Statistical Rare-Event Analysis and Parameter Guidance by Elite Learning Sample Selection

YUE ZHAO, TAEYOUNG KIM, HOSOON SHIN, and SHELDON X.-D. TAN,
University of California, Riverside
XIN LI, Carnegie Mellon University
HAIBAO CHEN, Shanghai Jiao Tong University
HAI WANG, University of Electronic Science and Technology of China

Accurately estimating the failure region of rare events for memory-cell and analog circuit blocks under process variations is a challenging task. In this article, we propose a new statistical method, called *EliteScope*, to estimate the circuit failure rates in rare-event regions and to provide conditions of parameters to achieve targeted performance. The new method is based on the iterative blockade framework to reduce the number of samples, but consists of two new techniques to improve existing methods. First, the new approach employs an elite-learning sample-selection scheme, which can consider the effectiveness of samples and well coverage for the parameter space. As a result, it can reduce additional simulation costs by pruning less effective samples while keeping the accuracy of failure estimation. Second, the *EliteScope* identifies the failure regions in terms of parameter spaces to provide a good design guidance to accomplish the performance target. It applies variance-based feature selection to find the dominant parameters and then determine the in-spec boundaries of those parameters. We demonstrate the advantage of our proposed method using several memory and analog circuits with different numbers of process parameters. Experiments on four circuit examples show that *EliteScope* achieves a significant improvement on failure-region estimation in terms of accuracy and simulation cost over traditional approaches. The 16b 6T-SRAM column example also demonstrates that the new method is scalable for handling large problems with large numbers of process variables.

CCS Concepts: • **Computing methodologies** → **Rare-event simulation**; • **Theory of computation** → *Approximation algorithms analysis;*

Additional Key Words and Phrases: Process variations, rare event analysis, statistical analysis, Monte Carlo method

This work is supported in part by NSF grant No. CCF-1116882 and in part by NSF grant No. OISE-1130402. Haibao Chen and Hai Wang are supported by 985 research funds from Shanghai Jiao Tong University and University of Electronic Science and Technology of China.

Authors' addresses: Y. Zhao and S. X.-D. Tan (corresponding author), Department of Electrical Engineering, University of California, Riverside, Suite 343 Winston Chung Hall, University of California, Riverside, Riverside, CA 92521-0429; emails: {yzhao011, stan}@ee.ucr.edu; T. Kim, Department of Computer Science, University of California, Riverside, 351 Winston Chung Hall, University of California, Riverside, Riverside, CA 92521-0429; X. Li, Department of Electrical and Computer Engineering, Carnegie Mellon University, 2123 Hamerschlag Hall, 5000 Forbes Avenue, Pittsburgh, PA 15213; email: xinli@cmu.edu; H. Chen, Department of Micro/nano-electronics, Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai 200240, China; email: haibaochen@sjtu.edu.cn; H. Wang, School of Microelectronics and Solid-State Electronics, University of Electronic Science and Technology of China, No.4, Sec.2, North Jianshe Rd., Chengdu 610054, China; email: wanghai@uestc.edu.cn.
Author's current address: H. Shin, Qualcomm, 5775 Morehouse Dr, San Diego, CA 92121, USA; email: dashin@qti.qualcomm.com.

DOI: http://dx.doi.org/10.1145/2875422

ACM Reference Format:
Yue Zhao, Taeyoung Kim, Hosoon Shin, Sheldon X.-D. Tan, Xin Li, Haibao Chen, and Hai Wang. 2016. Statistical rare-event analysis and parameter guidance by elite learning sample selection. ACM Trans. Des. Autom. Electron. Syst. 21, 4, Article 56 (May 2016), 21 pages.
DOI: http://dx.doi.org/10.1145/2875422

1. INTRODUCTION

As the CMOS technology scaling continues, performance uncertainties related to process variation have become a major concern for IC development [Calhoun et al. 2008]. Many IC components such as SRAM bit-cells need to be tremendously robust as they are duplicated in the millions [Masuda et al. 2005]. Such modules require accurate statistical-failure analysis in rare-event regions. However, the traditional Monte Carlo (MC)–based statistical analysis method faces a challenge, as it may require millions of simulations [Singhee and Rutenbar 2009].

To mitigate this problem, a number of statistical analysis algorithms have been developed in the literature [Hocevar et al. 1983; Singhee et al. 2008; Wu et al. 2014a, 2014b; Singhee and Rutenbar 2009; Mukhopadhyay et al. 2004; Agarwal and Nassif 2006; Qazi et al. 2010; Nassif et al. 2006; Gong et al. 2012; Shen et al. 2012a, 2012b]. One existing work for yield analysis using a spectral stochastic method has been proposed [Gong et al. 2012; Shen et al. 2012a, 2012b]. The spectral stochastic method applies stochastic orthogonal polynomials to represent the variational performance results. The analysis process boils down to determination of the coefficients of those polynomials, which is more efficient than the MC methods if the number of variables is not large. It become less efficient, however, for high-dimensional problems (large number of random variables).

Another approach is by means of importance sampling (IS) [Hocevar et al. 1983], which consist of two steps. First, it shifts the mean value of the initial performance distribution and places it in an interested failure region. The standard deviation based on the shifted mean is recalculated by considering samples placed only in the failure region. The new probability density function (PDF) is generated based on an updated mean and standard deviation so that more samples in the failure region can be drawn. Nassif et al. [2006] applied the mixture of IS for cross-validation of multiple failure regions due to disjointed process parameters. However, these approaches can estimate only a single performance metric. Multiple importance samplings are required to estimate more metrics. Also, it is difficult to calculate the failure probability for a generated distribution by IS.

The statistical blockade (SB) is another effective approach for improving the performance of the MC method [Singhee and Rutenbar 2009]. The idea of this approach is using a threshold bound to separate an interested failure region from the whole distribution so that it can *block* some unnecessary sampling and simulation for efficiency improvement. This method builds a supervised learning model with the threshold bound and initial simulation data, which is known as a "classifier," to recognize failure samples. Later samples that tend to be placed in the failure region can be captured without simulation. This approach was improved by using the recursive statistical blockade (RSB) scheme to locate the rare-event failure region in an iterative way [Singhee et al. 2008]. This method can improve the accuracy of the classifier iteratively by increasing the number of samples in the failure region of interest. However, this method can incur significant extra cost as it needs more samples for the simulation. Wu et al. [2014b] applied a nonlinear Support Vector Machine (SVM) classifier to model nonlinear and multiple disjoint failure regions of circuits. This method applies the generalized Pareto distribution (GPD) fitting for tail distribution to model failure probability in each iteration. However, this method cannot further investigate the failure region without

rerunning the whole algorithm. The reason is that the pruned parameters, depending on the initial samples, cannot remain as important in the failure region, which keeps changing. Also, the previous approaches cannot provide the design guideline in terms of the design parameters to explicitly avoid performance failure, which is important for improving the yield of the circuit.

