

On reliable computation by noisy random Boolean formulas

Alexander Mozeika and David Saad

Abstract—We study noisy computation in randomly generated k -ary Boolean formulas. We establish bounds on the noise level above which the results of computation by random formulas are not reliable. This bound is saturated by formulas constructed from a single majority-like gates. We show that these gates can be used to compute any Boolean function reliably below the noise bound.

Index Terms—Random Boolean formulas, ϵ -noise, reliable computation.

I. INTRODUCTION

One of computation models for a Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ is a Boolean circuit or formula [1]. A *circuit* is a directed acyclic graph in which nodes of in-degree zero are either the Boolean constants or variables, nodes of in-degree $k \geq 1$ are logical gates, computing Boolean functions of k arguments, and nodes of out-degree zero correspond to the circuit outputs. If a circuit has only a single output and the output of each gate is used as an input to at most one gate then this circuit is called a *formula*. In circuits, as in any other

model of computation, the computational complexity and effects of noise are important questions [2].

The circuit complexity of a Boolean function is the minimum number of gates (circuit *size*) or the minimum *depth*¹ of a circuit, constructed from a particular set of gates, which computes this function. However, to find a circuit representation of a Boolean function with a bounded size or depth is a difficult problem [1]. One approach to this problem is to study complexity of *typical* Boolean functions computed by random formulas [3].

The two most studied methods of generating random formulas use random tree generation and a growth process as their core procedures. In the first method, a rooted k -ary tree is sampled from the uniform distribution of all rooted k -ary trees; the leaves of this random tree are then labelled by reference to the Boolean variables and internal nodes are labelled by the Boolean gates. This method was used to investigate the complexity of typical functions computed by random AND/OR formulas [4], [5], [6] and allowed to obtain a close relation between the probability $P[f]$ of a random formula to compute a Boolean function f and its size (complexity). However, it seems that this probability distribution is biased towards very low complexity functions [5].

¹The depth of a circuit is the number of gates on the longest path from an input node to the output node

Alexander Mozeika is currently with the Department of Information and Computer Science, Aalto University, PO Box 15400, 00076 Aalto, Finland. e-mail: alexander.mozeika@aalto.fi; the work had been carried out when he was with the Non-linearity and Complexity Research Group, Aston University

David Saad is with the Non-linearity and Complexity Research Group, Aston University, Birmingham, B4 7ET, United Kingdom.

Manuscript received June X, 2012; revised June X, 2012.

The second method uses the following growth process: Firstly, one defines the initial probability distribution $P_0[f]$ over the set F^0 of simple Boolean functions of N variables. Secondly, and in further steps, the functions chosen from the distributions $P_t[f]$ defined in previous steps are combined by Boolean gates: $F^{t+1} = \{\alpha(f_1, \dots, f_k); f_j \in F^t \text{ for } j = 1, 2, \dots, k\}$. This process can be seen as a growth of k -ary balanced trees and was first used by Valiant to obtain an upper bound on the size of monotone formulas computing the majority function [7]. Savický recently showed for one of these processes, for $P_0[f]$ that is uniform on some set of Boolean functions F^0 and under very broad conditions on α , the probability $P_t[f]$ tends to the uniform distribution over all Boolean functions of N variables when $t \rightarrow \infty$ [8]. The convergence rates of the Savický's process and its variants with different gates and initial conditions were studied in [3].

Another important question in the circuit theory is a reliable computations of Boolean functions in the presence of noise. One of the first to study the effect of noise in computing systems was von Neumann who attempted to explain the robustness of biologically-inspired computing circuits [9]. His model represented neural activities by a circuit (or formula) composed of ϵ -noisy Boolean gates. The ϵ -noisy gate is designed to compute a Boolean function $\alpha : \{-1, 1\}^k \rightarrow \{-1, 1\}$, but for each input $\sigma \in \{-1, 1\}^k$ there is an error probability ϵ such that $\alpha(\sigma) \rightarrow -\alpha(\sigma)$. To simplify the analysis, error-probability is taken to be independent for each gate in the circuit. Clearly, a noisy circuit ($\epsilon > 0$) cannot perform any given computation in a deterministic manner: for any circuit-input there is a non-vanishing probability that the circuit will produce the wrong output. The maximum of this error probability δ over all circuit-inputs determines the *reliability* of the

circuit. In his paper, von Neumann showed that reliable computation ($\delta < 1/2$) is possible for a sufficiently small ϵ [9] and demonstrated how reliability of a Boolean noisy circuit can be improved by using constructions based only on ϵ -noisy gates.

There had been little development in the analysis of noisy computing systems until the seminal work of Pippenger [10] who addressed the problem from an information theory point of view. He showed that if a noisy k -ary formula is used to compute a Boolean function f with the error probability $\delta < 1/2$, then (i) there is an upper bound for the gate-error $\epsilon(k)$ which is strictly less than $1/2$ and (ii) there is a lower bound for the formula-depth $\hat{d}(k, \epsilon, \delta) \geq d$, where d is the depth of a noiseless formula computing f . In comparison to its noiseless counterpart, a noisy formula that computes reliably has greater depth due to the presence of restitution-gates, implying longer computation times [10].

A number of papers have followed and extended Pippenger's results. For instance, similar results were derived for circuits by Feder [11], who also improved the bounds obtained by Pippenger for formulas. The exact noise thresholds for k -ary Boolean formulae were later determined for odd k [12], [13] and for formulas constructed from 2-input NAND gates [14]; the latter was recently suggested as the exact noise threshold for general 2-input gate formulas [15].

