An Efficient Non-Gaussian Sampling Method for High Sigma SRAM Yield Analysis

JINYUAN ZHAI and CHANGHAO YAN, Fudan University
SHENG-GUO WANG, University of North Carolina at Charlotte
DIAN ZHOU, Fudan University and University of Texas at Dallas
HAI ZHOU, Fudan University and Northwestern University
XUAN ZENG, Fudan University

Yield[1] analysis of SRAM is a challenging issue, because the failure rates of SRAM cells are extremely small. In this article, an efficient non-Gaussian sampling method of cross entropy optimization is proposed for estimating the high sigma SRAM yield. Instead of sampling with the Gaussian distribution in existing methods, a non-Gaussian distribution, i.e., a joint one-dimensional generalized Pareto distribution and (n-1)-dimensional Gaussian distribution, is taken as the function family of practical distribution, which is proved to be more suitable to fit the ideal distribution in the view of extreme failure event. To minimize the cross entropy between practical and ideal distributions, a sequential quadratic programing solver with multiple starting points strategy is applied for calculating the optimal parameters of practical distributions. Experimental results show that the proposed non-Gaussian sampling is a 2.2–4.1× speedup over the Gaussian sampling, on the whole, it is about a 1.6–2.3× speedup over state-of-the-art methods with low- and high-dimensional cases without loss of accuracy

CCS Concepts: • **Hardware → Design for manufacturability**; **Yield and cost modeling**;

Additional Key Words and Phrases: Failure rate, SRAM, cross entropy minimization, generalized pareto distribution

ACM Reference format:
Jinyuan Zhai, Changhao Yan, Sheng-Guo Wang, Dian Zhou, Hai Zhou, and Xuan Zeng. 2018. An Efficient Non-Gaussian Sampling Method for High Sigma SRAM Yield Analysis. *ACM Trans. Des. Autom. Electron. Syst.* 23, 3, Article 36 (March 2018), 23 pages.
https://doi.org/10.1145/3174866

This research was supported partly by National Key Research and Development Program of China 2016YFB0201304; partly by the National Major Science and Technology Special Project of China (2017ZX01028-101-003); partly by National Natural Science Foundation of China (NSFC) research projects 61674042, 61574046, 61574044, 61774045, and 61628402; partly by the Recruitment Program of Global Experts (the Thousand Talents Plan); partly by National Science Foundation (NSF) under CNS-1441695, CCF-1533656, and CNS-1651695; and partly by NSF Grant 1115564.
Authors' addresses: J. Zhai and C. Yan (corresponding authors), State Key Laboratory of ASIC and System, School of Microelectronics, Fudan University, Shanghai 201203, China; emails: {jydi15, yanch}@fudan.edu.cn; S.-G. Wang, Department of Engineering Technology, University of North Carolina at Charlotte, Charlotte, North Carolina 28223-0001; email: swang@uncc.edu; D. Zhou, State Key Laboratory of ASIC and System, School of Microelectronics, Fudan University, Shanghai 201203, China, and also Department of Electrical Engineering, University of Texas at Dallas, Richardson, Texas 75080; email: zhoud@utdallas.edu; H. Zhou, State Key Laboratory of ASIC and System, School of Microelectronics, Fudan University, Shanghai 201203, China, and also Department of Electrical Engineering and Computer Science, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208-3118; email: haizhou@northwestern.edu; X. Zeng (corresponding authors), the State Key Laboratory of ASIC and System, School of Microelectronics, Fudan University, Shanghai 201203, China; email: xzeng@fudan.edu.cn.

https://doi.org/10.1145/3174866

1 INTRODUCTION

As semiconductor fabrication technology enters sub-14nm, process variances, such as doping fluctuation and threshold voltage variation, have significant impacts on performance and reliability of SRAM. SRAM cell is generally designed in the minimum size of the technology node for cost consideration, which makes it more vulnerable to process variations. Meanwhile, SRAM cells are replicated into a large array. For a memory chip with moderate yield, the failure rate of a SRAM bit cell is required to be extremely low (typically $10^{-6} \sim 10^{-8}$) [1]. Therefore, the Monte Carlo (MC) method, though straightforward and easy to implement, is extremely inefficient for SRAM yield analysis in that most samples fall into the feasible region. To address the efficiency issue of MC, efficient and accurate methods are imperative to estimate SRAM failure rates.

Many statistical methods have already been proposed. Generally, these methods can be roughly divided into three categories: Importance Sampling (IS) methods [1–9], failure boundary searching methods [12–16], and extreme value theory [18, 19].

The basic idea of IS is sampling near the failure region based on an assumed distribution termed *practical distribution*, instead of the original probability density functions (PDF) of process parameters. Minimized Norm Importance Sampling (MNIS) [3] and Spherical Sampling (SS) [4] try to shift the original distribution of process variances with an *optimal shift vector* (OSV), which is defined by the failure point with the minimal norm distance. This shifted distribution makes the success-to-fail an almost half-to-half event, which improves the sampling efficiency tremendously. However, because the ideal distribution is unknown, they simply adopt the Gaussian distribution as the practical distribution, and therefore large amounts of SPICE (Simulation Program with Integrated Circuit Emphasis, circuit simulatior used in integrated circuit to perdict circuit behacior) simulations are still inevitable. Solido's commercial tool Optimal Importance Sampling (OptIS) is an application of IS technology, which shows tremendous speedup (6–200×) over Monte Carlo methods for rare-event simulations [11].

Failure boundary searching methods try to find the boundary between failure and success regions in the parameter space, and the failure rate is obtained with numerical quadrature in failure regions without time-consuming SPICE simulations. The main drawback of boundary searching methods is that they become more and more difficult to obtain the hyper-surface of failure boundary with increasing dimensionality of space. In fact, boundary-based methods struggle to handle the problems where the dimensionalities are more than 100 [17].

Extreme Value Theory (EVT) and data filtering are combined in References [18] and [19] to efficiently capture rare failure events of SRAM. The key ideas in References [18] and [19] are using EVT to fit the tail distribution of performances (e.g., writing time) and applying a support vector machine (SVM) *classifier* to filter out most unlikely failed samples. However, these analytic models are not accurate enough [3] and also suffer from the curse of dimensionality [4].

Recently, Importance Boundary Sampling (IBS) [9] combines the advantage of importance sampling and boundary search method by introducing a surrogate model for part of the boundaries, which improves the sampling efficiency greatly. However, it seems only applicable for low-dimensional cases in that the surrogate model is hard to build on high-dimensional cases [5]. Reference [5] extends IS methods to address high-dimensional problems. The key idea is to apply sequential quadratic programing (SQP) with a multiple starting points (MSP) strategy to find the OSV efficiently. However, when sampling near OSV, it still applies the same Gaussian distribution as MNIS and SS do, which severely influences its efficiency.

For importance sampling methods, if the practical distribution is equal to the ideal distribution, then the convergence speed of IS becomes the highest [7]. Unfortunately, the ideal distribution is unknown in advance. Therefore, the main challenge of IS is how to determinate the proper or even

optimal practical PDF, which will dramatically improve the efficiency of IS. An inevitable problem is how to measure the *difference* between the practical and ideal distribution.

Reference [6] firstly introduced the concept of *cross entropy* for SRAM failure rate analysis to quantitatively evaluate the difference between practical shifted normal distribution $N(\boldsymbol{\mu},\mathbf{1})$ and ideal distribution. The optimal parameter, i.e., mean $\boldsymbol{\mu}^*$ of this normal distribution, can be simply calculated within a closed form. However, for constructing the ideal distribution, it randomly generates many samples to find failure points and then resamples near these failure points. This method is inaccurate and inefficient.

Reference [7] further extends the practical distribution from $N(\boldsymbol{\mu},\mathbf{1})$ [6] to $N(\boldsymbol{\mu},\boldsymbol{\sigma})$ with an extra standard deviation $\boldsymbol{\sigma}$, which allows the practical distribution to be similar to the ideal distribution. Besides, the ideal distribution is constructed in an iterative manner, where the resampling data are based on the latest $N(\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*)$. This is a reasonable strategy, although expensive. However, without any information about the failure boundary, it must set the initial value of $\boldsymbol{\mu}^*$ as zero, which leads to too many iterations and inefficiency.

To the best of our knowledge, References [6] and [7] are the only two references applying cross entropy for SRAM failure analysis. One of their efficiency losses is based on the assumption of sampling with Gaussian distribution. Such an assumption is essentially heuristic, although the optimal parameters $\boldsymbol{\mu}^*$ and $\boldsymbol{\sigma}^*$ of Gaussian distribution have beautifully analytic solutions. However, both theoretical analysis and experimental data show that the ideal distribution is significantly different with Gaussian distribution.

To address the above issues, we propose an efficient non-Gaussian sampling method of cross entropy optimization with a joint Generalized Pareto distribution (GPD) for high-sigma SRAM yield analysis. The main contributions of this article can be highlighted as follows. (1) Since SRAM failure can be view as a tail distribution of process parameters, a joint distribution with one-dimensional (1D) Generalized Pareto distribution and (n-1)-dimensional Gaussian distribution is proposed. Such a joint GPD can be proved in theory to be suitable for the function family of practical distribution. (2) The proposed joint GPD distribution, instead of Gaussian distribution, makes the closed forms of optimal parameters unavailable. A numerical framework is proposed for solving the cross entropy minimization with MSP-SQP algorithm, without any extra cost of SPICE simulations. (3) To more efficiently find the OSVs of failure boundaries, we combine the method proposed in Reference [5] and sparse recovery technology in Reference [26], which further achieve 1.6× speedup over Reference [5] in finding OSV. (4) Experimental results verify the effectiveness and efficiency of the proposed method. The proposed non-Gaussian sampling is about a 2.2–4.1× speedup over the Gaussian sampling, and in total it is about a 1.6–2.3× speedup over the-state-of-art method [5] without loss of accuracy.

The remainder of this article is organized as follows. Section 2 simply reviews the importance sampling and cross entropy methods. Section 3 and 4 describes the proposed method in detail. Section 5 verifies the significant speedup of proposed method by experiments both in low- and high-dimensional cases. Finally, Section 6 gives a short conclusion.