Recently, Sun et al. [2013] proposed scaled-sigma sampling (SSS) and the subset simulation technique (SUS) [Sun and Li 2014] as better solutions to estimate rare failure rates with high-dimensional variation space. Unlike traditional IS, SSS applied a soft maximum theorem to construct an analytical model, which is insensitive to dimensionality, for rare failure rate estimation. SUS, on the other hand, tends to express rare failure probability as a product of conditional probability of intermediate failure events, which is similar to the recursive statistical blockade concept. These intermediate terms are then accurately estimated using the Markov Chain Monte Carlo (MCMC) algorithm.

In this article, we propose a new statistical failure region diagnosis method. The new method, called *EliteScope*, is based on a recursive statistical blockade method to reduce the sample counts while still maintaining estimation accuracy, but consists of two new techniques to improve existing methods. The main contributions of this article are as follows:

- First, in the recursive statistical blockade method, more samples will be generated when the failure region is redefined gradually. To mitigate these problems, the new approach applies an elite sampling scheme, which considers both effectiveness of samples and well coverage for process parameter search space, to reduce the number of generated samples after the failure regions are relocated.
- Second, our approach provides safe boundaries or in-spec boundaries of process parameters to satisfy the design specification and manage yield variability of the circuit. The new method first applies variance-based feature selection to find the dominant parameters. A quasirandom sampling with dominant parameters is then used to quickly determine proper boundaries of those parameters.

The presented method has been tested on several types of digital and analog/mixed-signal circuits: a 4-gate logic circuit with 48 process parameters; a charge pump operation failure diagnosis in a PLL circuit with 81 process parameters; a 6T-SRAM reading failure diagnosis with 27 process parameters; and a 16b 6T-SRAM column reading failure diagnosis with 432 process parameters. Experimental results show that, given the same computing costs, the proposed method in general can be more accurate than all existing methods. For instance, for the 16b 6T-SRAM column, with a similar number of samples used, *EliteScope* can deliver about $3X$ to $10X$ accuracy improvement over existing methods. Furthermore, the 16b 6T-SRAM column example also shows that the new method can easily handle the statistical analysis problems with large numbers of process variables.

The rest of this article is organized as follows. Section 2 discusses essential background for sample-based failure analysis and revisits some important techniques for improving MC performance. In Section 3, we first introduce the overall flow of the proposed method and review of the mathematical framework for the recursive statistical blockade–based method. We then present the proposed elite sampling method to reduce the number samples for efficiency improvement. We then introduce a new guidance technique of parameters to meet target performance. Section 4 contains the experimental result for verifying the accuracy and efficiency of the proposed method. Computation complexity and convergence performance analysis of *EliteScope* is also discussed. Section 5 presents our conclusions.

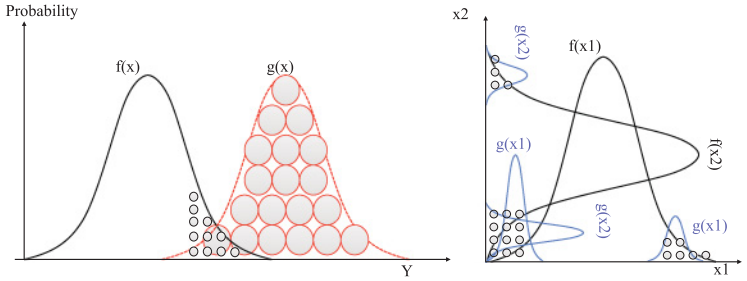


Fig. 1. Generated PDF by importance sampling in 2D random variables.

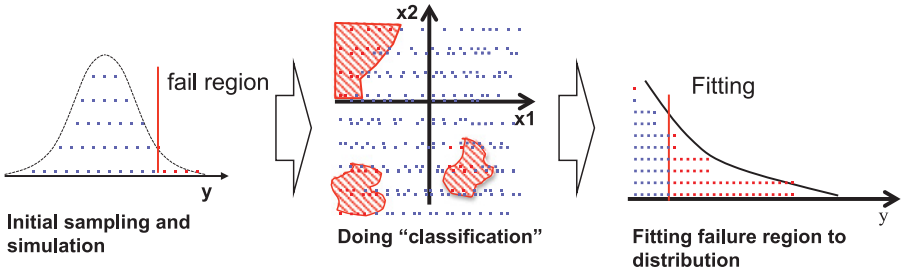


Fig. 2. General flow of statistical blockade approach.

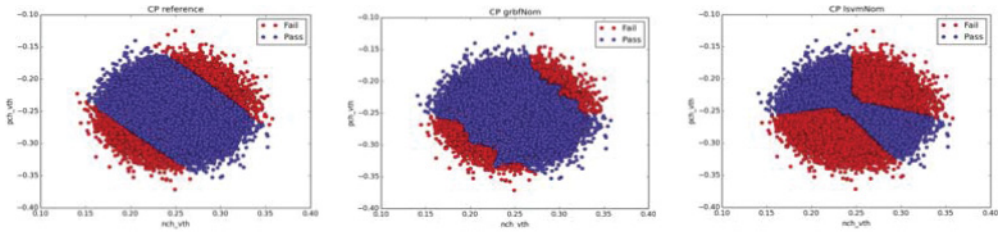
2. BACKGROUND

2.1. Importance Sampling

In sample-based statistical analysis, IS is a general technique to estimate properties of rare-event regions using the samples generated from the initial distribution. Figure 1 shows the generated distribution $g(x)$ by IS with two parameters. As we can see in the figure, the property of the failure region can be captured as more samples are obtained in the failure region. Therefore, the proper sampling scheme is needed to build the right distribution representing the rare-event region. One of the IS-based approaches is focusing on quasirandom sampling to explore the parameter space more uniformly. The samples can be selected by the initial MC sampling so that more regular space filling makes the initial samples to cover large variety combinations of parameters [Montgomery 2013]. Another well-known approach is mean shifting and variance reconstructing, for which the initial distribution is centered around the failure region [Nassif et al. 2006; Qazi et al. 2010]. However, all these approaches assume a linear relation between the reconstructed and the initial distributions; thus, the generated samples cannot reflect the nonlinear rare-event region correctly.

