)$ , and
- (2) If  $u \leq v$ , then  $\text{EVEN}(u) \leq \text{EVEN}(v)$ .

The proof of (1) is a straightforward verification. We defer the proof of (2) until after describing how to transform  $u$  into  $v$  in manageable steps. Close inspection will reveal that the proof in [12] does not find a transformation from  $u = (0, 1, 2)$  to  $v = (1, 1, 1)$ ; a successful procedure is as follows.

**Algorithm 22.** (Position vector transformation.)

Input: Position vectors  $u = (u(1), \dots, u(M))$  and  $v = (v(1), \dots, v(M))$  with  $u < v$ .

Output: A position vector  $u'$  with  $u < u' \leq v$ .

0. Initialize  $u' = u$ .
1. If  $\sum_{i=1}^M u(i) < \sum_{i=1}^M v(i)$ , then set  $u'(M) = u(M) + \sum_{i=1}^M v(i) - \sum_{i=1}^M u(i)$ .
2. Otherwise, if  $\sum_{i=1}^M u(i) = \sum_{i=1}^M v(i)$ :
  - 2a. Maximize  $j$  such that  $u(j) < v(j)$ .
  - 2b. Minimize  $k > j$  such that  $u(k) > v(k)$ .
  - 2c. Set  $u'(j) = u(j) + 1$  and  $u'(k) = u(k) - 1$ .

(By design of  $j$  and  $k$ ,  $u(j) < v(j)$ ,  $u(j+1) = v(j+1)$ ,  $\dots$ ,  $u(k-1) = v(k-1)$ ,  $u(k) > v(k)$ . Furthermore,  $u'$  is already in nondecreasing order.)

*Proof.* The algorithm is easy to verify for  $u'$  produced by Step 1. Suppose Step 2 is executed. Step 2a certainly produces a maximum  $j$ :  $u < v$  implies that  $\sum_{i=1}^{\ell} u(i) < \sum_{i=1}^{\ell} v(i)$  for some  $\ell$ , and so at least one choice for  $j$  with  $u(j) < v(j)$  exists. Step 2b produces a minimum  $k$ : using the  $j$  from Step 2a and combining the inequalities  $\sum_{i=1}^{j-1} u(i) \leq \sum_{i=1}^{j-1} v(i)$ ,  $u(j) < v(j)$ , and  $\sum_{i=1}^M u(i) = \sum_{i=1}^M v(i)$  yields  $\sum_{i=j+1}^M u(i) > \sum_{i=j+1}^M v(i)$ ; and so there is at least one choice of  $k$  for which  $u(k) > v(k)$ . For all indices  $i$  strictly between  $j$  and  $k$ ,  $u(i) < v(i)$  is impossible by maximality of  $j$ , and  $u(i) > v(i)$  is impossible by minimality of  $k$ . The middle entries of  $u$  and  $v$  are as follows:

$$u(j) < v(j), u(j+1) = v(j+1), \dots, u(k-1) = v(k-1), u(k) > v(k). \quad (5)$$

It remains to verify that  $u < u' \leq v$  for  $u'$  constructed in Step 2c. Already  $u'$  is in nondecreasing order, by definition of  $u'$ , inspection of (5), and noting that  $u(j) < u'(j) \leq$

$v(j)$  and  $v(k) \leq u'(k) < u(k)$ . Furthermore, for  $1 \leq \ell \leq j-1$ ,  $\sum_{i=1}^{\ell} u(i) = \sum_{i=1}^{\ell} u'(i) \leq \sum_{i=1}^{\ell} v(i)$ . With  $u(j)+1 = u'(j) \leq v(j)$ , we have  $1 + \sum_{i=1}^j u(i) = \sum_{i=1}^j u'(i) \leq \sum_{i=1}^j v(i)$ . Since  $u(i) = u'(i) = v(i)$  for all  $j+1 \leq i \leq k-1$ , we have  $1 + \sum_{i=1}^{\ell} u(i) = \sum_{i=1}^{\ell} u'(i) \leq \sum_{i=1}^{\ell} v(i)$  for all  $j+1 \leq \ell \leq k-1$ . With  $v(k) \leq u'(k) = u(k) - 1$ , we have  $\sum_{i=1}^k u(i) = \sum_{i=1}^k u'(i) \leq \sum_{i=1}^k v(i)$ . Since  $u \leq v$  and  $u'(i) = u(i)$  for  $i > k$ ,  $\sum_{i=1}^{\ell} u(i) = \sum_{i=1}^{\ell} u'(i) \leq \sum_{i=1}^{\ell} v(i)$  for  $k+1 \leq \ell \leq M$ .  $\square$