2 BACKGROUND

2.1 Problem Formulation

Let $\boldsymbol{x} = [x_1, \ldots, x_D]^T$ be a D-dimensional random variable that denotes all the process parameters, such as threshold voltage, oxide thickness, or carrier motilities. Because correlated random variables can be transformed to independent variables by principal component analysis [20], without loss of generality, the joint PDF $p(\boldsymbol{x})$ can be represented as

$$p(\boldsymbol{x}) = \prod_{d=1}^{D} p_d(x_d), \tag{1}$$

where $p_d(x_d)$ are independent PDFs of process parameters given by foundries.

SRAM failure rate can be calculated formally as

$$P_{fail} = \int_{\Omega} p(\boldsymbol{x}) d\boldsymbol{x} = \int_{-\infty}^{+\infty} I(\boldsymbol{x} \in \Omega)\, p(\boldsymbol{x})\, d\boldsymbol{x}, \tag{2}$$

where Ω denotes the failure region, which is a subset of variation space of process parameters, and $I(\boldsymbol{x} \in \Omega)$ is the indicator function of failure region Ω. $I(\boldsymbol{x} \in \Omega)$ is 1 if $\boldsymbol{x} \in \Omega$; otherwise, it is 0. Since the indicator function $I(\boldsymbol{x})$ and failure region Ω are unknown in advance, P_{fail} cannot be calculated directly by Equation (2).

2.2 Monte Carlo Method

The Monte Carlo method is the most common and traditional approach to estimate SRAM failure rate. In MC, after sampling N points following $p(\boldsymbol{x})$, the SRAM failure rate P_{fail} can be probability estimated as

$$P_{fail} \approx \hat{P}_{MC} = \frac{1}{N} \sum_{i=1}^{N} I(\boldsymbol{x}_i), \tag{3}$$

where $\boldsymbol{x_i}$ is the ith sample. The MC generates unbiased and asymptotically correct estimation of failure rate with enough samples.

The figure of merit (FOM), as a criterion for estimator, is given as [3]

$$\rho(\hat{P}) = \frac{\sqrt{Var(\hat{P})}}{\hat{P}}, \tag{4}$$

where $\hat{P}$ is the estimated failure rate and $Var(\hat{P})$is the variance of $\hat{P}$. FOM $\rho(\hat{P})$ reflects the accuracy and reliability of $\hat{P}$. Suppose $\rho(\hat{P}) \leq \varepsilon\sqrt{\log(1/\delta)}$; we can declare that the estimated of $\hat{P}$ is $(1-\varepsilon)100\%$ accurate with confidence $(1-\delta)100\%$ at least.

2.3 Importance Sampling

The most common approaches to improve the sampling efficiency are based on Importance Sampling. The main idea of IS is to replace the original distribution $p(\boldsymbol{x})$ by an assumed distribution $q(\boldsymbol{x})$, termed practical distribution. The failure rate then can be expressed as follows:

$$P_{IS} = \int_{-\infty}^{+\infty} \frac{I(\boldsymbol{x})\, p(\boldsymbol{x})}{q(\boldsymbol{x})} q(\boldsymbol{x})\, d\boldsymbol{x}. \tag{5}$$

In IS, with N samples generated from $q(\boldsymbol{x})$, the failure rate P_{fail} can be estimated as

$$\hat{P}_{IS} = \frac{1}{N} \sum_{i=1}^{N} \frac{I(\boldsymbol{x}_i)\, p(\boldsymbol{x}_i)}{q(\boldsymbol{x}_i)}. \tag{6}$$

In theory, the ideal distribution $p^{ideal}(\boldsymbol{x})$ for IS is given as

$$p^{ideal}(\boldsymbol{x}) = \frac{I(\boldsymbol{x})\, p(\boldsymbol{x})}{P_{fail}}. \tag{7}$$

If the practical distribution $q(\boldsymbol{x})$ in Equation (6) is so smartly chosen that it is nearly equal to $p^{ideal}(\boldsymbol{x})$, then the IS method will be much more efficient than MC with the same accuracy.

Although the ideal distribution (Equation (7)) is unknown, the IS method still inspires us on how to generate samples efficiently. We can see that the ideal distribution $p^{ideal}(\boldsymbol{x})$ is nonzero if and only

if sample $\boldsymbol{x_i}$ falls into failure region, which implies that more samples in the failure region will be better.

The variance of importance sampling $Var(\hat{P})$ is given as [3]

$$Var\left(\hat{P}_{IS}\right) = \frac{1}{N^2} \cdot \left[\sum_{i=1}^{N} \left(\frac{p\left(\mathbf{x}_i\right)}{q\left(\mathbf{x}_i\right)}\right)^2 I\left(\mathbf{x}_i\right) - N \cdot \hat{P}_{IS}^2\right]. \tag{8}$$

2.4 Cross Entropy Minimization

In probability theory and information theory, the *cross entropy* or *Kullback–Leibler divergence* measures the difference between two distributions [21]. The cross entropy between the practical distribution $p^{practical}(\boldsymbol{x})$ and the ideal distribution $p^{ideal}(\boldsymbol{x})$ is defined as

$$D = E_{p^{ideal}}\left[\log\left(\frac{p^{ideal}(\boldsymbol{x})}{p^{practical}(\boldsymbol{x})}\right)\right] = \int \log\left(\frac{p^{ideal}(\boldsymbol{x})}{p^{practical}(\boldsymbol{x})}\right) p^{ideal}(\boldsymbol{x}) d\boldsymbol{x}, \tag{9}$$

where $E[\cdot]$ is the expectation operator and both $p^{ideal}(\boldsymbol{x})$ and $p^{practical}(\boldsymbol{x})$ are defined over the same space which the random variable $\boldsymbol{x}$ belongs to.

The greater difference between $p^{practical}(\boldsymbol{x})$ and $p^{ideal}(\boldsymbol{x})$, the larger the cross entropy in absolute value. If two distributions are the same, then the cross entropy is zero. Therefore, the optimal practical distribution can be found by minimizing cross entropy.

If the function family of practical distributions is assumed to be Gaussian distribution, such as $N(\boldsymbol{\mu},\mathbf{1})$ in Reference [6] or $N(\boldsymbol{\mu},\boldsymbol{\sigma})$ in Reference [7], then analytical expressions of the optimal practical distribution can be obtained as

$$\boldsymbol{\mu}^* = \frac{\sum_{i=1}^{N} I\left(\boldsymbol{x}_i\right) \omega\left(\boldsymbol{x}_i; \boldsymbol{\mu}, \boldsymbol{\sigma}\right) \boldsymbol{x}_i}{\sum_{i=1}^{N} I\left(\boldsymbol{x}_i\right) \omega\left(\boldsymbol{x}_i; \boldsymbol{\mu}, \boldsymbol{\sigma}\right)}, \boldsymbol{\sigma}^* = \sqrt{\frac{\sum_{i=1}^{N} I\left(\boldsymbol{x}_i\right) \omega\left(\boldsymbol{x}_i; \boldsymbol{\mu}, \boldsymbol{\sigma}\right) \left(\boldsymbol{x}_i - \boldsymbol{\mu}^*\right)^2}{\sum_{i=1}^{N} I\left(\boldsymbol{x}_i\right) \omega\left(\boldsymbol{x}_i; \boldsymbol{\mu}, \boldsymbol{\sigma}\right)}}. \tag{10}$$

3 THEORY

3.1 Framework of the Proposed Method

In this section, we will describe the proposed method in detail. Figure 1 shows a general framework of the proposed method.

First, we search OSVs of each failure region by improving the method in Reference [5] with recovery technology [26]. It works well in both high-dimensional and multiple failure region cases. Second, we sample with a proposed non-Gaussian distribution, e.g., a joint GPD distribution, and evaluate these samples with SPICE simulation. The initial parameters of this joint GPD are set properly. Finally, an iterative MSP-SQP solver is applied for cross entropy minimization to find the optimal parameters in the joint GPD until the predefined FOM threshold, i.e., the accuracy of the yield estimation, is met. Note that no extra SPICE simulation is needed to solve the optimal problem.

3.2 The Proposed Non-Gaussian Distribution

Generally, a SRAM cell is considered a success (or failure) if the performance y of the SRAM cell is greater (or less) than a predefined performance threshold y_{th}. Though the relationship between process parameters $\boldsymbol{x}$ and performance y is usually complicated and nonlinear, it is usually true that in parameter space there is a boundary between the pass and fail regions, and the failure regions usually extend from the boundary to infinite as shown in Figure 5. This is the basic assumption of many failure boundary searching methods [12, 16]. In fact, for a *good* design, it is reasonable

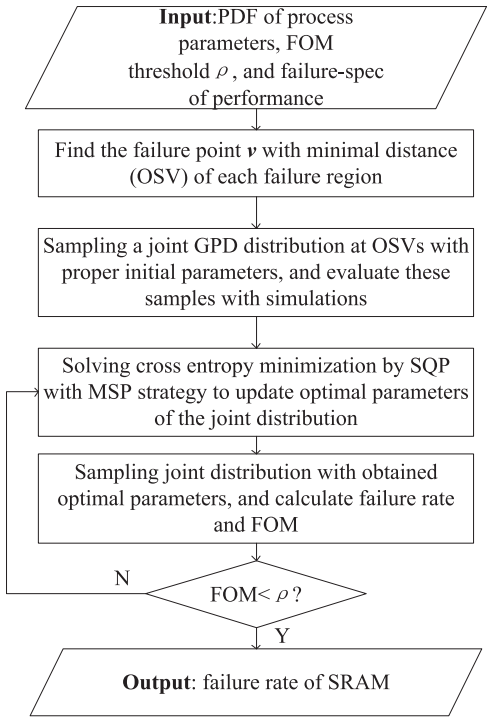

Fig. 1. The framework of the proposed method.

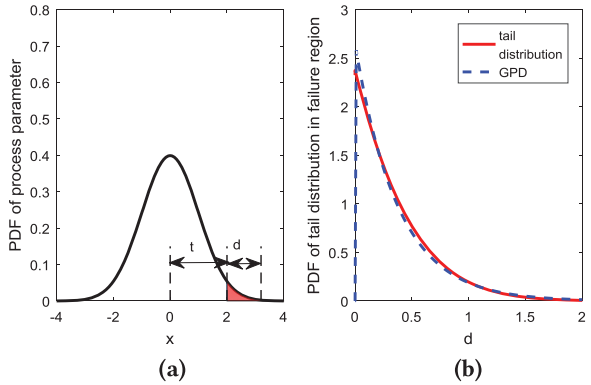

Fig. 2. Distribution of process parameter and failure boundary (a) and tail distribution of failure region and generalized Pareto distribution approximation (b) in a 1D example.

that the design point and its neighborhood (variation) should be successful, and the failure regions should be far from the design point.