2.2. Statistical Blockade

We first briefly review the concept of the SB approach for fast estimation of properties of the rare-event region. A general framework of SB is shown in Figure 2. This method starts with drawing initial samples with uniform or normal distribution to capture a crude shape of performance distribution by circuit simulation. The classifier can be built by training with initial simulation data. Once we obtain the classifier, samples that tend to fall into the failure criteria can be identified without actual simulation. With these filtered samples, SB calculates the probability of failure region by fitting samples in a proper distribution model. Thus, "classification" and "failure probability calculation" are both key steps in the SB method. The rest of this section describes these two steps.



(a) Reference (b) GRBF (c) LSVM

Fig. 3. The classification accuracy of two different methods.

2.2.1. Classification. Classification is a step in which the samples can be classified into likely-to-fail samples for circuit simulation. Building a classifier needs a training step with initial samples to render real shapes of the failure region. The classifier can shrink the number of samples, thus the simulation cost is reduced. However, it is not capable of fully replacing the simulator due to its accuracy. Thus, a marginal filtering approach is used to improve the accuracy of classification [Singhee et al. 2008; Wu et al. 2014a, 2014b; Singhee and Rutenbar 2009]. This method uses relaxed threshold bounds instead of a real failure criterion to capture more samples to minimize classification error. Meanwhile, it is not sufficient to use a simple and linear classifier due to the nonlinearity of the failure region [Singhee et al. 2008; Wu et al. 2014a; Singhee and Rutenbar 2009]. Thus, Gaussian radial basis function kernel (GRBF) and neural network methods are available for nonlinearclassifiers [Wu et al. 2014b; Hastie 2009]. Figure 3 shows the accuracy of classification in a 2-dimensional search space example. Even when the solution space is separated a with nonlinear relation, GRBF can recognize patterns properly while the LSVM draws a wrong boundary between two categories.

2.2.2. Failure Probability Calculation. The failure samples should be fitted to a particular distribution form in order to calculate the probability of the failure region. Suppose that simulation results for a certain performance metric Y can be fitted to the Gaussian distribution. The PDF of the result distribution can be represented as

$$f(y, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}, \tag{1}$$

where parameters μ and σ are the mean and standard deviation in this distribution. We define $F_Y(y)$ as the cumulative density function (CDF) of the performance metric Y. If we know the threshold value t that separates a tail region from the whole distribution $f(y)$, the conditional CDF of this region can be written as follows:

$$F_t(y) = P(Y \geq y \mid Y \geq t) = \frac{F_Y(y) - F_Y(t)}{1 - F_Y(t)}, \tag{2}$$

where $F_t(y)$ means the failure probability decided by y. Once we have a suitable fitting model for CDF of the failure region with a failure bound y, the failure probability with given values can be calculated as

$$P(Y \geq y) = [1 - F_Y(t)] \cdot [1 - F_t(y)]. \tag{3}$$

In the several generalized extreme value distributions, GPD is one of the most accurate models to describe tail distribution corresponding to failure region [Wallis and Hosking 1987]. With the location parameter μ, the scale parameter σ and the shape parameter

ξ, CDF of the failure region can be formulated by GPD fitting:

$$F_t(y) = G_{(\xi,\mu,\sigma)}(y) = \begin{cases} 1 - \left(1 + \dfrac{\xi(y-\mu)}{\sigma}\right)^{-1/\xi} & \text{for } \xi \neq 0. \\ 1 - e^{\frac{-(y-\mu)}{\sigma}} & \text{for } \xi = 0 \end{cases} \quad (4)$$

The location parameter μ means a starting point of GPD and corresponds to the threshold t of the tail distribution. Consequently, the failure probability with a given threshold t and failure bound y can be computed as follows:

$$P(Y \geq y) = [1 - F_Y(t)] \cdot [1 - G_{(\xi,t,\sigma)}(y)]. \quad (5)$$

To approximate the rest of the parameters for GPD fitting, we use the maximum likelihood estimation [Hosking 1985].

3. PROPOSED NEW FAILURE REGION DIAGNOSIS METHOD

3.1. The Overall Analysis Flow and Iterative Computing of Failure Probability

In this section, we first present the overall analysis flow of the proposed *EliteScope* method, which is illustrated in Figure 4. Nezxt, we present the mathematical framework for the iterative computing of failure probability. Then, we explain our three major contributions of the proposed method: (1) iterative computing of failure probability, (2) elite learning sample selection, and (3) parameter guidance for performance targeting.

Our algorithm starts with given data, such as process variations and some parameters for failure-region determination. The failure criteria t_c denotes the reference value of failure and the percentile bound p to calculate the threshold in each i_{th} iteration. The first step is to perform initial MC sampling and simulation to capture overall circuit performance metrics. After this, the relaxed threshold t_i can be obtained to separate a failure region from the main PDF; the probability of this region is $P(Y > t_i) = p$. The classifier can then be modeled with n simulation result of the initial samples. In the classification step, the GRBF nonlinear classifier is used for accurate sample filtering. With the simulation result and the classifier, the new method can calculate the in-spec conditions of process parameters to achieve a targeted yield in the i_{th} iteration. At the same time, the algorithm generates $n * m^i$ (m is a constant number) MC samples, which will be filtered by the classifier C_i to likely-to-fail samples based on t_i. Then, elite sample selection can be employed to further reduce the number of samples for actual simulation. After the simulation, the failure probabilities $P(Y > t_i)$ are updated by GPD fitting. Our approach iterates the whole procedure with the updated threshold bound t_i by percentile bound p and the increased number of MC samples to calculate the failure probability $P(Y > t_i)$. It finishes when the threshold bound meets the given failure criterion t_c.