Results derived for noisy Boolean formulas in [12], [13] rely on a specific construction which uses ϵ -noisy *majority* gates. The noiseless variant of this gate performs the majority-vote function² $\text{sgn}[\sum_{i=1}^k S_j]$ on the binary inputs $S_j \in \{-1, 1\}$ and naturally the number of these inputs k is *odd*. In contrast to previous work, in

²We use the definition $\text{sgn}[x] = 1$ for $x > 0$, $\text{sgn}[x] = -1$ for $x < 0$ and $\text{sgn}[0] = 0$ throughout this paper.

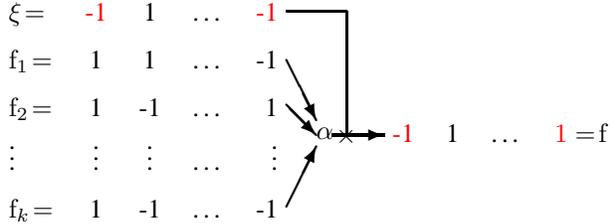


Fig. 1. Noisy growth process. i) Boolean functions f_1, \dots, f_k (represented by binary strings of length 2^N) are sampled randomly and independently from the distribution $P_t[f]$. ii) These functions are then used to compute a new Boolean function f via the gate α . At each step of this computation noise (represented by the binary string ξ) inverts the output of α (this operation is represented by the \times symbol) with probability ϵ . In this figure the first and the last bits of the function f (in red) are inverted by noise. Repeating operations i) and ii) many times gives rise to an ensemble described by the distribution $P_{t+1}[f]$.

this paper we concentrate on the possibility of reliable computation in *randomly generated* Boolean formulas. As a first step towards this goal, we study the effects of ϵ -noise on the formulas generated in the Savický's growth process.

II. NOISY GROWTH PROCESSES AND MAIN RESULTS

Let us introduce noise into the formulas generated by Savický's growth process. In order to do this we note that the noiseless case, as described in the Introduction, can be also seen as a computation, performed by gate α , of a new Boolean function f from k Boolean functions f_1, \dots, f_k . These functions, represented by binary ± 1 strings (or vectors) of length 2^N , are drawn randomly and independently from the same distribution [8]. However, each computation at the gate α may be corrupted by noise that inverts the result of this computation with probability ϵ (see Figure 1). Averaging the process over

many realizations of noise leads us to the equation

$$P_{t+1}[f] = \sum_{\{f_j\}} \prod_{j=1}^k \{P_t[f_j]\} \times \prod_{i=1}^{2^N} \frac{e^{\beta f^i \alpha(f_1^i, \dots, f_k^i)}}{2 \cosh \beta \alpha(f_1^i, \dots, f_k^i)} \quad (1)$$

which gives us the probability of a Boolean function f being computed by the noisy formulas of depth $t + 1$. Here for convenience we have introduced the inverse “temperature” parameter $\beta = 1/T$ which is related to the noise parameter ϵ via the equality $\tanh \beta = 1 - 2\epsilon$. The limits $\beta \rightarrow 0/\infty$ correspond to completely random/deterministic cases.

Without noise ($\beta \rightarrow \infty$) the equation (1) reduces to

$$P_{t+1}[f] = \sum_{\{f_j\}} \prod_{j=1}^k \{P_t[f_j]\} \times \prod_{i=1}^{2^N} \delta[f^i; \alpha(f_1^i, \dots, f_k^i)], \quad (2)$$

where we use $\delta[x; y]$ to denote Kronecker delta. Equation (2) was studied in the original Savický's work [8] and subsequent studies [3] where the stationary distribution $P_\infty[f] = \lim_{t \rightarrow \infty} P_t[f]$ of the noiseless process (2) was studied with the initial conditions $P_0[f] = \frac{1}{|F^0|} \sum_{g \in F^0} \prod_{i=1}^{2^N} \delta[f^i; g^i]$ for different initial sets F^0 of simple Boolean functions (constants, identities, etc.) and different gates α . Depending on these parameters the stationary distribution is either concentrated on a single function, i.e. $P_\infty[f] = \prod_{i=1}^{2^N} \delta[f^i; g^i]$ or on some set of functions F , i.e. $P_\infty[f] = \frac{1}{|F|} \sum_{g \in F} \prod_{i=1}^{2^N} \delta[f^i; g^i]$. There are also cases when for $t \rightarrow \infty$ the distributions $P_t[f]$ and $P_{t+1}[f]$ are distinct.

Our main contribution to these studies is the following result for the noisy process (1).

Theorem 2.1: For a *balanced* gate³ α the stationary

³The gate is balanced when it has an equal number of +1's and -1's in its output.

distribution $P_\infty[f] = \frac{1}{2^{2^N}}$ is a stable and unique solution of the process (1) when $\epsilon > \epsilon(k) = \frac{1-b(k)}{2}$, where $b(k) \equiv \left\{ 2^{k-1}/k \binom{k-1}{(k-1)/2}; 2^{k-2}/(k-1) \binom{k-2}{(k-2)/2} \right\}$, with $k \geq 3$, for k odd and even respectively.