For the 1D case, if there is only one failure region and the above assumption holds true, then this failure region is equivalent to $x \geq t$ without loss of generality, where t is the failure boundary as shown in Figure 2(a). The distribution of the failure region can be viewed as a normalized tail distribution of the given parameter distribution as the red solid line shown in Figure 2(b). Following

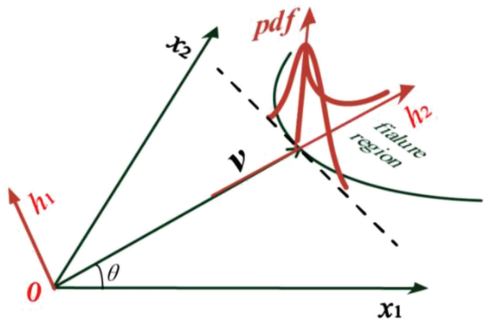

Fig. 3. A 3D schematic of Householder transformation of practical PDF.

References [22, 23], the tail distribution of many given distributions, such as normal distribution, exponential distribution, and so on, can be approached by 1D GPD accurately, as the dotted line shown in Figure 2(b).

For high-dimensional cases, it can be proved that if the failure boundary is a hyperplane $x_1 = t$, the tail distribution in failure region can be approximated by a joint distribution of 1D GPD and (n-1)-D dimensional independent distribution, i.e., Theorem 1. In addition, as t tents to infinity, the optimal sampling distribution for IS is the proposed joint GPD. The detailed proof can been found in Appendix A1.

If the boundary, i.e., the hyperplane is not perpendicular to $\boldsymbol{x}_1$ axis, then we can apply a linear Householder rotation transformation for the PDF, i.e., the corollary 1 in Appendix A2. Figure 3 gives a concrete description that the realistic failure boundary (red PDF) is formed by a counter-clockwise rotation of the hyperplane being perpendicular to x_1 axis with angle θ.

Specially, if the given PDFs of process parameters are all normal distributions, the practical failure distribution can be simplified as a joint distribution of 1D generalized Pareto distribution and (n-1)-D Gaussian distribution $p_t(\boldsymbol{H} \cdot \boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\xi})$ as Corollary 2 in Appendix A3.

For realistic SRAM, though the failure boundary should definitely be more complicated than any hyperplane, the above conclusions still give us a hint to apply a hyperplane for approximating the realistic failure boundary. This hyperplane, defined by the vector $\boldsymbol{v}$, can be approximated by the nearest point of failure boundaries to the origin, termed as OSV.

To fitting the realistic distribution in failure region well, we relax the joint distribution with freedoms of choosing distribution parameters, i.e., $\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\xi}$, by solving an optimization problem. The optimization objective is trying to minimize the *difference* between the practical failure distribution and ideal failure distribution, where the *difference* is measured by the cross entropy distance discussed in detail in the next subsection.

In other words, we apply a non-Gaussian distribution, i.e., a joint 1D generalized Pareto distribution and (n-1)-D Gaussian distribution, instead of the most frequently used n-D Gaussian distribution for sampling. This substitution will be more similar with the realistic SRAM failure distribution, and therefore it should be more efficient based on the important sampling theory.

Figure 4 gives a 6T bit cell example, i.e., the first case of Section 5 with only two variable parameters, threshold voltage *vth1* and *vth2* for M1 and M3, respectively. Both v*th1* and *vth2* are given normal distributions independently. Figure 4 gives the ideal failure PDF with read failure from 3.6×10^5 (600×600 grid) SPICE simulations, where the householder transformation has already been applied. From Figure 4, we can clearly see that the projection on *vth2* is very similar to GPD, while on *vth1,* similarly to normal distributions, which verifies the proposed joint GPD distribution.

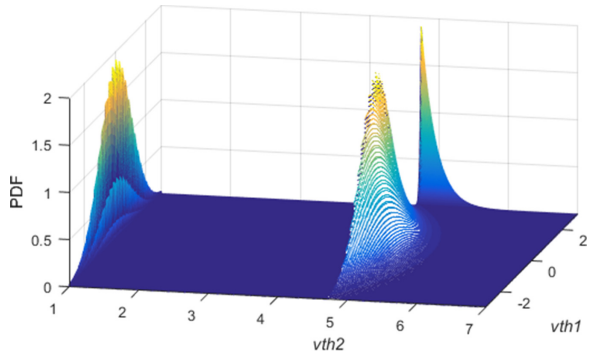

Fig. 4. An example of ideal PDF of read failure in a 6T bit cell case.

3.3 Cross Entropy Minimization

In this subsection, we will discuss how to solve the minimization of the cross entropy between the proposed joint GPD and the ideal failure distribution.

Based on the definition of cross entropy in Equation (9), the optimization objective of cross entropy minimization is

$$\begin{aligned}
&\arg\min_{\boldsymbol{\mu},\boldsymbol{\sigma},\xi} D = \arg\min_{\boldsymbol{\mu},\boldsymbol{\sigma},\xi} \int p^{ideal}(\boldsymbol{x}) \log\left(\frac{p^{ideal}(\boldsymbol{x})}{f_H(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\sigma},\xi)}\right) d\boldsymbol{x} \\
&= \arg\min_{\boldsymbol{\mu},\boldsymbol{\sigma},\xi} \left[\int p^{ideal}(\boldsymbol{x}) \log\left(p^{ideal}(\boldsymbol{x})\right) d\boldsymbol{x} - \int \frac{I(\boldsymbol{x})\, p(\boldsymbol{x})}{P_{fail}} \log\left(f_H(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\sigma},\xi)\right) d\boldsymbol{x}\right] \\
&= \arg\max_{\boldsymbol{\mu},\boldsymbol{\sigma},\xi} \int I(\boldsymbol{x})\, p(\boldsymbol{x}) \log\left(f_H(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\sigma},\xi)\right) d\boldsymbol{x} \\
&= \arg\max_{\boldsymbol{\mu},\boldsymbol{\sigma},\xi} \int \frac{I(\boldsymbol{x})\, p(\boldsymbol{x})}{f_H(\boldsymbol{x};\boldsymbol{\mu}^0,\boldsymbol{\sigma}^0,\xi^0)} \log\left(f_H(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\sigma},\xi)\right) f_H\left(\boldsymbol{x};\boldsymbol{\mu}^0,\boldsymbol{\sigma}^0,\xi^0\right) d\boldsymbol{x} \\
&\approx \arg\max_{\boldsymbol{\mu},\boldsymbol{\sigma},\xi} \sum_{i=1}^{N_0} \left[\frac{I(\boldsymbol{x}_i)\, p(\boldsymbol{x}_i)}{f_H(\boldsymbol{x}_i;\boldsymbol{\mu}^0,\boldsymbol{\sigma}^0,\xi^0)} \log\left(f_H(\boldsymbol{x}_i;\boldsymbol{\mu},\boldsymbol{\sigma},\xi)\right)\right],
\end{aligned} \tag{11}$$

where $f_H(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\sigma},\xi) = p_t(\boldsymbol{H}\cdot\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\sigma},\xi)$ substitutes the practical distribution $p^{practical}(\boldsymbol{x})$in Equation (9).

In the second equation of Equation (11), $p^{ideal}(\boldsymbol{x}) = \frac{I(\boldsymbol{x})p(\boldsymbol{x})}{P^{fail}}$, i.e., Equation (7) is applied. Though the ideal PDF $p^{ideal}(\boldsymbol{x})$ and failure rate P_{fail} are unknown, they are constant function and constant value, respectively. Therefore, they disappear in the optimization objective in the third equation. Note that $I(\boldsymbol{x})p(\boldsymbol{x})$ indicates the unknown ideal PDF before normalization, which is approximated by sampling $\boldsymbol{x_i}$ from the distribution of the function family $f_H(\boldsymbol{x};\boldsymbol{\mu}^0,\boldsymbol{\sigma}^0,\xi^0)$ in the last equation. Here, the index 0 of $\boldsymbol{\mu}^0,\boldsymbol{\sigma}^0,\xi^0$ only indicates that they are different with the optimization parameters $\boldsymbol{\mu},\boldsymbol{\sigma},\xi$.

For achieving better accuracy, the approximation of the ideal PDF can be done in an iterative manner, i.e., in the (K-1) step, we solve the optimization problem as

$$\arg\max_{\boldsymbol{\mu}^K,\boldsymbol{\sigma}^K,\xi^K} \sum_{k=0}^{K-1}\sum_{i=1}^{N_k} \left[\frac{I\left(\boldsymbol{x}_i^k\right) p\left(\boldsymbol{x}_i^k\right)}{f_H\left(\boldsymbol{x}_i^k;\boldsymbol{\mu}^k,\boldsymbol{\sigma}^k,\xi^k\right)} \log\left(f_H\left(\boldsymbol{x}_i^k;\boldsymbol{\mu}^K,\boldsymbol{\sigma}^K,\xi^K\right)\right)\right], \tag{12}$$

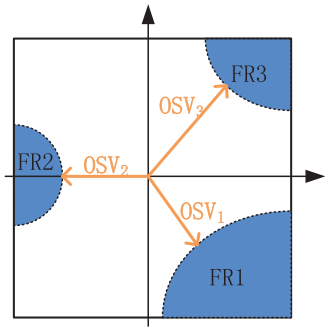

Fig. 5. A 2D example of multiple failure regions [5].

where k is the iterative step, $\boldsymbol{x}_i^k$ is the ith sample in k-step. All samples, from group 0 to group K-1, are taken into consideration, and the samples of each group follow the distribution $f_{\boldsymbol{H}}(\boldsymbol{\mu}^k, \boldsymbol{\sigma}^k, \xi^k)$, where $\boldsymbol{\mu}^k, \boldsymbol{\sigma}^k, \xi^k$ are the optimal parameters solved at the (k-1) step and $\boldsymbol{\mu}^0, \boldsymbol{\sigma}^0, \xi^0$ are given constants as initial values. Indeed, the ideal distribution can be thought as being constructed by all these groups of samples, i.e., by all distributions $f_{\boldsymbol{H}}(\boldsymbol{\mu}^k, \boldsymbol{\sigma}^k, \xi^k)$, $k = 0 \ldots K$-1, in an even weight manner.