*Proof of Lemma 21 Part (2).* Iterative application of Algorithm 22 produces a sequence of position vectors  $u = u_0 < u_1 < \dots < u_t = v$ . The sequence terminates because there are a bounded number of position vectors satisfying the precondition  $\sum_{i=1}^M u(i) = \sum_{i=1}^M v(i)$  to execute Step 2 of the algorithm. Now let  $0 \leq s < t$  and consider  $u_s < u_{s+1}$ . If  $u_{s+1}$  was created by applying Step 1 of the algorithm to  $u_s$  (thereby forcing  $s = 0$ ), then  $\text{EVEN}(u_s) \leq \text{EVEN}(u_{s+1})$  is easy to verify.

Otherwise Step 2 created  $u_{s+1}$  from  $u_s$ . Inspection of (5) reveals that  $u_s(j) < u_{s+1}(j) = u_s(j) + 1 \leq u_{s+1}(k) = u_s(k) - 1 < u_s(k)$ . Ignoring for a moment the  $j$ th and  $k$ th entries of  $u_s$  and  $u_{s+1}$ , and applying  $\text{EVEN}$  to all other entries and then resorting, we have the following identical structure for  $\text{EVEN}(u_s)$  and  $\text{EVEN}(u_{s+1})$ :

$$\dots \leq u_s(j) + \chi_{2|j} \mid \geq u_s(j) + 1 + \chi_{2|j} \quad \dots \leq u_s(k) - 1 + \chi_{2|k} \mid \geq u_s(k) + \chi_{2|k} \quad \dots$$

Here,  $\chi_{2|j}$  ( $\chi_{2|k}$ ) equals 1 if 2 divides  $j$  ( $k$ ) and equals 0 otherwise; the vertical separators denote that smaller entries lie to the left and larger to the right. Now we can see that  $\text{EVEN}(u_s)$  is the same as inserting  $u_s(j) + \chi_{2|j}$  and  $u_s(k) + \chi_{2|k}$  from left to right at the two separators without need for resorting. Similarly,  $\text{EVEN}(u_{s+1})$  is the same as inserting  $u_s(j) + 1 + \chi_{2|j}$  and  $u_s(k) - 1 + \chi_{2|k}$  from left to right at the two separators without need for resorting. With this observation it is simple to verify that  $\text{EVEN}(u_s) < \text{EVEN}(u_{s+1})$ .

Since  $s$  was arbitrary in the preceding argument, we have  $\text{EVEN}(u) = \text{EVEN}(u_0) < \text{EVEN}(u_1) < \dots < \text{EVEN}(u_t) = \text{EVEN}(v)$ , and so combined with the case  $t = 0$  for which  $\text{EVEN}(u) = \text{EVEN}(v)$ , Part (2) of the lemma holds.  $\square$

**Corollary 23.** Let  $u$  and  $v$  be position vectors of liar games with the same number of elements on the binary symmetric channel. If  $u \leq v$ , then  $\text{ODD}(u) \leq \text{ODD}(v)$ .