Proof: In order to show this, we first use the result, derived in Appendix A, that the distribution $P_{t+1}[f]$ can be represented via its moments $m^S(t+1) = \sum_{\hat{f}} P_{t+1}[\hat{f}] \prod_{i \in S} \hat{f}^i$, where S is a subset of the set $[2^N] = \{1, \dots, 2^N\}$, and $P_{t+1}[f]$ is given by

$$P_{t+1}[f] = \frac{1}{2^{2^N}} \left(1 + \sum_{S \subseteq [2^N]} m^S(t+1) \prod_{i \in S} f^i \right), \quad (3)$$

where the n -th moment is governed by the equation

$$\begin{aligned} & m^{i_1, \dots, i_n}(t+1) \\ &= \tanh^n(\beta) \sum_{\{f_j^i\}} \prod_{j=1}^k \left\{ \right. \\ & \quad \times \frac{1}{2^n} \left[1 + \sum_{S \subseteq \mathbf{I}} m^S(t) \prod_{i \in S} f_j^i \right] \left. \right\} \\ & \quad \times \prod_{i \in \mathbf{I}} \alpha(f_1^i, \dots, f_k^i) \end{aligned} \quad (4)$$

with $\mathbf{I} = \{i_1, \dots, i_n\}$. Thus the n -th moment at $t+1$ is a function of only the n -th and lower order moments at t .

Let us assume now that $\tanh(\beta) < b(k)$ then by the Lemma 4.1, with $n = 1$, the first moments of the distribution $P_{t+1}[f]$ are vanishing as $t \rightarrow \infty$. But then, by applying the same lemma to the order $n \geq 2$ moments, we conclude that all moments are vanishing as $t \rightarrow \infty$. ■

In addition to its direct interpretation that above $\epsilon(k)$ (see Figure 2) the noisy process (1) is *ergodic* and has only one stationary solution, the result of Theorem 2.1 also has consequences for computation in noisy random formulas. A feature of noisy formulas, which is essential for reliable computation, is their greater depth due to the presence of correcting ϵ -noisy gates [10]. This correction

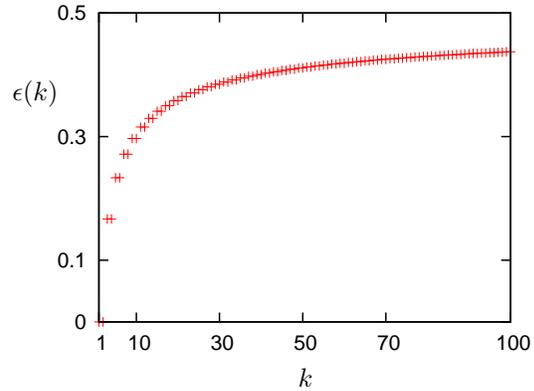


Fig. 2. Upper bound for reliable computation by noisy k -ary random formulas.

operation can be seen as a procedure which reduces the entropy, but in our case of very deep ($t \rightarrow \infty$) random formulas the entropy is at its maximum when $\epsilon > \epsilon(k)$. Thus any computation, even as simple as computing identity function, can not be performed reliably in this regime.

For odd k our result for the bound $\epsilon(k)$ is exactly equal to the *exact* threshold for reliable computation by general k -ary formulas [12], [13]. It is not clear however if this threshold is also exact, i.e. any Boolean function can be computed for $\epsilon \in (0, \epsilon(k))$ with the error $\delta < 1/2$, for randomly generated formulas. For even $k > 2$ this threshold is not known, but our result suggests that for balanced gates α it can not exceed the bound $\epsilon(k)$ of Theorem 2.1. Furthermore as $k \rightarrow \infty$ the $\epsilon(k)$ approaches $1/2$ as $1/2 - \epsilon(k) = O(1/\sqrt{k})$, this is follows from the Stirling's approximation of $b(k)$, which is in agreement with the bound computed in [16] for general formulas.

The results of Lemma 3.1, which are used in the proof of Theorem 2.1, can be also exploited to show that reliable computation in randomly generated formulas is possible. This can be shown as follows. Suppose that gate α in the process (1) is the same as the one stud-

ied in Lemma 3.1. Assume that the initial distribution $P_0[f]$ is such that the stationary distribution $P_\infty[f]$ of the noiseless process (2) is concentrated on only one Boolean function, i.e. all formulas compute the same function. This implies that for any input $I \in \{-1, 1\}^N$ all formulas simultaneously provide an output of $+1$ or -1 . Then in the presence of noise the average formula errors in its output with the probability $(1 - m_I(\infty))/2$, where $m_I(\infty)$ is the stationary solution of equation (5) corresponding to this input. From the analysis in Lemma 3.1 follows that the maximum of this error over all inputs is $\delta = (1 - m(\infty))/2$ and is bounded away from $1/2$ when $\epsilon < \epsilon(k)$. Furthermore, the output error can be reduced by decreasing ϵ or by increasing k . Thus in this regime any Boolean function can be computed with any desired accuracy.