The constrain of the optimization problem is given as

$$s.t. \quad \boldsymbol{\sigma} > \mathbf{0}. \tag{13}$$

We use SQP 0 to solve the above optimization problem in Equation (12) with the constraint of Equation (13). More details can be found in Section 4.1. Note that the optimization parameters $\boldsymbol{\mu}^K, \boldsymbol{\sigma}^K, \xi^K$ of the Kth step are solved over the already existing samples in the ($0 \ldots K$-1) groups, and therefore it is free in the sense of SPICE simulations.

3.4 Multiple Failure Regions

The above discussion is only suitable for cases with only one failure region. If there are multiple failure regions in the parameter space as shown in Figure 5, then the proposed joint GPD will suffer changes to a mixture form.

If all OSV $v_m, m = 1, \ldots, M$ have been found, then both References [5] and [9] take a mixture normal distribution as practical distribution, i.e.,

$$p^{practical}(\boldsymbol{x}) = \sum_{m=1}^{M} \omega_m \cdot N(\boldsymbol{x}; \boldsymbol{\mu}_m), \tag{14}$$

where ω_m is

$$\omega_m = \frac{\exp\left(-\frac{1}{2}\|v_m\|^2\right)}{\sum_{m=1}^{M_2} \exp\left(-\frac{1}{2}\|v_m\|^2\right)}. \tag{15}$$

It seems reasonable that ω_m becomes smaller as $\|v_m\|$ becomes larger. However, such intuition has no mathematical guarantee.

Similarly, we construct the practical mixture distribution for multiple failure regions as

$$p^{practical}(\boldsymbol{x}) = \sum_{m=1}^{M} \omega_m \cdot f_{\boldsymbol{H}_m}(\boldsymbol{x}; \boldsymbol{\mu}_m, \boldsymbol{\sigma}_m, \xi_m), \tag{16}$$

where the weight coefficient ω_m is

$$\omega_m = \frac{P_m^{fail}}{P^{fail}}, \quad P^{fail} = \sum_{i=1}^{m} P_m^{fail}, \tag{17}$$

where P_m^{fail} is the failure rate of the mth failure region and P^{fail} is the total failure rate. Because we use an iterative manner for parameter estimations, the failure rate of every failure region and total failure rate can be easily obtained and updated from the last iteration, and the initial value of ω_m is simply given by Equation (15). The change of the weight coefficient ω_m is based the fact that if there is no overlap among failure regions, the mixture form will be exactly correct with weight coefficient ω_m in form of Equation (17), i.e., the Theorem 2 in Appendix A4.

4 IMPLEMENTATION

4.1 MSP-SQP Algorithm for Cross Entropy Minimization

SQP is an iterative method for nonlinear optimization 0. The basic idea of SQP is modeling the problem at a given approximate solution $\boldsymbol{x}_j$ by a Quadratic Programming sub problem and using the solution of this sub problem to construct a better approximations $\boldsymbol{x}_{j+1}$, where j and j+1 are the iterative steps, and this process continues until the approximation converges to a local minimal solution $\boldsymbol{x}^*$.

The optimization problem is defined as

$$\begin{aligned} &\min_{\boldsymbol{x}} f(\boldsymbol{x}) \\ &\text{s.t.} \quad \boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{0}, \end{aligned} \tag{18}$$

where $f: R^n \to R$, $g: R^n \to R^m$, $\boldsymbol{x}: R^n$.

Given the cost function (18), a Lagrangian function is constructed as

$$L(\boldsymbol{x}, \boldsymbol{v}) = f(\boldsymbol{x}) + \boldsymbol{v}^T \boldsymbol{g}(\boldsymbol{x}), \tag{19}$$

where $\boldsymbol{v}$ is an estimate of Lagarange multipliers.

The quadratic programing (QP) problem is usually expressed as follows:

$$\begin{aligned} &\min_{\boldsymbol{d}_j} \quad \nabla f(\boldsymbol{x}_j)^T \boldsymbol{d}_j + \frac{1}{2} \boldsymbol{d}_j^T \boldsymbol{B}_j \boldsymbol{d}_j \\ &\text{s.t.} \quad \nabla \boldsymbol{g}_i(\boldsymbol{x}_j)^T + \boldsymbol{g}_i(\boldsymbol{x}_j) \le \boldsymbol{0}, \end{aligned} \tag{20}$$

where $\boldsymbol{B_j}$ denotes the quasi-Newton approximation to Hessian matrix of Lagrangian function (19).

The next iteration approximations $\boldsymbol{x}_{j+1}$ is generated as

$$\boldsymbol{x}_{j+1} = \boldsymbol{x}_j + \alpha_j \boldsymbol{d}_j, \tag{21}$$

where α_j is the step length determined by a line search procedure.

Both Jacobian and Hessian metrics can be approximated with first-order derivatives 0. Hence, the complexity of the MSP-SQP framework is linear with respect to the dimensionality, which makes it linearly extendable to high-dimension cases.

However, SQP are only guaranteed to find a local solution of Equation (18), and to avoid trapping in the local optimization, a multiple starting-point strategy is further applied. Typically, we solve the above optimization problem with multiple, e.g., 100, starting points, which are generated randomly around the starting point $\boldsymbol{\mu}^0$, $\boldsymbol{\sigma}^0$, ξ^0. The best result of local optimizations will be chosen after 100 times repeated optimizations are completed.

4.2 Setting Initial Parameters

The initial parameters $\boldsymbol{\mu}^0, \boldsymbol{\sigma}^0, \xi^0$ of the joint GPD are chosen with different strategies. The (n-1)-D Gaussian distribution is simply set as standard Gaussian distributions, i.e.,

$$\left[\mu_2^0, \ldots, \mu_n^0\right] = [0, \ldots, 0], \left[\sigma_2^0, \ldots, \sigma_n^0\right] = [1, \ldots, 1]. \tag{22}$$

For the 1D GPD, the initial parameters $\mu_1^0, \sigma_1^0, \xi^0$ are calculated by solving the following sub-optimization problem for a better estimate:

$$\arg\min_{\mu_1, \sigma_1, \xi} D = \arg\min_{\mu_1, \sigma_1, \xi} \int p_{1,t}(h_1) \log\left(\frac{p_{1,t}(h_1)}{g(h_1; \mu_1, \sigma_1, \xi)}\right) dh_1, \tag{23}$$

where $p_{1,t}(h_1)$ is the 1D tail distribution of $p_1(h_1)$ as

$$p_{1,t}(h_1) = \begin{cases} \frac{p_1(h_1)}{\int_{\|\boldsymbol{v}\|}^{+\infty} p_1(h_1) dh_1} & h_1 > \|\boldsymbol{v}\| \\ 0 & h_1 \le \|\boldsymbol{v}\| \end{cases}, \tag{24}$$

where v is the same vector in Householder transformation (39). Since $p_1(h_1)$ is given and $\boldsymbol{v}$, i.e., OSV, is calculated at this time, $p_{1,t}(h_1)$ can be easily calculated numerically from Equation (24)

Then the initial parameters $\boldsymbol{\mu}^0$ and $\boldsymbol{\sigma}^0$ are then combined together as

$$\boldsymbol{\mu}^0 = \left[\mu_1^0, \mu_2^0, \ldots, \mu_n^0\right], \quad \boldsymbol{\sigma}^0 = \left[\sigma_1^0, \sigma_2^0, \ldots, \sigma_n^0\right]. \tag{25}$$

4.3 Other Non-Gaussian Distribution

Although we choose the joint GDP distribution as the function family $f_{\boldsymbol{H}}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}, \xi)$, it can also be replaced with other Gaussian or non-Gaussian function families, for example, the normal distribution (ND), which is adopted in References [6] and [7], or the split normal distribution (SND).

The 1D split normal distribution joins two halves of normal distributions with the same mean but with different standard deviations, σ_1 for left and σ_2 for right,

$$N(x; \mu, \sigma_1, \sigma_2) = \frac{\sqrt{2/\pi}}{\sigma_1 + \sigma_2} \begin{cases} \exp\left(-\frac{(x-\mu)^2}{2\sigma_1^2}\right) & x < \mu \\ \exp\left(-\frac{(x-\mu)^2}{2\sigma_2^2}\right) & x \ge \mu \end{cases}. \tag{26}$$

The high-dimensional split normal distribution is

$$N(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2) = \prod_{d=1}^{D} N(x_d; \mu_d, \sigma_{d1}, \sigma_{d2}), \tag{27}$$

where $\boldsymbol{x}$, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ are D-dimensional vectors and their entries are x_d, μ_d, σ_{d1}, and σ_{d2}, respectively.

A schematic result of the above cross entropy minimization method is shown in Figure 6. In this case, if the proposed joint GPD is applied, then the algorithm will end within two iterative steps. Therefore, we choose SND as the function family, for its slow convergence will make the iteratively approaching process more clear.

In Figure 6, the blue dotted line is the given original PDF of process parameters, and the black solid line is the ideal and unknown PDF of failure region. Based on the PDF of the failure region, efficient samples should be in the fail region and near the boundary. In the $k = 0$ step, the practical PDF (red dotted line) is shifted from standard normal PDF to near the failure boundary for sampling. The histogram is the samples under current practical PDF of failure region $N(\boldsymbol{\mu}, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)$, which is not efficient enough. However, after optimizing $\boldsymbol{\mu}$, $\boldsymbol{\sigma}_1$, $\boldsymbol{\sigma}_2$ and resampling iteratively, i.e., k = 1,2,3, the practical distributions of the failure region become closer and closer to the ideal

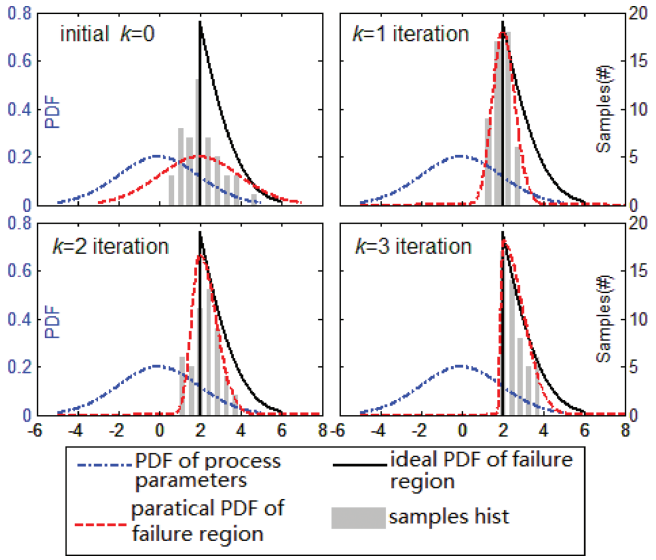

Fig. 6. A schematic approaching process of cross entropy minimization.

distribution of failure region, and the histograms of samples show that they become more and more efficient, which verifies the effectiveness of the proposed iterative cross entropy optimization method.