It is a typically difficult task to choose the right threshold bound in the failure analysis for an extreme rare-event region. For instance, the failure region is decided by the failure criterion t_c and the probability of this region is around 99.9999%. Suppose that we use the single threshold method; then, we can choose a very loose threshold t as $P(Y > t)$ around 99% to safely cover the whole failure region even though the threshold can be quite far away from t_c. Moreover, the number of MC samples for filtering will be determined at once. If the number of MC samples is relatively enormous, a classifier will select too many likely-to-fail samples, which will significantly increase simulation cost. Meanwhile, if the number of selected likely-to-fail samples is every small,

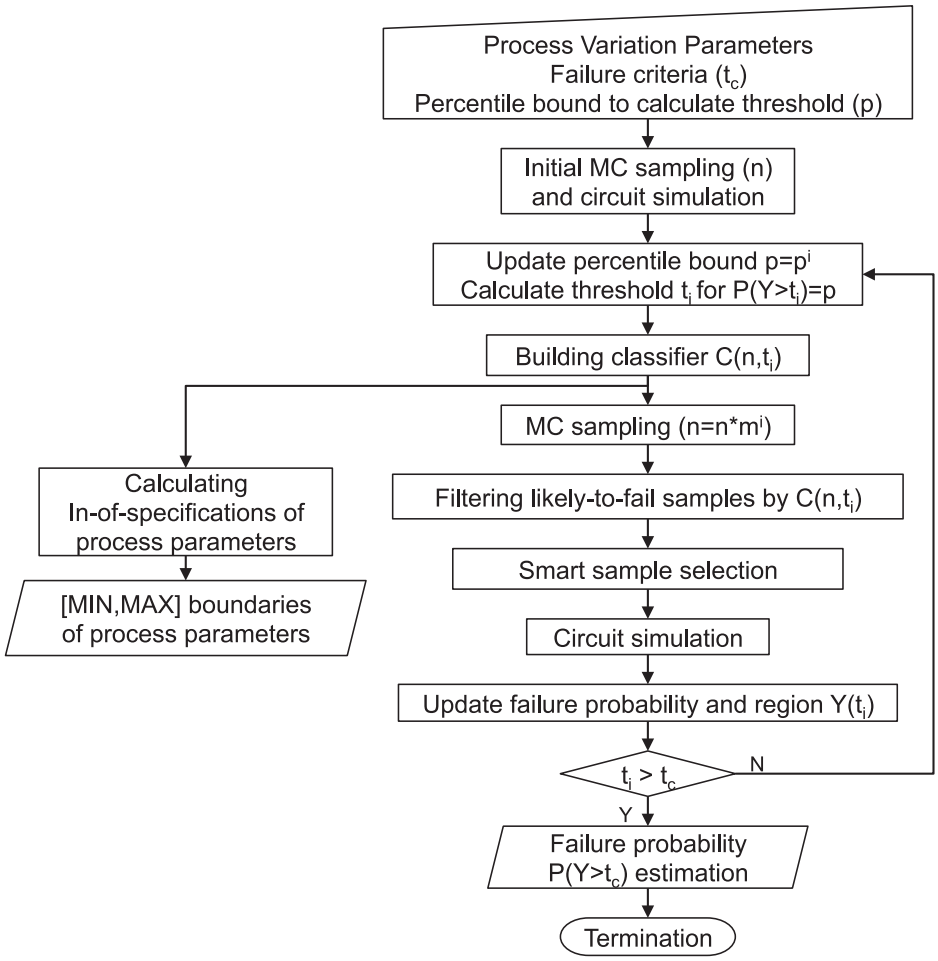


Fig. 4. The proposed iterative failure region diagnosis flow.

this will cause an inaccurate estimation of the failure region, and thus the failure probability.

To mitigate this problem, one idea is to gradually locate the failure region in an iterative way based on the RSB scheme [Singhee et al. 2008]. Unlike the single-threshold method, which calculates a failure probability at once, our approach updates a failure region $Y_1(>99\%)$ and the probability by GPD fitting after the first iteration. With this updated failure region, the threshold bound is recomputed for a newly updated region $Y_2(>99.99\%)$. The GRBF classifier is trained by failure samples in the first iteration so that it can capture likely-to-fail samples more precisely in the updated failure region. In the second iteration, the number of MC samples increases from $10n$ to 10^2n as the increasing ratio is 10; thus, the new classifier will capture more likely-to-fail samples, better estimating the updated failure region.

As the algorithm iterates, the failure region is scoped continuously close to the given failure criterion t_c based on the recomputed t_i, and likely-to-fail samples are converged on the updated failure region. Therefore, the proposed method can achieve more accurate failure analysis than a single-threshold method with relatively less total

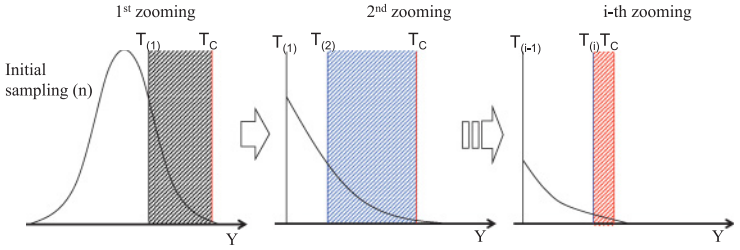


Fig. 5. Iterative locating of failure region by changing thresholds.

simulation cost. Figure 5 shows an iterative locating procedure for finding the failure region.

Mathematically, as discussed in Section 2, the CDF with given a threshold t and a failure criterion t_c can be calculated as

$$\begin{aligned} P_{IS}(Y \geq t_c) &= P_{MC}(Y \geq t) \cdot P(Y \geq t_c | Y \geq t) \\ P(Y \geq t_c | t \geq t) &= \frac{P(Y \geq t_c, Y \geq t)}{P(Y \geq t)} \quad . \end{aligned} \tag{6}$$