III. COMPUTATION OF THE LOWER BOUND VALUES

In this section we compute the values of lower bounds appearing in Theorem 2.1. In order to do this we choose a balanced gate $\chi(\sigma)$ from the set of gates $\text{sgn} \left[\sum_{j=1}^k \sigma_j \right] + \mathbf{1} \left[\sum_{j=1}^k \sigma_j = 0 \right] \gamma(\sigma)$, where $\gamma(\sigma) \in \{-1, 1\}$ is such that $\sum_{\sigma} \mathbf{1} \left[\sum_{j=1}^k \sigma_j = 0 \right] \gamma(\sigma) = 0$, and consider the first moments $m_i(t) = \sum_f P_t[f] f^i$. These are governed by the equations

$$\begin{aligned}
 m(t+1) &= F_\chi(m(t)) \\
 &= \tanh(\beta) \sum_{\ell=0}^k \binom{k}{\ell} \text{sgn}[2\ell - k] \\
 &\quad \times \left[\frac{1+m(t)}{2} \right]^\ell \left[\frac{1-m(t)}{2} \right]^{k-\ell}
 \end{aligned} \tag{5}$$

Lemma 3.1: For $k \geq 3$ the function $F_\chi(m)$, defined in the equation (5), has the following properties: i) if $\tanh \beta \leq b(k)$ then $m > F_\chi(m)$ for $m \in (0, 1]$ and $F_\chi(m) > m$ for $m \in [-1, 0)$; ii) if $\tanh \beta > b(k)$ then $\exists m^* \neq 0$ such that $m^* = F_\chi(m^*)$, where $b(k)$ is defined in the Theorem 2.1.

Proof: This lemma follows from the equalities $F_\chi(\pm 1) = \pm \tanh \beta$, $F_\chi(0) = 0$ (this can be shown by direct substitution) and the fact that $F_\chi(m)$ is a strictly increasing function, which is also convex and concave on the intervals $(-1, 0)$ and $(0, 1)$, respectively (to show this we study properties of $F_\chi(m)$ in the Appendix B). Then i) is true because $\frac{dF_\chi}{dm}|_{m=0} < 1$ when $\tanh \beta < b(k)$ and ii) is true because of $\frac{dF_\chi}{dm}|_{m=0} \geq 1$ when $\tanh \beta \geq b(k)$. ■

IV. MOMENTS OF $P_t[f]$

Let us consider equation (4) for an n -th moment m . Assuming that all lower order moments vanish allows us to write this equation in a very simple form

$$\begin{aligned}
 m(t+1) &= F_\alpha(m(t)) \\
 &= \tanh^n(\beta) \sum_{\{\sigma_j^i\}} \prod_{j=1}^k \left\{ \right. \\
 &\quad \times \frac{1}{2^n} \left[1 + m(t) \prod_{i=1}^n \sigma_j^i \right] \left. \right\} \\
 &\quad \times \prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i).
 \end{aligned} \tag{6}$$

For a balanced gate α the point $m = 0$ is a stationary solution of the above equation and has the following property.

Lemma 4.1: The point $m = 0$ is a stable and unique solution of (6) when $\tanh^n(\beta) < b(k)$.

Proof: In order to prove this we first show that

$$\begin{aligned}
 &\tanh(\beta)^{n-1} F_\chi(m) \\
 &= \tanh^n(\beta) \sum_{\{\sigma_j^i\}} \prod_{j=1}^k \frac{1}{2^n} \left[1 + m \prod_{i=1}^n \sigma_j^i \right] \\
 &\quad \times \text{sgn} \left[\sum_{j=1}^k \prod_{i=1}^n \sigma_j^i \right],
 \end{aligned} \tag{7}$$

where $F_\chi(m)$ is defined in (5). This can be shown by a direct calculation as follows

$$\begin{aligned}
 & \tanh^n(\beta) \sum_{\{\sigma_j^i\}} \prod_{j=1}^k \frac{1}{2^n} \left[1 + m \prod_{i=1}^n \sigma_j^i \right] \\
 & \times \operatorname{sgn} \left[\sum_{j=1}^k \prod_{i=1}^n \sigma_j^i \right] \\
 & = \tanh^n(\beta) \frac{1}{2^{n-1}} \sum_{\{\sigma_j^i\}} \prod_{j=1}^k \left[\frac{1 + m\sigma_j^1}{2} \right] \\
 & \times \operatorname{sgn} \left[\sum_{j=1}^k \sigma_j^1 \right] = \tanh(\beta)^{n-1} F_\chi(m).
 \end{aligned} \tag{8}$$

In the above the first equality was obtained by applying transformation $\sigma_j^1 \rightarrow \prod_{i=1}^n \sigma_j^i$ and the last equality followed from comparing this result with the right hand side of equation (5).

Next, for a balanced gate α we compute the difference $\Delta(m) = \tanh^{n-1}(\beta)F_\chi(m) - F_\alpha(m)$ in Appendix C. The result of this computation is that $\Delta(m) \geq 0$ and $\Delta(m) \leq 0$ on the intervals $m \in [0, 1)$ and $m \in (-1, 0]$, respectively, from which the bounds $\tanh^{n-1}(\beta)F_\chi(m) \geq F_\alpha(m)$ and $\tanh^{n-1}(\beta)F_\chi(m) \leq F_\alpha(m)$ on the same intervals follow. The behaviour of $\tanh^{n-1}(\beta)F_\chi(m)$ with respect to the inverse temperature β is the same as of $F_\chi(m)$, which we described in Lemma 3.1, but with the $\tanh(\beta)$ being replaced by the $\tanh^n(\beta)$. ■

V. CONCLUSION

The paper extends previous work [12], [13] on the reliability of computation in Boolean formulas and generation of random Boolean functions [8], [3], by investigating the properties of formulas constructed by a random growth process whereby computing elements, primarily k -ary balanced gates, are subject to ϵ -noise.