*Proof.* We use a trick to piggyback on Lemma 21 Part (2). Set  $u' = (-2, u(1), \dots, u(M))$  and  $v' = (-2, v(1), \dots, v(M))$  and observe that  $u \leq v$  implies  $u' \leq v'$ . The first entry of  $u'$  and of  $v'$  is sufficiently separated, and so  $\text{EVEN}(u') = (-2, \text{ODD}(u))$  and  $\text{EVEN}(v') = (-2, \text{ODD}(v))$ . Applying Lemma 21 to  $u'$  and  $v'$  yields  $\text{EVEN}(u') \leq \text{ODD}(v')$ . As  $\text{EVEN}(u')(1) = \text{EVEN}(v')(1) = -2$ , this forces  $\text{ODD}(u) \leq \text{ODD}(v)$ .  $\square$

Next we show that when Paul's strategy is to always ask the alternating question, Carole's best possible response strategy in the pathological liar game is to move the odd-numbered elements. This will provide an upper bound on the minimum number of elements required for Paul to have a winning strategy in the  $(x, n, e)_2^*$ -game.

**Theorem 24.** Let  $x$  be an initial state vector, and  $n, e \in \mathbb{N}$ . Assume that Paul always asks the alternating question. In the  $(x, n, e)_2^*$ -game, Carole's best strategy is to move the odd-numbered elements.

*Proof.* Let  $u_s$  be the position vector after  $s$  rounds of the game, where  $u_0$  is the position vector corresponding to the initial state vector  $x$ . Carole wins the  $(x, n, e)_2^*$ -game iff  $u_n(1) > e$ . Consider the  $2^n$  leaves of the strategy tree of the game determined by every possible length  $n$  sequence of choices for Carole to select  $\text{ODD}(u_s)$  or  $\text{EVEN}(u_s)$  to complete round  $s + 1$ . Thus  $\text{ODD}^n(u_0)$  is the leaf corresponding to Carole always moving the odd elements. It suffices to show that  $\text{ODD}^n(u_0) \geq v$  for all other leaves  $v$  of the strategy tree. We prove this by induction on  $n$ . The base case  $n = 1$  is provided by Lemma 21 Part (1). Now let  $0 < s < n$ , assume that  $v$  is a position vector after  $s$  rounds, and assume that  $v \leq \text{ODD}^s(u_0)$ . By Corollary 23,  $\text{ODD}(v) \leq \text{ODD}(\text{ODD}^s(u_0)) = \text{ODD}^{s+1}(u_0)$ , and by Lemma 21 Part (1) and transitivity,  $\text{EVEN}(v) \leq \text{ODD}^{s+1}(v)$ . All position vectors after  $s + 1$  rounds are obtained by applying  $\text{ODD}$  or  $\text{EVEN}$  to a position vector after  $s$  rounds, and so the induction succeeds.  $\square$

By a simple transformation, Carole's odd response strategy is equivalent to the time-evolution of the liar machine.

**Corollary 25.** Let the liar machine have initial configuration  $f_0$  with  $M$  chips at the origin and none elsewhere. If  $\sum_{i=-n}^{-n+2e} f_n(i) \geq 1$ , then Paul can win the  $((M, 0, \dots, 0), n, e)_2^*$ -game.

*Proof.* Let  $u_s$  and  $x_s$  be the position and state vectors, respectively at the end of round  $s$ , of the  $((M, 0, \dots, 0), n, e)_2^*$ -game in which Paul always asks the alternating question, and Carole always chooses  $u_{s+1} = \text{ODD}(u_s)$ . By Theorem 24, we need only transform  $\text{ODD}^s(u_0)$  into  $f_s$ , where  $u_0$  is the position vector corresponding to the initial state vector  $(M, 0, \dots, 0)$ . By definition of  $\text{ODD}(u)$  and of one step of the liar machine, this is accomplished by observing that  $x_s(i) = f_s(-s + 2i)$  for all  $0 \leq i \leq s$ . Consequently  $\text{ODD}^n(u_0)(1) \leq e$  iff  $\sum_{i=0}^e f_n(-n + 2i) \geq 1$ .  $\square$

The converse is not true. For some games the alternating question strategy is not optimal, so that Paul has a winning strategy, but  $\text{ODD}^n(u_0(1)) > e$ . For example, Paul can win the  $((1, 11), 4, 1)_2^*$ -game (as the reader can readily verify – the first question is  $(1, 4)$ ), but the progression of configurations given by the liar machine is  $(1, 11) \rightarrow (0, 7) \rightarrow (0, 3) \rightarrow (0, 1) \rightarrow (0, 0)$ .