The conditional probability part in Equation (6) can be estimated by GPD fitting using simulated failure samples. Therefore, Equation (6) can be rewritten as

$$P_{IS}(Y > t_c) = P_{MC}(Y \geq t) \cdot P_{MIS}(Y \geq t_c | Y \geq t), \tag{7}$$

where P_{MIS} represents the conditional probability in updated distribution by GPD fitting. If the proposed method iterates twice with t_1 and t_2 as threshold bounds, the second failure probability can be calculated based on the first failure region. Thus, the failure probability in each step can be calculated as

$$\begin{aligned} P_{IS}(Y \geq t_c) &= P_{IS(2)}(Y \geq t_c) \\ P_{IS(1)}(Y \geq t_2) &= P_{MC}(Y \geq t_1) \\ &\quad \cdot P_{MIS(1)}(Y \geq t_2 | Y \geq t_1) \\ P_{IS(2)}(Y \geq t_c) &= P_{MC}(Y \geq t_1) \cdot P_{MIS(1)}(Y \geq t_2) \\ &\quad \cdot P_{MIS(2)}(Y \geq t_c | Y \geq t_2) \\ P_{MIS(i)}(Y \geq t_c | Y \geq t_i) &= \frac{P_{MIS(i)}(Y \geq t_c)}{P_{MIS(i)}(Y \geq t_i)} \end{aligned} \tag{8}$$

Without loss of generality, we can formulate the iterative failure probability calculation as

$$\begin{aligned} &P_{IS(i)}(Y \geq t_c) \\ &= \begin{cases} P_{MC}(Y \geq t_i) \cdot P_{MIS(i)}(Y \geq t_c | Y \geq t_i) & \text{for } i = 1 \\ P_{MC}(Y \geq t_i) \cdot \prod_{i=1}^{k-1} (P_{MIS(i)}(Y \geq t_{i+1}) & \\ \quad \cdot P_{MIS(k)}(Y \geq t_c | Y \geq t_k)) & \text{for } i > 1, \end{cases} \end{aligned} \tag{9}$$

where k is the number of iterations. Finally, the failure probability can be obtained by combining all calculated probabilities in each iteration.

3.2. Elite Learning Sample Selection

Simulation cost is a major bottleneck in the statistical analysis of the circuit. The proposed iterative failure diagnosis method can lead to an extra simulation cost in

each iteration. To mitigate this problem, we propose the elite learning sample selection scheme, which significantly reduces the number of samples required. The elite sample selection process is represented in the box named "Smart sample selection" in Figure 4. Effectiveness of the sample group is the first factor. Each sample consists of the combination of process parameters, which affect simulation results differently. Therefore, the sensitivity of each parameter should be considered for the sample selection.

Suppose that we have a set of n samples, which are represented by the parameter vectors $x_i, i = 1, \ldots, n$. Each sample has m process variables (m dimensions). Together, they form a process parameter matrix $X = [x_1 \quad x_2 \quad \ldots \quad x_n]$ such that each column indicates an x_i. It is not difficult to see that each row of X is the n samples of a single parameter. Denote X^j as the vector formed from the jth row of X (jth process variable). A scalar vector $y = [y_1, y_2, \ldots, y_n]^T$ contains all the corresponding n simulation results. σ_{X^j} and σ_Y are variances of X^j and y, respectively. The proposed selection method calculates correlation coefficients between parameters and simulation results for the sensitivity analysis as follows:

$$\begin{aligned} & X^j, y \in R^n \\ & \rho_{X^j,y} = \frac{cov(X^j, y)}{\sigma_{X^j}\sigma_y} \qquad j = 1, 2, \ldots, m\,, \\ & \rho_{X,y} = [\rho_{X^1,y} \quad \rho_{X^2,y} \quad \ldots \quad \rho_{X^m,y}]^T \in R^m \end{aligned} \tag{10}$$

where $\rho_{X,y}$ means the covariance coefficient of simulation results and m process parameters.

The second factor is the coverage ratio of parameter search space by selected samples. The diversity of samples can be calculated by Euclidean distances with the reference sample, which is the median from simulation results. Samples around the median can be chosen as the median is located on the highest probability region in the distribution of simulation results. Simultaneously, samples found in the boundary region of the search space can be selected, as these samples represent the maximum and minimum conditions of parameters. Thus, the proposed sampling method can calculate two distance factors of a given sample that cover both central and boundary regions of the search space as follows:

$$\begin{aligned} & \tilde{y} = median(y) \\ & x_{ref} = \{x | \tilde{y} = f(x), x \in R^m\} \\ & D_{central}(x) = \frac{1}{\left|\frac{x - x_{ref}}{Range(X)}\right|} \in R^m \\ & D_{boundary}(x) = \left|\frac{x - x_{ref}}{Range(X)}\right| \in R^m \end{aligned} \quad , \tag{11}$$

where $Range(X)$ is a normalization term such that the jth element of vector $x - x_{ref}$ is normalized by $|max(X^j) - min(X^j)|$, the value range of jth row in X. In Equation (11), $D_{central}$ increases when the sample is closer to the reference sample. On the other hand, $D_{boundary}$ increases. Since $D(x)$ and $\rho_{X,y}$ mean the distance and correlation coefficient in the same dimensions, we can obtain the sample's weight by the inner product of $\rho_{X,y}$ and $D(x)$ of each sample:

$$\begin{aligned} & W_{central}(x) = {\rho_{X,y}}^T \cdot D_{central}(x) \\ & W_{boundary}(x) = {\rho_{X,y}}^T \cdot D_{boundary}(x) \end{aligned}. \tag{12}$$

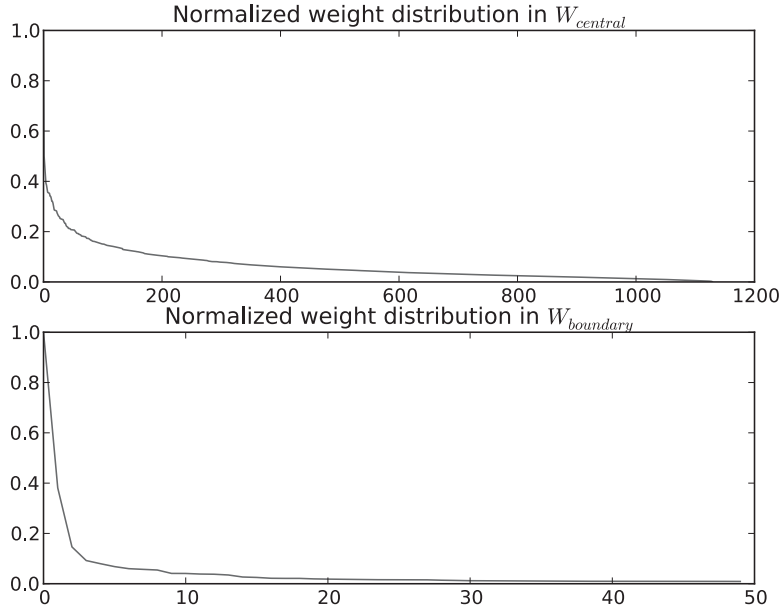


Fig. 6. Sample candidates weight distribution of a single-bit SRAM test.