4.4 Improvement of Searching OSV

To find all OSVs of multiple failure regions, we adopt the strategy from MFRIS [5]. Searching OSV is equivalent to solving the following optimization problem

$$\begin{aligned} &\min \|\boldsymbol{v}\| \\ &s.t.\ \ y_i(\boldsymbol{v}) = perf_i(\boldsymbol{v}) - spec_i \le 0, i = 1, 2, \ldots, m, \end{aligned} \tag{28}$$

where $\boldsymbol{v}$ is the point in parameter space, $\|\cdot\|$ is the 2-norm, $perf_i(\boldsymbol{v})$ is the ith performance obtained from SPICE simulations, and $spec_i$ is the ith threshold for distinguishing success and failure. This optimization problem means to find the failure point with the minimal distance to the origin.

In Reference [5], searching OSV is composed of two phases: (1) global phase: explore the variation space with Sobol sequence and cluster failure points to find multiple failure regions; and (2) local phase: find the OSV of each corresponding failure regions with SQP starting from the centroid of these failure regions.

Although SQP is a reliable method for local search, for each gradient evaluation with n variables, n times of SPICE simulations are needed, which is almost unaffordable for problems with hundreds or even thousands of variables. To address this issue, we apply the idea of sparse recovery in Reference [26] to recover the sparse gradients, even if the number of SPICE simulations is less than n.

The basic idea of sparse recovery is to approximate finite difference with directional derivatives for a performance $y(\boldsymbol{x})$

$$\boldsymbol{A} \cdot grad(\boldsymbol{x}_0) = \boldsymbol{b}, \tag{29}$$

$$\boldsymbol{A} = \begin{bmatrix} (\boldsymbol{x}_1 - \boldsymbol{x}_0)^T \\ (\boldsymbol{x}_2 - \boldsymbol{x}_0)^T \\ \cdots \\ (\boldsymbol{x}_t - \boldsymbol{x}_0)^T \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} y(\boldsymbol{x}_1) - y(\boldsymbol{x}_0) \\ y(\boldsymbol{x}_2) - y(\boldsymbol{x}_0) \\ \cdots \\ y(\boldsymbol{x}_t) - y(\boldsymbol{x}_0) \end{bmatrix}, \tag{30}$$

where $\boldsymbol{x}_0$ is the point whose gradient $grad(\boldsymbol{x}_0)$ is required, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$ are the selective SPICE simulation points near $\boldsymbol{x}_0$ with Gram–Schmidt process to filter correlative points, and $y(\boldsymbol{x}_1), \ldots, y(\boldsymbol{x}_t)$ are performances obtained from corresponding SPICE simulations.

For saving SPICE simulations, there is $t < n$ and (29) is underdetermined. Considering the fact that the performance of the circuit is not sensitive to all variables, the gradient is usually sparse. Thus, the gradient can be estimated by solving the following optimization problem:

$$\begin{aligned} &\min \quad \|grad(\boldsymbol{x}_0)\|_1 \\ &s.t. \quad \boldsymbol{A} \cdot grad(\boldsymbol{x}_0) = \boldsymbol{b}, \end{aligned} \tag{31}$$

where $\|grad\|_1$ is the L1 norm of grad. Equation (31) can be solved by use of the interior-point method [29].

Therefore, combining the sparse recovery technology with OSV searching algorithm in Reference [5], the CPU time needed can decrease about 40%.

4.5 Flow of Proposed Algorithm

The overall flow of proposed method is summarized in Algorithm 1.

ALGORITHM 1: Cross Entropy Minimization with non-Gaussian Distribution.

Input: joint PDF of process parameters $p(x)$, FOM threshold ρ_{th}
Output: SRAM failure rate $\hat{P}$ and FOM

1: Find all OSVs $v_m, m = 1, \ldots, M$ for each failure region following the method in Section 4.4
2: For each $\boldsymbol{v}_{\mathrm{m}}$, calculate transformation matrix $\boldsymbol{H}_{\mathrm{m}}$ by (39)
3: Set iterative step $k = 0$, the practical mixture distribution as (16) and (17) with initial parameters as (25)
4: **do**
5: **for** m = 1 to M **do**
6: Generate $N_{km} = N_k \omega_m$ samples $\boldsymbol{x}_i^k, i = 1, \ldots, N_{km}$ following $f_{\boldsymbol{H}_m}(\boldsymbol{x}; \boldsymbol{\mu}_m^k, \boldsymbol{\sigma}_m^k, \xi_m^k)$.
7: Evaluate these samples $\boldsymbol{x}_i^k$ by SPICE.
8: Calculate the failure rate of each failure region $\hat{P}_m^{fail}$ and total failure rate $\hat{P}^{fail}$ by Equation (6), and FOM by Equation (4) from the samples on the k-th iteration $\boldsymbol{x}_i^k, i = 1, \ldots, N_{km}$.
9: solve the optimization problem of cross entropy minimization in Equation (12) by MSP-SQP by all samples $\{\boldsymbol{x}_i^{\mathrm{k}}, \mathrm{k} = 1..k\}$ of all iterative steps, and obtain the optimal parameters $\boldsymbol{\mu}_m^{k+1}, \boldsymbol{\sigma}_m^{k+1}, \xi_m^{k+1}$ of practical distribution on the (k+1)-th step.
10: **end for**
11: $k = k + 1$
12: update the weight factor ω_m^{k+1} by Equation (17)
13: **until** FOM < ρ_{th}
14: **return** failure rate $\hat{P}$ and FOM

In line 1, we apply the method proposed in Section 4.4 to find all OSVs of every failure region.

In lines 2 and 3, after all OSVs are obtained, the transformation matrix $\boldsymbol{H}_{\mathrm{m}}$ and mixture distribution with all initial parameters can be set.

In lines 6–9, in each iterative step, we sample on each failure region following the joint GPD and evaluate it by SPICE simulation. The number of samples is proportional to their latest ratios

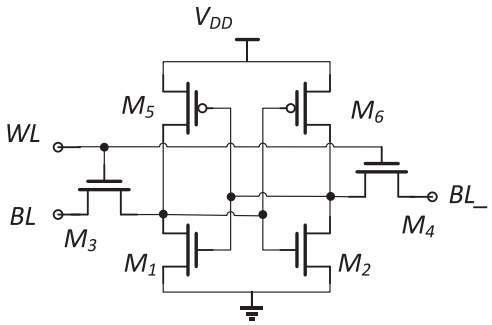

Fig. 7. Circuit schematic of a 6T SRAM cell.

of failure rates. Then we update the optimal parameter $\boldsymbol{\mu}_m^{k+1}, \boldsymbol{\sigma}_m^{k+1}, \xi_m^{k+1}$ by solving the problem of cross entropy minimization with MSP-SQP. The iteration repeats until the accuracy FOM is good enough.

5 EXPERIMENT RESULTS

The proposed method is verified by a 6T SRAM bit cell in low dimension (6D) with a single failure region and by a SRAM column in high dimensions (192D) with multiple failure regions. All test cases are under a 28nm CMOS process. The results are compared with several state-of-the-art methods. Since all methods are intrinsically stochastic, we repeatedly run 100 estimations to acquire the mean and deviation for fair comparisons. The golden results are obtained by the Monte Carlo method with enough samples. The relative error of mean reflects the accuracy, and the deviation represents its reliability and stability. FOM is set to 0.0865 for ending loops.

5.1 A Low-Dimensional Case: 6T SRAM Bit Cell

Figure 7 shows the circuit schematic of a 6T SRAM bit cell. The read current, the difference between bit line current I_{BL} and $I_{BL\_}$, considered as the interest performance, must be larger than a given threshold for "*Success.*" The same as Reference [5], there are six independent random variables with normal distributions for the threshold voltages V_{TH} of six transistors. To verify the efficiency of different function families, normal distribution (ND), split normal distribution (SND), and joint GPD distribution (GPD) are compared in the same proposed algorithm framework.

In Table 1, Mean and Dev are the mean and standard deviation of all methods with 100 repeated runnings and are not the mean or standard deviation of sampling distributions (e.g., ND, SND, GPD). MRE is the mean of relative error of every running, and the golden result is from 10^8 MC simulations, obtained from a 20-core computer with about 18 days. #Total is the total number of SPICE simulations.

From Table 1, we can clearly see that MFRIS is the most efficient one among the already-existing methods. Our proposed GPD method produces a 1.58× (400/252) speedup over MFRIS on the total number of SPICE simulations. All three function families, ND, SND, and GPD, under the same cross entropy minimization framework can work well, and they achieve high accuracy, i.e., around 5–6% relative error, with smaller standard deviations. Note that solving the cross entropy minimization by MSP-SQP needs no SPICE simulations and needs only about 0.04min of CPU time, accounting for only 3% of the total time. The average iterative steps of solving the cross entropy minimization are 1.2.

Table 1. Read Failure Rate of a 6T SRAM Bit Cell

Method	Mean (10^{-6})	Dev (10^{-7})	MRE (%)	#Total	CPU time (min)
Monte Carlo	1.20	-	-	10^8	18 days on 20 cores
SS [4]	1.16	1.18	7.73	1820	9.10
SUS [10]	1.61	23.5	168	2978	14.98
CEM[6]	1.16	0.93	6.21	2310	11.50
[7]	1.18	1.04	7.84	1850	9.13
IBS [9]	1.22	1.44	9.56	671	3.46
MFRIS[5]	1.17	0.86	6.27	400/888	2.06
ND	1.23	0.80	5.36	730	3.89
SND	1.21	0.89	5.70	477	2.55
GPD	1.18	0.84	5.53	252	1.38

Table 2. Number of SPICE Simulation of Two Stages

Method	#OSV	#Sampling	#Total	Speedup (Sampling)	Speedup (Total)
MFRIS[5]	130	270/758	400/888	1	1
ND	130	600	730	0.45	0.55
SND	130	347	477	0.77	0.83
GPD	130	122	252	2.21/6.21	1.58/3.52

The proposed GPD method can be roughly divided into two successive stages: finding OSVs and sampling near OSV. Table 2 gives numbers of samplings in detail, where #OSV and #Sampling are the numbers of finding OSVs and sampling, respectively.