We show that the noisy growth process is ergodic above the noise bound $\epsilon(k)$ and hence the formulas

generated by it are unreliable. We also show that formulas constructed from majority-like gates, which saturate this bound, can be used for computing any Boolean function when $\epsilon < \epsilon(k)$. Our earlier work, which uses methods of non-equilibrium statistical physics, suggests that the same noise bound also applies to the noisy feed-forward [17] and recurrent Boolean networks [18].

The current analysis is restricted to reliable computation in a growth process that uses only *balanced gates*⁴ and produces (without noise) only *one* Boolean function; but we envisage that it can be extended to study more general scenarios of non-balanced gates and a richer distributions of Boolean functions [3].

APPENDIX A

MOMENT REPRESENTATION OF $P_t[f]$

The probability distribution $P_t[f]$ can be represented via its moments. In order to find this representation we can use the identity $\sum_{\hat{f}} \delta[\hat{f}; f] = 1$ to write $P_t[f] = \sum_{\hat{f}} \delta[\hat{f}; f] P_t[\hat{f}]$. Then, because $\delta[x; y] = \frac{1}{2}(1 + xy)$ for $x, y \in \{-1, 1\}$, we obtain

$$\begin{aligned}
 P_t[f] & = \sum_{\hat{f}} P_t[\hat{f}] \prod_{i=1}^{2^N} \frac{1}{2} (1 + \hat{f}^i f^i) \\
 & = \frac{1}{2^{2^N}} \left(1 + \sum_{\hat{f}} P_t[\hat{f}] \sum_{S \subseteq [2^N]} \prod_{i \in S} \hat{f}^i f^i \right) \\
 & = \frac{1}{2^{2^N}} \left(1 + \sum_{S \subseteq [2^N]} m^S(t) \prod_{i \in S} f^i \right),
 \end{aligned} \tag{9}$$

where $m^S(t) = \sum_{\hat{f}} P_t[\hat{f}] \prod_{i \in S} \hat{f}^i$ are the moments of $P_t[f]$.

Let us now derive equation governing evolution of the n -th moment $m^{i_1, i_2, \dots, i_n}(t)$. This can be obtained by multiplying both sides of equation (1) by the monomial

⁴The results of this paper can be easily extended to the *distributions* over balanced gates [18].

$\prod_{i \in \mathbf{I}} f^i$, where $\mathbf{I} = \{i_1, \dots, i_n\}$, and taking the sums over f as follows

$$\begin{aligned}
 & \sum_f P_{t+1}[f] \prod_{i \in \mathbf{I}} f^i \\
 &= \sum_f \sum_{\{f_j\}} \prod_{j=1}^k \{P_t[f_j]\} \\
 & \times \prod_{i=1}^{2^N} \frac{e^{\beta f^i \alpha(f_1^i, \dots, f_k^i)}}{2 \cosh \beta \alpha(f_1^i, \dots, f_k^i)} \prod_{\ell \in \mathbf{I}} f^\ell \\
 &= \tanh^n(\beta) \sum_{\{f_j^i\}} \prod_{j=1}^k \{P_t[f_j^{i_1}, \dots, f_j^{i_n}]\} \\
 & \times \prod_{i \in \mathbf{I}} \alpha(f_1^i, \dots, f_k^i) \\
 &= m^{i_1, \dots, i_n}(t+1),
 \end{aligned} \tag{10}$$

where in the above we have used the property $\tanh(-x) = -\tanh(x)$. Finally, using the moment representation (9) in the above, we obtain

$$\begin{aligned}
 & m^{i_1, \dots, i_n}(t+1) \\
 &= \tanh^n(\beta) \sum_{\{f_j^i\}} \prod_{j=1}^k \left\{ \right. \\
 & \times \frac{1}{2^n} \left[1 + \sum_{S \subseteq \mathbf{I}} m^S(t) \prod_{i \in S} f_j^i \right] \left. \right\} \\
 & \times \prod_{i \in \mathbf{I}} \alpha(f_1^i, \dots, f_k^i).
 \end{aligned} \tag{11} \quad \text{but}$$

APPENDIX B

DERIVATION OF F_χ AND ANALYSIS OF ITS PROPERTIES

Here we first derive the function $F_\chi(m)$ then we study its properties. Let us first compute the sum

$$\begin{aligned}
 & \sum_{\{f_j\}} \prod_{j=1}^k \left[\frac{1+f_j m}{2} \right] \left\{ \operatorname{sgn} \left[\sum_{j=1}^k f_j \right] \right. \\
 & \left. + \mathbf{1} \left[\sum_{j=1}^k f_j = 0 \right] \gamma(f_1, \dots, f_k) \right\} \\
 &= \sum_{\{f_j\}} \left[\frac{1+m}{2} \right]^{(\sum_{j=1}^k f_j + k)/2} \left[\frac{1-m}{2} \right]^{(k - \sum_{j=1}^k f_j)/2} \\
 & \times \left\{ \operatorname{sgn} \left[\sum_{j=1}^k f_j \right] + \mathbf{1} \left[\sum_{j=1}^k f_j = 0 \right] \gamma(f_1, \dots, f_k) \right\} \\
 &= \sum_{\ell=0}^k \binom{k}{\ell} \left[\frac{1+m}{2} \right]^\ell \left[\frac{1-m}{2} \right]^{k-\ell} \operatorname{sgn}[2\ell - k],
 \end{aligned} \tag{12}$$

in the definition (10) for the specific choice of $\alpha \equiv \chi$. This result leads to the function $F_\chi(m)$ used in equation (5).