We again use the following notation. Let  $n \rightarrow \infty$ , fix  $f \in (0, 1/2)$ , and set  $n_1 = n - \left\lfloor \frac{4}{(1-2f)^2} \log \log n \right\rfloor$  and  $n_2 = n - n_1$ . Define  $F = \lfloor fn \rfloor$ ,  $F_1 = \lfloor fn_1 \rfloor$ , and  $F_2 = F - F_1$ .

**Theorem 26.** Let  $n, M \in \mathbb{Z}^+$ . Let  $f_0 : \mathbb{Z} \rightarrow \mathbb{N}$  be the initial configuration of the liar machine defined by  $f_0(0) = M$ , and  $f_0(j) = 0$  otherwise. For  $n$  sufficiently large, if

$$M \geq \frac{2^n}{\binom{n}{\leq F}} (2 + o(1)) c' \sqrt{n_2},$$

where  $c'$  is the constant from Theorem 3, then  $\sum_{i=F_1}^F f_n(-n + 2i) \geq 1$ .

*Proof.* Set  $g_0 = f_0$  and let  $g_s$  be the chip distribution in the linear machine after  $s$  rounds. Then for  $F_1 \leq j \leq F$ , the number of chips at position  $-n_1 + 2j$  in the linear machine after  $n_1$  rounds is

$$g_{n_1}(-n_1 + 2j) = \frac{\binom{n_1}{j}}{2^{n_1}} \frac{2^n}{\binom{n}{\leq F}} (2 + o(1)) c' \sqrt{n_2}. \quad (6)$$

Since  $F < n_1/2$  for  $n$  sufficiently large, the minimum occurs at  $j = F_1$ , and is  $\omega(\log n)$  by Lemma 10. Applying Theorem 2, for  $F_1 \leq j \leq F$  we have

$$f_{n_1}(-n_1 + 2j) \geq \frac{\binom{n_1}{j}}{2^{n_1}} \frac{2^n}{\binom{n}{\leq F}} (2 + o(1)) c' \sqrt{n_2}. \quad (7)$$

Now for  $F_1 \leq j \leq F$ , define  $h_{n_1}(-n_1 + 2j) = f_{n_1}(-n_1 + 2j)$ , and define  $h_{n_1}(j) = 0$  elsewhere. Thus  $h_{n_1}$  is obtained from  $f_{n_1}$  by removing chips outside of the interval  $[-n_1 + 2F_1, -n_1 + 2F]$ . We run the linear machine with initial state  $h_{n_1}$  for  $n_2$  rounds to produce  $h_n$ , and obtain for  $F_1 \leq i \leq F$  that

$$h_n(-n + 2i) \geq \sum_{j=F_1}^i \frac{\binom{n_1}{j}}{2^{n_1}} \frac{2^n}{\binom{n}{\leq F}} (2 + o(1)) c' \sqrt{n_2} \frac{\binom{n_2}{i-j}}{2^{n_2}},$$

as for  $i$  and  $j$  fixed, the contribution to  $h_n(-n + 2i)$  from  $h_{n_1}(-n_1 + 2j)$  is  $h_{n_1}(-n_1 + 2j) \binom{n_2}{i-j} / 2^{n_2}$ . Summing  $h_n(-n + 2i)$  over  $i$  and applying Proposition 15,

$$\sum_{i=F_1}^F h_n(-n + 2i) \geq c' \sqrt{n_2} (1 + o(1)).$$

Noting that  $\sqrt{n_2} = o(F - F_1)$  and applying Theorem 3 to  $h_{n_1}$ , we obtain  $\sum_{i=F_1}^F f_n(-n + 2i) \geq 1$  as desired.  $\square$

*Proof of Theorem 4.* Corollary 25 reduces the  $((M, 0, \dots, 0), n, e)_2^*$ -game to the liar machine with winning condition  $\sum_{i=-n}^{-n+2e} f_n(i) \geq 1$ , which Theorem 26 shows is satisfied for the given form of  $M$ .  $\square$