From Table 2, we can see that the proposed GPD method produces a 2.21× (270/122) speedup over MFRIS on sampling near OSVs, even if MFRIS uses a trained surrogate model inside [5]. Otherwise, our method can gain 6.21× (758/122) speedup on sampling. There is no need for GPD to build a surrogate model, for it needs only 122 samples, which is not enough to train the model successfully. In the whole of two stages, the proposed GPD produces a 1.58× speedup over MFRIS.

Figure 8(a) compares convergence speeds of failure rate vs. simulation number by different methods in detail. MFRIS and IBS apply SPICE surrogate models for saving SPICE samples. Once the models are trained accurately enough, the number of SPICE simulation will be constant. The number of SUS is fixed and is hard to increase gradually, so results of SUS are not listed. From Figure 8(a), it can be seen that the proposed GPD method can converge most quickly.

To show the convergence speed more clearly, Figure 8(b) gives FOM failure rate vs. simulation number by different methods. SS, CEM [6, 7], MFRIS, and ND are Gaussian distribution-based sampling. They converge together when the number of samplings is high enough and the accuracy required is very high (FOM>10^{-2}). MFRIS shows higher convergence speed only because it applies the trick of surrogate model, which, however, will be invalid on higher-dimensional cases as shown later.

Both SND and GPD, being with non-Gaussian distributions, show remarkable speedup of convergence over Gaussian distribution-based methods. The proposed joint GPD distribution, although proved to be optimal only for a failure boundary, is a hyperplane, beating the existing best method, MFRIS, with a 1.6× speedup as FOM=0.0865. The experimental results also verify that choosing the suitable function family is one of the key factors for efficient sampling.

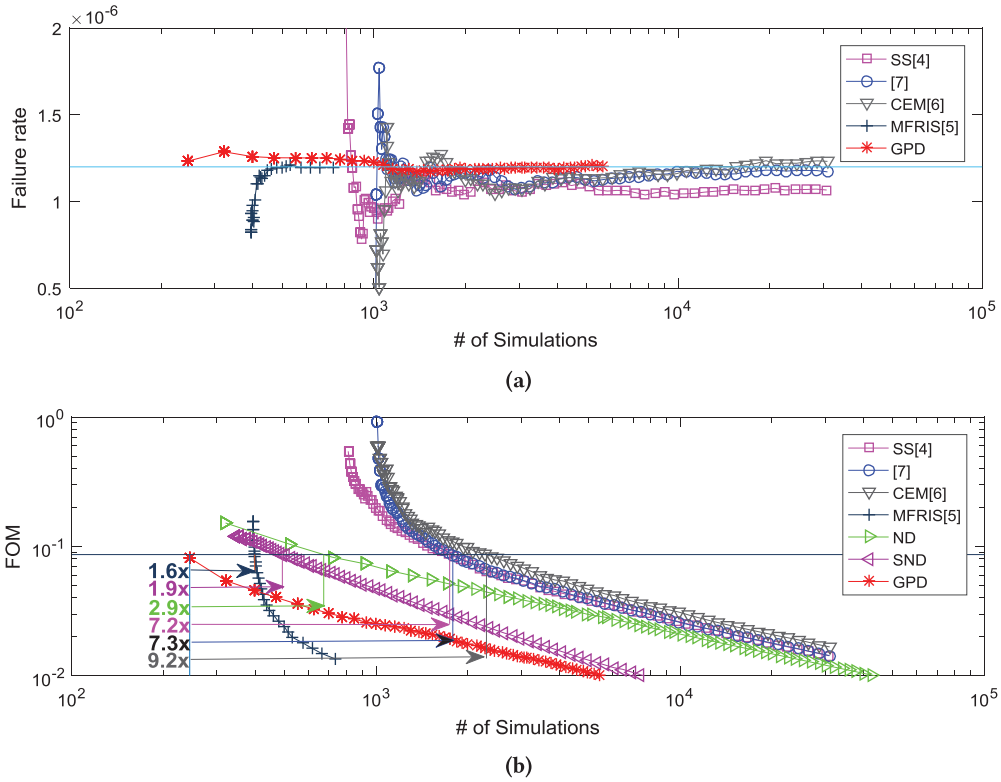

Fig. 8. Convergence speed for different methods in the 6D case.

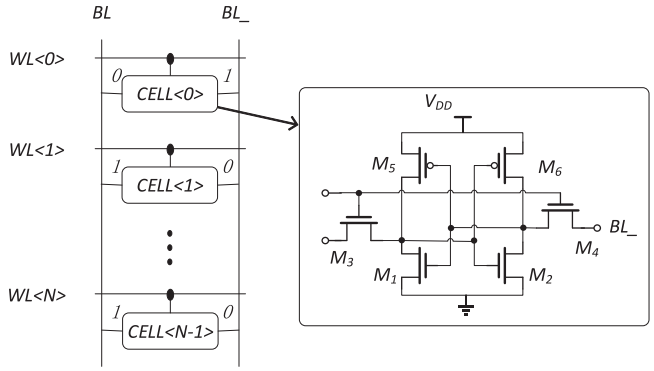

Fig. 9. Circuit schematic of a SRAM column.

5.2 High-Dimensional and Multiple-Failure Region Test Cases: SRAM Column

To verify the proposed method in high-dimensional and multiple-failure region cases, the failure rate estimations of an SRAM column with 192D is tested.

Figure 9 shows the circuit schematic of a SRAM column consisting of N cells. Besides the read current, the read node voltage of cell<0> is also added into consideration. We store "0" in cell<0> and "1" in all other cells to maximize the impact of leakage current [10]. During the read operation,

Table 3. Read Failure Rate of a SRAM Column

Method	Mean (10^{-6})	Dev (10^{-7})	MRE (%)	#Total	CPU time (min)
MC	2.10	-	-	5×10^7	29 days on 20 cores
SS [4]	-	-	-	>20,000	-
SUS [10]	2.24	8.23	32.7	19,379	321.0
CEM [6]	-	-	-	>20,000	-
[7]	-	-	-	>20,000	-
IBS [9]	-	-	-	>20,000	-
MFRIS [5]	2.04	1.77	7.15	2,655	44.15
ND	2.05	1.65	6.08	2,126	36.40
SND	1.99	1.78	7.82	1,293	22.64
GPD	2.03	1.85	7.11	1,136	20.01

Table 4. Number of SPICE Simulation of Two Stages

Method	#OSV	# Sampling	#Total	Speedup (Sampling)	Speedup (Total)
MFRIS [5]	1295	1,360	2,655	**1**	1
ND	806	1,320	2,126	**1.03**	1.25
SND	806	487	1,293	**2.79**	2.05
GPD	806	330	1,136	**4.12**	2.34

after the bit line I_{BL} is charged, the node voltage will increase to V_{read} due to the voltage division between the access transistor and poll-down transistor. If V_{read} is higher than the trip voltage V_{trip} of the inverter, then the cell flips, and the stored data in the cell are corrupted, and a read failure occurs. All threshold voltages V_{TH} of transistors are independent normal random variables.

Let $N = 32$, and thus the test case has 192 (6 transistors/cell $\times$ 32 cells) random variables. The golden failure rate is estimated by a Monte Carlo method with 5×10^7 samples needing about 29 days on 20 CPU cores. Table 3 summarizes the failure rate estimated by all methods, where the convergence condition is FOM=0.0865 except SUS, because its simulation number of each step is predefined.

From Table 3, we can clearly see that in this high-dimensional case and with a limitation of maximal 20,000 samples, only SUS, MFRIS and our proposed methods can obtain an unbiased estimation of failure rate. SS, CEM, and the method in Reference [7] fail, for they cannot find the suitable OSV efficiently, and IBS fails, for it cannot build a suitable model in high-dimensional cases within limited training data. The proposed GPD method has about a 2.34× (2655/1136) speed-up over MFRIS with similar accuracy, which verifies that the proposed method can offer accurate and stable estimation in high-dimensional cases.

In GPD, the average iterative steps of solving the cross entropy minimization are 2.4. The CPU time is 0.9min, accounting for about 5% of the total CPU time. No extra SPICE samples are needed. Again, the cost of cross entropy optimization is negligible.

Since Solido's trial version is only available for company, we compare the results with a very similar 195D flip-flop case from Solido OptIS [11], where Solido OptIS needs about 5,000 samples. In our proposed GPD method for a 192D SRAM case, it shows a 4.4× (5,000/1,136) speedup over Solido OptIS.

Table 4 gives the numbers of simulations for MFRIS and three proposed methods in detail. From Table 4, the proposed GPD method produces a 2.34× speedup over MFRIS on the total number of simulations and a 4.12× speedup on sampling.

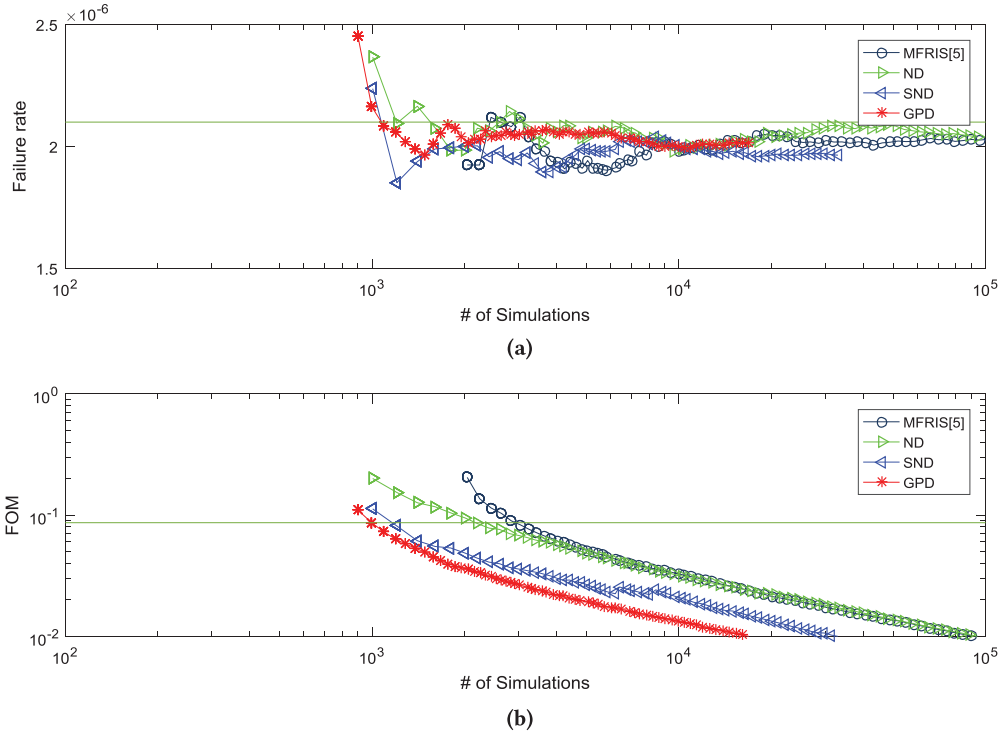

Fig. 10. Convergence speed for different methods in the 192D case.