We are interested in how the function $F_\chi(m)$ behaves on the interval $m \in [-1, 1]$ and how this behaviour is affected by the parameter $\tanh \beta$. In order to find this out we first rewrite $F_\chi(m)$ as follows

$$\begin{aligned}
 & F_\chi(m) \\
 &= \tanh(\beta) \\
 & \times \sum_{\ell=0}^k \binom{k}{\ell} \left[\frac{1+m}{2} \right]^\ell \left[\frac{1-m}{2} \right]^{k-\ell} \\
 & \times \{ \mathbf{1}[2\ell - k > 0] - \mathbf{1}[2\ell - k < 0] \},
 \end{aligned} \tag{13}$$

$$\begin{aligned}
 & \sum_{\ell=0}^k \binom{k}{\ell} \left[\frac{1+m}{2} \right]^\ell \left[\frac{1-m}{2} \right]^{k-\ell} \\
 &= \sum_{\ell=0}^k \binom{k}{\ell} \left[\frac{1+m}{2} \right]^\ell \left[\frac{1-m}{2} \right]^{k-\ell} \\
 & \times \{ \mathbf{1}[2\ell - k \geq 0] + \mathbf{1}[2\ell - k < 0] \} \\
 &= 1
 \end{aligned} \tag{14}$$

so

$$\begin{aligned}
 F_\chi(m) & \\
 &= \tanh(\beta) \left(1 - 2 \sum_{\ell=0}^{\tilde{k}} \binom{k}{\ell} \right. \\
 &\quad \times \left[\frac{1+m}{2} \right]^\ell \left[\frac{1-m}{2} \right]^{k-\ell} \\
 &\quad \left. - \mathbf{1}[k \equiv 0 \pmod{2}] \right. \\
 &\quad \times \left. \binom{k}{k/2} \left(\left[\frac{1+m}{2} \right] \left[\frac{1-m}{2} \right] \right)^{k/2} \right), \tag{15}
 \end{aligned}$$

where

$$\begin{aligned}
 \tilde{k} &= \mathbf{1}[k \equiv 0 \pmod{2}](k/2 - 1) \\
 &+ \mathbf{1}[k \equiv 0 \pmod{1}](k - 1)/2.
 \end{aligned}$$

Now we use the above representation of $F_\chi(m)$ to compute

$$\begin{aligned}
 \frac{d}{dm} F_\chi(m) & \\
 &= \tanh(\beta) \sum_{\ell=0}^{\tilde{k}} \binom{k}{\ell} \\
 &\quad \times \left((k-\ell) \left[\frac{1+m}{2} \right]^\ell \left[\frac{1-m}{2} \right]^{k-\ell-1} \right. \\
 &\quad \left. - \ell \left[\frac{1+m}{2} \right]^{\ell-1} \left[\frac{1-m}{2} \right]^{k-\ell} \right) \\
 &\quad + \tanh(\beta) \frac{k}{4} \mathbf{1}[k \equiv 0 \pmod{2}] \binom{k}{k/2} \\
 &\quad \times m \left(\left[\frac{1+m}{2} \right] \left[\frac{1-m}{2} \right] \right)^{k/2-1} \\
 &= \tanh(\beta) \binom{k}{\tilde{k}+1} (\tilde{k}+1) \\
 &\quad \times \left[\frac{1+m}{2} \right]^{\tilde{k}} \left[\frac{1-m}{2} \right]^{k-\tilde{k}-1} \\
 &\quad + \tanh(\beta) \frac{k}{4} \mathbf{1}[k \equiv 0 \pmod{2}] \binom{k}{k/2} \\
 &\quad \times m \left(\left[\frac{1+m}{2} \right] \left[\frac{1-m}{2} \right] \right)^{k/2-1} \tag{16}
 \end{aligned}$$

So, using definition of \tilde{k} , we obtain

$$\begin{aligned}
 \frac{d}{dm} F_\chi(m) &= \tanh(\beta) \binom{k}{(k+1)/2} \left(\frac{k+1}{2} \right) \\
 &\quad \times \left(\left[\frac{1+m}{2} \right] \left[\frac{1-m}{2} \right] \right)^{(k-1)/2} \tag{17}
 \end{aligned}$$

for k odd and

$$\begin{aligned}
 \frac{d}{dm} F_\chi(m) &= \tanh(\beta) \binom{k}{k/2} \left(\frac{k}{4} \right) \\
 &\quad \times \left(\left[\frac{1+m}{2} \right] \left[\frac{1-m}{2} \right] \right)^{k/2-1} \tag{18}
 \end{aligned}$$

for k even.

Thus $\frac{d}{dm} F_\chi(m) > 0$ on for all $m \in (-1, 1)$ and hence $F_\chi(m)$ is a strictly increasing function. Furthermore, the function $F_\chi(m)$ at the point $m = 0$ changes its slope from $\frac{d}{dm} F_\chi(m)|_{m=0} < 1$ to $\frac{d}{dm} F_\chi(m)|_{m=0} \geq 1$ at

$$\tanh(\beta) = 2^{k-1}/k \binom{k-1}{(k-1)/2}$$

for k odd and

$$\tanh(\beta) = 2^{k-2} / \binom{k-2}{(k-2)/2} (k-1)$$

for k even.