According to the selection ratio r, which determines the number of selected samples, the final set of samples can be chosen in the following way:

$$E(n, r) = S\left(\frac{nr}{2}, W_{central}(x_n)\right) \cup S\left(\frac{nr}{2}, W_{boundary}(x_n)\right), \tag{13}$$

where $S(n, W(x_n))$ is the set of n samples sorted by $W(x_n)$.

An example of normalized sample candidates weight distribution is shown in Figure 6. Nearly 1200 sample candidates are filtered out by the nonlinear SVM classifier. By employing the Elite Learning Sample Selection scheme, the weight concerning both central and boundary distance are calculated sorted in descending order. All weights in the same set are then normalized by sets maximum. For the normalized weight distribution in the set $W_{boundary}$, only the first 50 samples are listed since rest samples' weights are very close to zero, and thus negligible. It is very clear to see that weight values decease dramatically. According to previous discussion, a sample with larger weight leads to a more significant contribution in constructing the failure region. Unlike the traditional RSB method, which directly simulates all sample candidates, we need to utilize only samples with larger weight to perform actual simulation to estimate failure regions with great efficiency.

3.3. Parameters Guidance for Performance Targeting

In order to improve the yield of a circuit, designers need to know good ranges of process parameters with regard to the circuit performance specification. However, applying all possible combinations of parameters is impossible due to exponential possibilities with a large number of parameters.

The proposed method ranks priorities of process parameters based on its variances. The parameter guidance operation is represented by the two left boxes in Figure 4. Since the parameters with huge variance mainly lead to spread samples in search spaces, these parameters must be handled to avoid certain failure regions. In Figure 4,

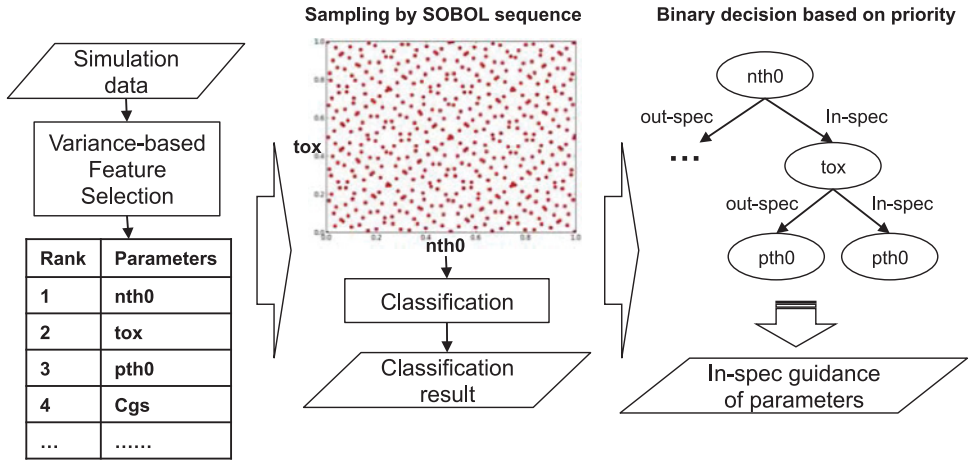


Fig. 7. Overall flow of parameter guidance for performance targeting: (1) feature selection, (2) sampling and classification, and (3) calculating the boundaries for in-spec conditions.

the variances of parameters can be calculated from simulation data of the updated failure region in our iterative framework. Given n samples with m process parameters, as denoted by $X = [x_1 \quad x_2 \quad ... \quad x_n]$, the variance of samples can be written mathematically as follows:

$$Var(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2, \tag{14}$$

where x_i is an ith X and μ is the mean vector of n samples of X. Next, our method redraws samples considering only the distributions of high-ranked parameters. Nominal values are assigned for not chosen parameters. We use the SOBOL sequence [Joe and Kuo 2003] to redraw these samples. It uses a quasirandom low-discrepancy sequence; thus, these samples can cover the search spaces of parameters more uniformly than the previous samples for simulation. Suppose that l is the number of redrawn samples and the first k high-ranked parameters are chosen. Redrawn samples can be formed as

$$\begin{aligned} &x_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,k}, x_{i,(k+1)}, \ldots x_{i,m}]^T \in R^m \\ &[x_{i,1}, x_{i,2}, \ldots, x_{i,k}] = SOBOL[x_{1,\ldots,l}, MIN(x_{1,\ldots,l}), MAX(x_{1,\ldots,l})], \\ &[x_{i,(k+1)}, x_{i,(k+2)}, \ldots x_{i,m}] = NOMINAL(x_{1,\ldots l}) \end{aligned} \tag{15}$$

where $i = 1, 2, \ldots, l$, $x_{i,k}$ denotes the kth element of x_i, terms $MIN(x_{1,\ldots,l})$ and $MAX(x_{1,\ldots,l})$ mean the minimum and maximum values of l vectors in X, respectively. We assign nominal values for the rest of the $m-k$ parameters of the redrawn samples. As a result, we can generate samples with not only reduced dimensions, but also well coverage of the failure region. The classifier with the updated threshold can filter out these samples to determine the pass or fail condition of process parameters. With the classification result of samples, the proposed method induces *if-then* rules from the highest-ranked parameters so that all failure conditions of parameters can be filtered. The overall steps of the new in-spec guidance method are explained in Figure 7.

4. NUMERICAL RESULTS AND DISCUSSIONS

The proposed method (*EliteScope*) has been implemented in Python 2 and tested on a Linux workstation with 32 CPUs (2.6GHz Xeon processors) and 64GB RAM.