Figure 10(a) and (b) compare failure rate and FOM vs. simulation number, respectively, with different methods, where SUS is not listed, for it needs a fixed number of samplings. Again, the proposed GPD method shows the best convergence speed over the other methods.

5.3 Accuracy Versus OSV's Variation

In this subsection, we will study how the accuracy of failure rate is affected by the OSV's errors.

Figure 11 gives the mean relative error of failure rate vs. relative error of OSV by use of ND, SND, and GPD for the 6T bit cell case. The x-axis is the relative error of OSV obtained by artificially injecting variance into the golden OSV, which is obtained from Reference [5] with very fine iterative step. The y-axis is the relative error of failure rate. To suppress the randomness of y, the mean of relative error (MRE) of 20 repeated estimations is calculated.

From Figure 11, we can see that to obtain 10% relative error of failure rate estimation, ND is the most robust for its tolerance width of variance OSV is 23.6%. SND is the worst one, with only 9.5%. GPD is between them with 15.0%. After checking all test cases, the practical relative errors of OSVs obtained in this article are less than 2%. Therefore, even the narrowest tolerance region of SND is still big enough.

However, if we cannot calculate OSV accurately, for example, reaching a maximal 20% relative error of OSV, then the accuracy of failure rate of ND decays gradually, but the accuracies of SPD (both with positive and negative deviation) and GPD (with positive deviation only) degenerate to 60%. It shows that if sampling points are far from the failure boundary, it will be difficult for the cross entropy method to converge gradually to the boundary, even if the iterative strategy is already applied.

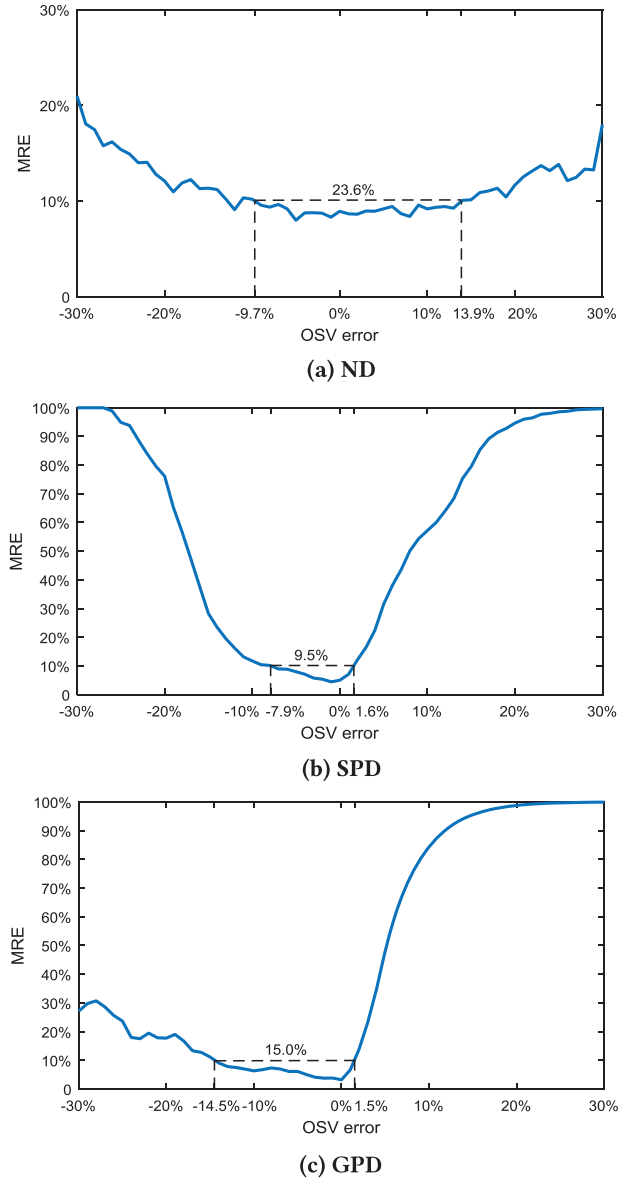

Fig. 11. Accuracy of failure rate vs. relative error of OSV for ND (a), SND (b), and GPD(c).

A greater potential threat is from the symmetry. Due to the symmetry of normal distribution, the tolerance width of ND's OSV variance is roughly balanced, from −9.7% to 13.9%. However, the tolerance width of GPD is severe asymmetric, from −14.5% to 1.5%, for its mathematical expression is asymmetric. Intuitively, it suggests that we should approach the failure boundary from the successful region (inside) and not from the failure region (outside).

6 CONCLUSIONS

An efficient non-Gaussian sampling method is proposed for efficient SRAM failure rate estimation. By setting the practical distribution as a joint GPD and applying SQP with MSP strategy to

minimize the cross entropy between practical and ideal distribution, our proposed method can obtain the parameters of practical distributions efficiently. Because the joint GPD is much more similar to the ideal distribution, it can estimate the failure rate of SRAM with high convergence speed and fewer SPICE simulations. Experimental results demonstrate the efficiency and effectiveness of the proposed method.

APPENDIX

A.1 The Proof of Theorem 1

LEMMA 1. *For a large class of distributions (including normal, exponential, gamma, t-distributions, and beta distributions [18]) with the cumulative distribution function (CDF) H(x), GPD is the limiting distribution for tail distribution of anyone in the large class of distributions, as the threshold t tends to infinity, i.e.,*

$$\lim_{t\to\infty} sup_{d\geq 0} \left|H_t\left(d\right) - G\left(d;\sigma,\xi\right)\right| = 0, \tag{32}$$

where $H_t(d)$ is the CDF of tail distributions, which is defined as the probability of $X \geq d + t$ under the condition of $X \geq t$, i.e.,

$$H_t(d) = P\left\{X - t \geq d | X \geq t\right\} = \frac{H\left(d+t\right) - H\left(t\right)}{1 - H\left(t\right)} \text{ for } d \geq 0. \tag{33}$$

$G(d;\sigma,\xi)$ and $g(d;\sigma,\xi)$ are the CDF and PDF of GPD, respectively. $g(d;\sigma,\xi)$, as a two-parameter PDF distribution, with scale σ and shape ξ, is defined as

$$g\left(d;\sigma,\xi\right) = \frac{1}{\sigma}\begin{cases}\left(1+\frac{\xi d}{\sigma}\right)^{-\frac{\xi+1}{\xi}} & \xi \neq 0\\ e^{-\frac{d}{\sigma}} & \xi = 0\end{cases}, \tag{34}$$

where $\sigma > 0$ and the support is $d \geq 0$ when $\xi \geq 0$ and $0 \leq d \leq -\frac{\sigma}{\xi}$ when $\xi < 0$. The GPD subsumes three different distributions under its parameter $\xi > 0$, $\xi < 0$, $\xi = 0$.

GPD can be extended by adding an extra parameter μ, which shifts GPD right with μ, i.e., $g(d;\mu,\sigma,\xi) = g(d-\mu;\sigma,\xi)$.

THEOREM 1. *If the tail region of a D-dimensional independent distribution $p(\boldsymbol{x}) = \prod_{d=1}^{D} p_d(x_d)$ is defined by $x_1 > t$, then the tail distribution of tail region can be approximated as*

$$p_t(\boldsymbol{x}) \approx g\left(x_1;\mu_1,\sigma_1,\xi\right) \cdot \prod_{d=2}^{D} p_d\left(x_d\right), \tag{35}$$

where $g(x_1;\mu_1,\sigma_1,\xi)$ is the PDF of generalized Pareto distribution and $\boldsymbol{x} = [x_1,\ldots,x_D]^T$.

Based on the definition of the tail region defined in Theorem 1, the PDF of tail distribution is

$$\begin{aligned} p_t(\boldsymbol{x}) &= \frac{I\left(x_1 > t\right) p\left(\boldsymbol{x}\right)}{P_t} = \frac{I\left(x_1 > t\right) \prod_{d=2}^{D} p_d\left(x_d\right)}{P_t} \\ &= \prod_{d=2}^{D} p_d\left(x_d\right) \cdot \frac{I\left(x_1 > t\right) p_1\left(x_1\right)}{P_t} = \prod_{d=2}^{D} p_d\left(x_d\right) \cdot \begin{cases}\frac{p_1\left(x_1\right)}{P_t} x_1 > t\\ 0 x_1 \leq t\end{cases}. \\ &\approx g\left(x_1;\mu_1,\sigma_1,\xi\right) \cdot \prod_{d=2}^{D} p_d\left(x_d\right), \end{aligned} \tag{36}$$

where $I(x_1 > t)$ is the indicator function as in Equation (2). P_t is the volume of tail region, i.e., $P_t = \int_t^{+\infty} p_1(x_1)dx_1 \cdot \int \cdots \int_{-\infty}^{+\infty} \prod_{d=2}^{D} p_d(x_d)dx_2 \cdots dx_n$. In the last approximate equation lemma 1 is applied for the x_1 axis.