Let us now compute the second derivative of $F_\chi(m)$. Differentiating equations (17) and (18) with respect to m gives us

$$\begin{aligned}
 \frac{d^2}{dm^2} F_\chi(m) &= -m \tanh(\beta) \\
 &\quad \times \binom{k}{(k+1)/2} \left(\frac{k+1}{2} \right) \frac{(k-1)}{4} \\
 &\quad \times \left(\left[\frac{1+m}{2} \right] \left[\frac{1-m}{2} \right] \right)^{(k-1)/2-1} \tag{19}
 \end{aligned}$$

for k odd and

$$\begin{aligned}
 \frac{d^2}{dm^2} F_\chi(m) &= -m \tanh(\beta) \\
 &\quad \times \binom{k}{k/2} \left(\frac{k}{4} \right) \frac{(k-2)}{4} \\
 &\quad \times \left(\left[\frac{1+m}{2} \right] \left[\frac{1-m}{2} \right] \right)^{k/2-2} \tag{20}
 \end{aligned}$$

for k even. We note that both are of the form $\frac{d^2}{dm^2} F_\chi(m) = -mG(m)$, where $G(m) > 0$ for all $m \in (-1, 1)$. Thus the function $F_\chi(m)$ is strictly

convex and concave on the intervals $(-1, 0)$ and $(0, 1)$ to obtain respectively.

APPENDIX C

COMPUTATION OF $\Delta(m)$ AND ANALYSIS OF ITS PROPERTIES

Let us first define the average $\langle \dots \rangle_{\sigma|m} = \sum_{\{\sigma_j^i\}} \prod_{j=1}^k \frac{1}{2^n} [1 + m \prod_{i=1}^n \sigma_j^i] (\dots)$ and shorthand notations $\{\mathbf{1}_+[\sigma], \mathbf{1}_-[\sigma], \mathbf{1}_0[\sigma]\}$ for the indicator functions $\{\mathbf{1}[\sum_{j=1}^k \prod_{i=1}^n \sigma_j^i > 0], \mathbf{1}[\sum_{j=1}^k \prod_{i=1}^n \sigma_j^i < 0], \mathbf{1}[\sum_{j=1}^k \prod_{i=1}^n \sigma_j^i = 0]\}$. Then the rescaled difference $\Delta(m)/4 \tanh^n(\beta)$ can be computed as follows

$$\begin{aligned} & \frac{\Delta(m)}{4 \tanh^n(\beta)} \quad (21) \\ &= \frac{1}{4} \left\langle \operatorname{sgn} \left[\sum_{j=1}^k \prod_{i=1}^n \sigma_j^i \right] - \prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) \right\rangle_{\sigma|m} \\ &= \frac{1}{4} \left\langle \mathbf{1}_+[\sigma] - \mathbf{1}_-[\sigma] \right. \\ &\quad \left. - (\mathbf{1}_+[\sigma] + \mathbf{1}_-[\sigma] + \mathbf{1}_0[\sigma]) \prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) \right\rangle_{\sigma|m} \\ &= \frac{1}{2} \left\langle \mathbf{1}_+[\sigma] \mathbf{1} \left[\prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) = -1 \right] \right. \\ &\quad \left. - \mathbf{1}_-[\sigma] \mathbf{1} \left[\prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) = +1 \right] \right. \\ &\quad \left. - \frac{1}{2} \mathbf{1}_0[\sigma] \prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) \right\rangle_{\sigma|m}. \end{aligned}$$

In the above we can use the identity

$$\begin{aligned} & \prod_{j=1}^k \frac{1}{2^n} \left[1 + m \prod_{i=1}^n \sigma_j^i \right] = \quad (22) \\ & \left[\frac{1+m}{2^n} \right]^{\frac{k + \sum_{j=1}^k \prod_{i=1}^n \sigma_j^i}{2}} \left[\frac{1-m}{2^n} \right]^{\frac{k - \sum_{j=1}^k \prod_{i=1}^n \sigma_j^i}{2}} \end{aligned}$$

$$\begin{aligned} & \frac{\Delta(m)}{4 \tanh^n(\beta)} \quad (23) \\ &= \frac{1}{2} \left(\left[\frac{1+m}{2^n} \right] \left[\frac{1-m}{2^n} \right] \right)^{\frac{k}{2}} \\ &\times \left\{ \sum_{\{\sigma_j^i\}} \left[\frac{1+m}{1-m} \right]^{\frac{|\sum_{j=1}^k \prod_{i=1}^n \sigma_j^i|}{2}} \right. \\ &\times \mathbf{1}_+[\sigma] \mathbf{1} \left[\prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) = -1 \right] \\ &\quad \left. - \sum_{\{\sigma_j^i\}} \left[\frac{1-m}{1+m} \right]^{\frac{|\sum_{j=1}^k \prod_{i=1}^n \sigma_j^i|}{2}} \right. \\ &\times \mathbf{1}_-[\sigma] \mathbf{1} \left[\prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) = +1 \right] \\ &\quad \left. - \frac{1}{2} \sum_{\{\sigma_j^i\}} \mathbf{1}_0[\sigma] \prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) \right\}. \end{aligned}$$