## 6 Concluding remarks and acknowledgements

The major open question is whether the time-evolution of the liar machine with  $M$  elements at the origin and zero elsewhere can be given in closed form, or at least whether the leftmost chip can be tracked more tightly. Either case would yield an improvement by decreasing the minimum  $M$  for which Paul can win the  $((M, 0, \dots, 0), n, e)_2^*$ -game. We suppose that the best hope is for the optimal  $M$  to be asymptotically a constant multiple above the sphere bound. Similarly, by the reduction in [12] from the  $((M, 0, \dots, 0), n, e)_2$ -game (original liar game) to the linear machine, improved tracking of the leftmost chip could provide an

alternative proof of Theorem 3 of [14], which is equivalent to a lower bound on  $M$  for which Paul can win the original liar game. Optimistically, the bound in [14] on  $M$  might be improved to a constant multiple below the sphere bound.

We thank Joel Spencer for discussions that helped to crystallize the ideas for this paper—with the first author during an extended collaboration on deterministic random walks, and with the second author at a conference in 2004 on alternate viewpoints for the liar game. We also thank the anonymous referee for various helpful comments.

## References

- [1] Elwyn R. Berlekamp. *Block coding with noiseless feedback*. PhD thesis, Massachusetts Institute of Technology, 1964.
- [2] Gérard Cohen, Iiro Honkala, Simon Litsyn, and Antoine Lobstein. *Covering codes*. North-Holland Mathematical Library, **54**. North-Holland Publishing Co., Amsterdam, 1997.
- [3] Joshua Cooper, Benjamin Doerr, Joel Spencer, and Gábor Tardos. Deterministic random walks on the integers. *European J. Combin.*, 28(8):2072–2090, 2007.
- [4] Philippe Delsarte and Philippe Piret. Do most binary linear codes achieve the Gobblick bound on the covering radius? *IEEE Trans. Inform. Theory*, 32(6):826–828, 1986.
- [5] Robert B. Ellis and Kathryn L. Nyman. Two-batch liar games on a general bounded channel. *J. Combin. Theory Ser. A*, 116(8):1253–1270, 2009.
- [6] Robert B. Ellis and Catherine H. Yan. Ulam’s pathological liar game with one half-lie. *Int. J. Math. Math. Sci.*, (29-32):1523–1532, 2004.
- [7] Robert B. Ellis, Vadim Ponomarenko, and Catherine H. Yan. The Rényi-Ulam pathological liar game with a fixed number of lies. *J. Combin. Theory Ser. A*, 112(2):328–336, 2005.
- [8] Robert B. Ellis, Vadim Ponomarenko, and Catherine H. Yan. The Rényi-Ulam pathological liar game with a fixed number of lies. Preprint, [arXiv:math/0407504](https://arxiv.org/abs/math/0407504) [[math.CO](https://arxiv.org/abs/math/0407504)], 2004.
- [9] Gregory F. Lawler. *Intersections of random walks*. Birkhäuser Boston, Inc., Boston, MA, 1991.
- [10] Alfréd Rényi. On a problem in information theory. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 6:505–516 (1962), 1961.
- [11] Alan Siegel, Median bounds and their application. *Tenth Annual ACM-SIAM Symposium on Discrete Algorithms* (Baltimore, MD, 1999). *J. Algorithms* **38** (2001), no. 1, 184–236.

- [12] Joel Spencer and Peter Winkler. Three thresholds for a liar. *Combin. Probab. Comput.*, 1(1):81–93, 1992.
- [13] Stanisław M. Ulam. *Adventures of a mathematician*. Charles Scribner's Sons, New York, 1976.
- [14] Kamil Sh. Zigangirov. On the number of correctable errors for transmission over a binary symmetrical channel with feedback. *Probl. Peredachi Inf.*, 12(2):3–19, 1976. English translation in *Probl. Inf. Trans.* 12(2):85–97, 1976.