Table I. Process Parameters of MOSFET

Variable name	Std(σ)	Unit
Flat-band voltage (V_{fb})	0.1	V
Gate oxide thickness (t_{ox})	0.05	m
Mobility (μ_0)	0.1	m^2/Vs
Doping concentration at depletion N_{dep})	0.1	cm^{-3}
Channel-length offset (ΔL)	0.05	m
Channel-width offset (ΔQ)	0.05	m
Source/drain sheet resistance (R_{sh})	0.1	Ohm/mm^2
Source-gate overlap unit capacitance (C_{gso})	0.1	F/m
Drain-gate overlap unit capacitance (C_{gdo})	0.1	F/m

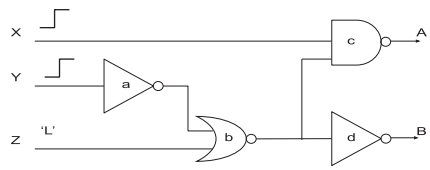


Fig. 8. The schematic of the 4-gates logic circuit.

Table II. Comparison of the Accuracy and Efficiency on the 4-Gates Circuit

	Failure probability	# Sim. runs	Speed-up(x)	Error (%)
Monte Carlo	1.25E-04	600K	–	–
Rare Event Microscope (REscope)	6.00E-04	4531	132.4	380.16
Recursive Statistical Blockade (RSB)	1.49E-04	12369	48.5	19.2
Proposed method (EliteScope)	1.51E-04	5620	106.8	20.8

The performance and accuracy of the proposed method have been evaluated on a number of circuits: (1) the critical path delay of the 4-gate logic circuit; (2) the failure rate of a 6T-SRAM single-bit cell; (3) the failure rate of a 6T-SRAM 16b column; and (4) a charge pump circuit in PLL, which are highly replicated instances for system-on-chip (SOC) designs. All circuits were designed with the BSIM4 transistor model and simulated in NGSPICE [Nenzi and Holger 2010]. Table I shows 9 major process parameters of MOSFETs. To demonstrate the advantage of the proposed method, we compare the proposed *EliteScope* against three other methods, Monte Carlo (MC), REscope [Wu et al. 2014b], and the RSB method [Singhee et al. 2008] in terms of their accuracies and performances. The three other methods are also implemented in Python 2 and tested on the same workstation. In the last part of the section, we will discuss algorithm and classifier complexity as well as the issue of the convergence performance of *EliteScope*.

4.1. The Critical Path Delay of the Simple Logic Circuit

The test logic circuit consists of four gates (2 INVs, 1 NOR, and 1 NAND), as shown in Figure 8. The critical path delay in the circuit is $max(fall\_A, fall\_B)$ (X,Y is rising and Z is 0). Two critical paths can be found, and the total number of process parameters is 48. The failure criterion is set to be $P(Y < t_c) = 0.999875$, which indicates the 4-sigma range in the distribution of the critical path delay. Two iteration threshold bounds are $P(Y < t_1) = 0.93$ and $P(Y < t_2) = 0.9951$, respectively.

As we can see from Table II, both *EliteScope* and RSB have similar accuracies for failure-region estimation, but RSB takes $2.20X$ more simulation time. By applying elite learning sample selection, *EliteScope* uses only a small amount of samples, which are

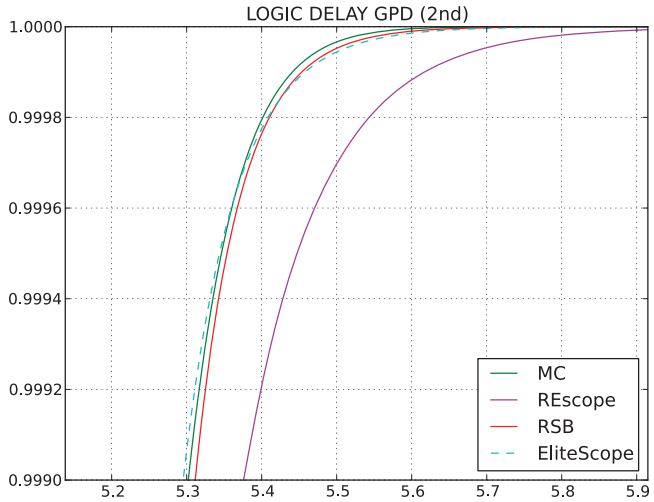


Fig. 9. The failure distribution $P(Y < t_c) = 0.999875$ of the critical path delay of the simple circuit.

filtered by classifier, for simulation and further tail distribution fitting; RSB directly simulates all of them.

In this case, REscope does not deliver very accurate estimation. There is about $19X$ accuracy difference between REscope and *EliteScope* even though their simulation cost difference is only about 20%[1].

4.2. Failure Rate Diagnosis of Single-Bit 6T-SRAM Cell

The second example is a single-bit 6T-SRAM circuit. The schematic design of the single-bit 6T-SRAM cell using the BSIM4v4.7 MOSFET Model is shown in Figure 10. 6T-SRAM fails when the voltage gap between BL and $\overline{BL}$ is not large enough to be determined by sense amplifiers in a certain period. We measure the delay of discharging $\overline{BL}$ as the failure criterion. Our experimental setup for the initial conditions are $\bar{Q} = 1$, $Q = 0$, BL and $\overline{BL} = 0$. When WL turns on, $\overline{BL}$ is discharged by MN2 and MN1, and BL is charged by MP2. For process variables, we use the 9 model parameters in Table I. To guarantee unbiased behavior, transistors on the left-hand side should be totally identical to their corresponding transistors on the right (e.g., Mp1 and Mp2 share identical process parameters). Thus, the number of the process parameters is $27(3 * 9)$ in this reading operation. The initial number of samples for capturing the circuit behavior is 2,000.

We set the failure criterion t_c as $P(Y \geq t_c) = 0.00023$, which means 3-sigma in terms of the yield level. The proposed method iterates twice with 97% percentile bound for each iteration to separate the failure region from the initial distribution. Hence, threshold bounds t_1 and t_2 are calculated as $P(Y \geq t_1) = 0.03$ and $P(Y \geq t_2) = 0.0009$ ($0.03 \times 3\%$), respectively. Table III shows the accuracy and performance of failure analysis performed by different approaches.

As we can see, compared to REscope, EliteScope obtains better accuracy with similar computing costs compared to the RSB method, which obtains better accuracy but needs almost $2X$ computing time. Figure 11 shows that our proposed method is more accurate than previous methods since the tail of CDF depicting the 3-sigma failure

[1]Note that it is difficult to make the simulation samples exactly the same for both methods, as we do not control them directly.