A.2 Householder transformation

Corollary 1. *If the tail region of a D-dimensional independent distribution* $p(\boldsymbol{x}) = \prod_{d=1}^{D} p_d(x_d)$ *is outside a hyperplane, defined using a vector* $\boldsymbol{v}$*, i.e., the tail region is* $\{\boldsymbol{x}|\boldsymbol{n} \cdot (\boldsymbol{x} - \boldsymbol{v}) > 0, \boldsymbol{n} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\}$*, then the tail distribution of this tail region can be approximated with a Householder transformation* $\boldsymbol{H}$ *as*

$$p_t(\boldsymbol{H} \cdot \boldsymbol{x}) \approx g(h_1; \mu_1, \sigma_1, \xi) \cdot \prod_{d=2}^{D} p_d(x_d), \tag{37}$$

$$\boldsymbol{h} = \boldsymbol{H} \cdot \boldsymbol{x} = [h_1, \ldots, h_D]^T, \tag{38}$$

$$\boldsymbol{H} = \boldsymbol{I} - 2\boldsymbol{u}\boldsymbol{u}^T, \boldsymbol{u} = \frac{(v_1 - a, v_2, \ldots, v_D)^T}{\left\|(v_1 - a, v_2, \ldots, v_D)^T\right\|}, a = \|\boldsymbol{v}\|, \tag{39}$$

where $\boldsymbol{I}$ *is the identity matrix.*

A.3 The Proposed Joint GPD

Corollary 2. The tail region of a D-dimensional standard normal distribution $p(\boldsymbol{x}) = \prod_{d=1}^{D} N(x_d; \mu_d, \sigma_d)$ can be approached as

$$p_t(\boldsymbol{H} \cdot \boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}, \xi) \approx g(h_1; \mu_1, \sigma_1, \xi) \cdot \prod_{d=2}^{D} N(h_d; \mu_d, \sigma_d), \tag{40}$$

where $g(h_1; \mu_1, \sigma_1, \xi)$ is generalized Pareto distribution on the first dimension, $N(h_d; \mu_d, \sigma_d)$ is the normal distribution on the dth dimension, $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_D]$, $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_D]$, $\boldsymbol{H} \cdot \boldsymbol{x} = \boldsymbol{h}, \boldsymbol{x} = \boldsymbol{H}^{-1} \cdot \boldsymbol{h}$, $\boldsymbol{H}$ and $\boldsymbol{H}^{-1}$ are the Householder and its reverse transformation respectively from point $\boldsymbol{x} = [\mathrm{x}_1, \ldots, \mathrm{x}_D]$ to $\boldsymbol{h} = [h_1, \ldots, h_D]$.

A.4 The Proof of Theorem 2

Theorem 2. *Given a distribution* $\mathrm{p}(\mathbf{x})$ *of process parameter* $\boldsymbol{x}$*, if there are m failure regions without overlap among them, then the ideal distribution and failure rate for* m*th failure region are* $p_m^{ideal}(\boldsymbol{x})$ *and* $P_m^{fail} = \int_{-\infty}^{+\infty} I(\boldsymbol{x} \in A_m)p(\boldsymbol{x})d\boldsymbol{x}$*, respectively, failure rate for all failure regions is* $P^{fail} = \sum_{m=1}^{M} P_m^{fail}$*, then the mixture failure distribution of all failure regions is* $p^{ideal}(\boldsymbol{x}) = \sum_{m=1}^{M} \omega_m \cdot p_m^{ideal}(\boldsymbol{x})$ *with* $\omega_m = P_m^{fail}/P^{fail}$.

Proof. From Equation (7), the ideal distribution for mth failure region and all failure regions are, respectively,

$$p_m^{ideal}(\boldsymbol{x}) = \frac{I(\boldsymbol{x} \in A_m)\, p(\boldsymbol{x})}{P_m^{fail}}, p^{ideal}(\boldsymbol{x}) = \frac{I(\boldsymbol{x} \in A)\, p(\boldsymbol{x})}{P^{fail}}, \tag{41}$$

$$p^{ideal}(\boldsymbol{x}) = \frac{\sum_{m=1}^{M} I(\boldsymbol{x} \in A_m)\, p(\boldsymbol{x})}{P^{fail}} = \sum_{m=1}^{M} \frac{P_m^{fail}}{P^{fail}} \frac{I(\boldsymbol{x} \in A_m)\, p(\boldsymbol{x})}{P_m^{fail}} = \sum\nolimits_{m=1}^{M} \frac{P_m^{fail}}{P^{fail}} p_m^{ideal}(\boldsymbol{x}). \tag{42}$$

If there are overlaps among these failure regions, then it becomes very complicated, and we still use the same mixture distribution of Equation (14) for the approximation.

ACKNOWLEDGMENTS

We thank the Xiulong Wu of Anhui University, Hefei, China, for the experimental circuits.

REFERENCES

[1] Changdao Dong and Xin Li. 2011. Efficient SRAM failure rate prediction via Gibbs sampling. In *Proceedings of the Design Automation Conference*. 200–205.

[2] Rouwaida Kanj, Rajiv Joshi, and Sani Nassif. 2006. Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events. In *Proceedings of the Design Automation Conference*, 2006. ACM/IEEE. 69–72.

[3] Lara Dolecek, Masood Qazi, Devavrat Shah, and Anantha Chandrakasan. 2008. Breaking the simulation barrier: SRAM evaluation through norm minimization. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. 322–329.

[4] Qazi Masood, Tikekar Mehul, Dolecek Lara, Shah Devavrat, Chandrakasan and Anantha. 2010. Loop flattening & spherical sampling: Highly efficient model reduction techniques for SRAM yield analysis. In *Proceedings of the Conference on Design, Automation and Test in Europe*. 801–806.

[5] Mengshuo Wang, Changhao Yan, Xin Li, Dian Zhou, and Xuan Zeng. 2017. High-dimensional and multiple-failure-region importance sampling for SRAM yield analysis. *IEEE Trans. Very Large Scale Integr. Syst.* 25, 3 (2017), 806–819.

[6] Mohammed Abdul Shahid. 2012. Cross entropy minimization for efficient estimation of SRAM failure rate. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition*. 230–235.

[7] Fang Gong, Sina Basir-Kazeruni, Lara Dolecek, and Lei He. 2012. A fast estimation of SRAM failure rate using probability collectives. In *Proceedings of the ACM International Symposium on International Symposium on Physical Design*. 41–48.

[8] Sgibbs2hupeng Sun, Yamei Feng, Changdao Dong, and Xin Li. 2011. Efficient SRAM failure rate prediction via gibbs sampling. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 31, 12 (2011), 1831–1844.

[9] Jian Yao, Zuochang Ye, and Yan Wang. 2014. Importance boundary sampling for SRAM yield analysis with multiple failure regions. *Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 31, 12 (2011), 1831–1844.

[10] Shupeng Sun and Xin Li. 2014. Fast statistical analysis of rare circuit failure events via subset simulation in high-dimensional variation space. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. 324–331.

[11] Trent McConaghy and Patrick Drennan. 2011. Variation-aware custom IC design: Improving PVT and monte carlo analysis for design performance and parametric yield. In *Solido White Paper*.

[12] Zhenyu Wu, Changhao Yan, Xuan Zeng, and Sheng Guo Wang. 2015. Rapid estimation of the probability of SRAM failure via adaptive multi-level sliding-window statistical method. *Integr. VLSI J.* 50 (2015), 1–15.

[13] Fang Gong, Hao Yu, Yiyu Shi, Daesoo Kim, Junyan Ren, and Lei He. 2010. quickYield: An efficient global-search based parametric yield estimation with performance constraints. In *Proceedings of the Design Automation Conference*. 392–397.

[14] Chenjie Gu and Jaijeet Roychowdhury. 2008. An efficient, fully nonlinear, variability-aware non-monte-carlo yield estimation procedure with applications to SRAM cells and ring oscillators. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASPDAC'08)*. 754–761.

[15] Shweta Srivastava and Jaijeet Roychowdhury. 2007. Rapid estimation of the probability of SRAM failure due to MOS threshold variations. In *Proceedings of the IEEE Custom Integrated Circuits Conference*. 229–232.

[16] R. A. Fonseca, L. Dilillo, A. Bosio, P. Girard, S. Pravossoudovitch, A. Virazel, and N. Badereddine. 2010. A statistical simulation method for reliability analysis of SRAM core-cells. In *Proceedings of the Design Automation Conference*. 853–856.

[17] Fparametric_yieldang Gong, Yiyu Shi, Hao Yu, and Lei He. 2010. Parametric yield estimation for SRAM cells: Concepts, algorithms and challenges. In *Proceedings of the Design Automation Conference*.

[18] Amith Singhee and Rob A. Rutenbar. 2007. Statistical blockade: A novel method for very fast monte carlo simulation of rare circuit events, and its application. In *Proceedings of the Conference on Design, Automation and Test in Europe*. 1379–1384.

[19] Amith Singhee and Rob A. Rutenbar. 2009. Statistical blockade: Very fast statistical simulation and modeling of rare circuit events and its application to memory design. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 28, 8 (2009), 1176–1189.

[20] A. Papoulis and S. Pillai, 2001, *Probability, Random Variables and Stochastic Processe*. McGraw–Hill, New York, NY.

[21] Tomas M. Cover and Joy A. Tomas. 2003. *Elements of Information Theory*. Wiley. 1600–1601.

[22] A. A. Balkema and L. De Haan. 1974. Residual Life Time at Great Age. *Ann. Probab.* 2, 5 (1974), 792–804.

[23] James Pickands. 1975. Statistical Inference Using Extreme Order Statistics. *Ann. Stat.* 3, 1 (1975), 119–131.

[24] Paul T. Boggs and Jon W. Tolle. 1995. Sequential quadratic Programming. *Acta Numer.* 4, 4 (1995), 1–51
[25] Alston S Householder. 1958. Unitary Triangularization of a Nonsymmetric Matrix. *J. ACM* 5, 4 (1958), 339–342.
[26] Bo Peng, Fan Yang, Changhao Yan, and Xuan Zeng. 2016. Efficient multiple starting point optimization for automated analog circuit optimization via recycling simulation data. In *Design, Automation Test in Europe Conference Exhibition.* 1417–1422.
[27] N. L. Johnson, S. Kotz, and N. Balakrishnan. 1994. Continuous Univariate Distributions, Volume 1. John Wiley & Sons. 173.
[28] Zhang Jizhe, and S. Gupta. 2014. SRAM array yield estimation under spatially-correlated process variation. In *Proceedings of the IEEE Test Symposium.* 149–155.
[29] Winterioraltz R. A., J. L. Morales, J. Nocedal, and D. Orban. 2006. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Math. Program.* 107, 3 (2006), 391–408.

Received July 2017; accepted December 2017