Now because α is a balanced gate we have the following identity

$$\begin{aligned} & \sum_{\{\sigma_j^i\}} \prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) \quad (24) \\ &= \sum_{\{\sigma_j^i\}} \left(\mathbf{1}_+[\sigma] + \mathbf{1}_-[\sigma] + \mathbf{1}_0[\sigma] \right) \\ &\quad \times \left(\mathbf{1} \left[\prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) = +1 \right] \right. \\ &\quad \left. - \mathbf{1} \left[\prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) = -1 \right] \right) \\ &= \sum_{\{\sigma_j^i\}} \left(\frac{1}{2} \mathbf{1}_0[\sigma] \prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) \right. \\ &\quad \left. + \mathbf{1}_-[\sigma] \mathbf{1} \left[\prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) = +1 \right] \right. \\ &\quad \left. - \mathbf{1}_+[\sigma] \mathbf{1} \left[\prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) = -1 \right] \right) \\ &= 0. \end{aligned}$$

Adding the above representation of zero to the terms inside the curly brackets in equation (23) leads to the

final result

$$\begin{aligned}
 \Delta(m) &= 2 \tanh^n(\beta) \left(\left[\frac{1+m}{2^n} \right] \left[\frac{1-m}{2^n} \right] \right)^{\frac{k}{2}} \\
 &\times \left\{ \sum_{\{\sigma_j^i\}} \left(\left[\frac{1+m}{1-m} \right]^{\frac{|\sum_{j=1}^k \prod_{i=1}^n \sigma_j^i|}{2}} - 1 \right) \right. \\
 &\quad \times \mathbf{1}_+[\sigma] \mathbf{1} \left[\prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) = -1 \right] \\
 &\quad + \sum_{\{\sigma_j^i\}} \left(1 - \left[\frac{1-m}{1+m} \right]^{\frac{|\sum_{j=1}^k \prod_{i=1}^n \sigma_j^i|}{2}} \right) \\
 &\quad \left. \times \mathbf{1}_-[\sigma] \mathbf{1} \left[\prod_{i=1}^n \alpha(\sigma_1^i, \dots, \sigma_k^i) = +1 \right] \right\}.
 \end{aligned} \tag{25}$$

From the above it is clear that $\Delta(m) \geq 0$ for $m \in [0, 1)$ and $\Delta(m) \leq 0$ for $m \in (-1, 0]$.

ACKNOWLEDGMENT

This work is supported by the EU FET project STAMINA (FP7-265496) and the Leverhulme trust grant F/00 250/H.

REFERENCES

[1] I. Wegener, *The Complexity of Boolean Functions*. Stuttgart: B. G. Teubner, 1987.

[2] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán, "Threshold circuits of bounded depth," *J. Comput. Syst. Sci.*, vol. 46, no. 2, pp. 129 – 154, 1993.

[3] A. Brodsky and N. Pippenger, "The boolean functions computed by random boolean formulas or how to grow the right function," *Random Struct. Algor.*, vol. 27, no. 4, pp. 490–519, 2005.

[4] H. Lefmann and P. Savický, "Some typical properties of large and/or boolean formulas," *Random Struct. Algor.*, vol. 10, no. 3, pp. 337–351, 1997.

[5] B. Chauvin, P. Flajolet, D. Gardy, and B. Gittenberger, "And/or trees revisited," *Comb. Probab. Comput.*, vol. 13, no. 4-5, pp. 475–497, 2004.

[6] D. Gardy and A. Woods, *DMTCS Proceedings*, vol. AD, pp. 139–146, 2005.

[7] L. Valiant, "Short monotone formulae for the majority function," *J. Algorithm.*, vol. 5, no. 3, pp. 363 – 366, 1984.

[8] P. Savický, "Random boolean formulas representing any boolean function with asymptotically equal probability," *Discrete Math.*, vol. 83, no. 1, pp. 95–103, 1990.

[9] J. Von Neumann, *Probabilistic logics and the synthesis of reliable organisms from unreliable components*, ser. Automata Studies. Princeton, NJ: Princeton University Press, 1956, p. 43i; $\frac{1}{2}$ 98.

[10] N. Pippenger, "Reliable computation by formulas in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 34, no. 2, pp. 194–197, 1988.

[11] T. Feder, "Reliable computation by networks in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 35, no. 3, pp. 569–571, 1989.

[12] B. Hajek and T. Weller, "On the maximum tolerable noise for reliable computation by formulas," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 388–391, 1991.

[13] W. Evans and L. Schulman, "On the maximum tolerable noise of k-input gates for reliable computation by formulas," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 3094–3098, 2003.

[14] W. Evans and N. Pippenger, "On the maximum tolerable noise for reliable computation by formulas," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 1299–1305, 1998.

[15] F. Unger, "Noise threshold for universality of two-input gates," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3693–3698, 2008.

[16] W. Evans and L. Schulman, "Signal propagation and noisy circuits," *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 2367–2373, 1999.

[17] A. Mozeika, D. Saad, and J. Raymond, "Noisy random boolean formulae: A statistical physics perspective," *Phys. Rev. E.*, vol. 82, p. 041112, 2010.

[18] A. Mozeika and D. Saad, "Phase transitions and memory effects in the dynamics of Boolean networks," *Philos. Mag.*, vol. 92, pp. 210–229, 2012.