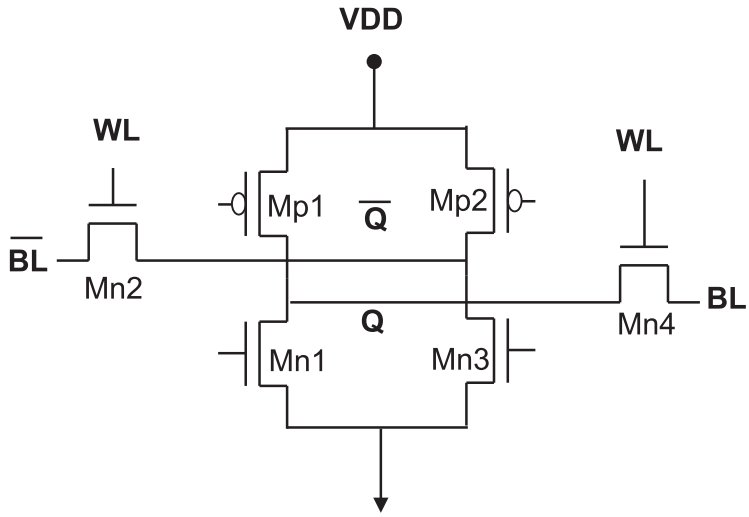


Fig. 10. The schematic of the 6T-SRAM single-bit cell.

Table III. Comparison of the Accuracy and Efficiency on the 6T-SRAM Circuit

	Failure probability	**# Sim. runs**	**Speed-up(x)**	**Error (%)**
Monte Carlo (MC)	2.300E-04	1 million	–	–
Rare Event Microscope (REscope)	3.79E-04	5009	199.6	64.78
Recursive Statistical Blockade (RSB)	2.78E-04	29260	34.2	20.86
Proposed method (EliteScope)	2.85E-04	15730	63.6	24.00

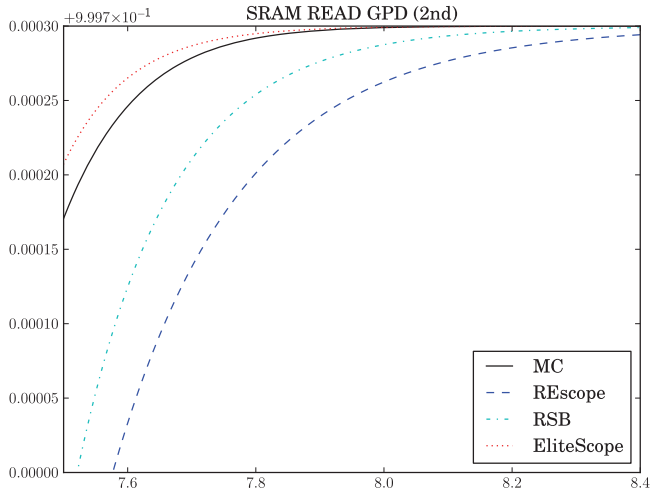


Fig. 11. Estimating the CDF of the 6T-SRAM read time around the 3-sigma region.

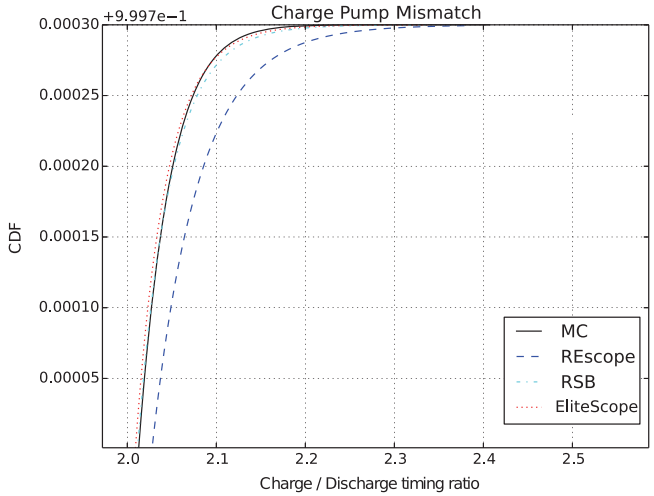


Fig. 12. Estimating the CDF of a charge pump mismatch around the 3-sigma region.

Table IV. Estimated In-Spec Guidance of Parameters on the 6T-SRAM Circuit

Rank	**Parameter @MOSFET**	**Initial condition (μ, σ)**	**In-spec Guidance [MIN,MAX]**
1	vfb@MN1	(-5.5E-01,0.1)	[-7.31E-01,-3.78E-01]
2	vfb@MP2	(5.5E-01,0.1)	[3.54E-01,7.15E-01]
3	$ndep$@MN1	(2.8E+18,0.1)	[1.84E+18,3.76E+18]
4	$ndep$@MN2	(2.8E+18,0.1)	[1.70E+18,3.84E+18]
5	$ndep$@MP2	(2.8E+18,0.1)	[1.89E+18,3.78e+18]
Failure probability	$0.0009(= t_2)$	**Estimation Error (%)**	**1.21**

region is more correlated to the gold standard reference (MC). For the estimated specification guidance for parameters, we find that only 1.2% of samples, which meet the in-spec guidance, are determined as the misclassification samples by the classifier in Table IV.

4.3. Charge Pump Failure Rate Diagnosis

The third example is a charge pump circuit. In a large logic circuit, a clock is frequently distributed to several subclocks; thus, frequencies of subclocks are prone to be inaccurate due to propagation delays. A PLL is frequently used to adjust the phase of the clock. The functional block diagram of PLL is shown in Figure 13. After comparing the output clock (CLK_{out}) with the reference clock (CLK_{ref}) by phase detector, a charge pump circuit adjusts the frequency of a clock signal by charging and discharging capacitors controlled by input signals (UP and DN). The mismatch of MOSFETs in a charge pump can cause unbalanced timing and phase jitter between two different operation modes. Hence, we measure the timing ratio of charging and discharging operations, which can be formulated mathematically as $r_{min} \leq \frac{t_{discharge}}{t_{charge}} \leq r_{max}$ ($r_{min,max}$ represents the minimum and maximum ratio to determine failures). A charge pump circuit consists of 9 MOSFETs, as shown in Figure 14. The total number of process parameters is $81(9 * 9)$; thus, the dimensionality of parameters is much higher than in the 6T-SRAM case. We initially perform 3,000 samplings and simulations to model the initial

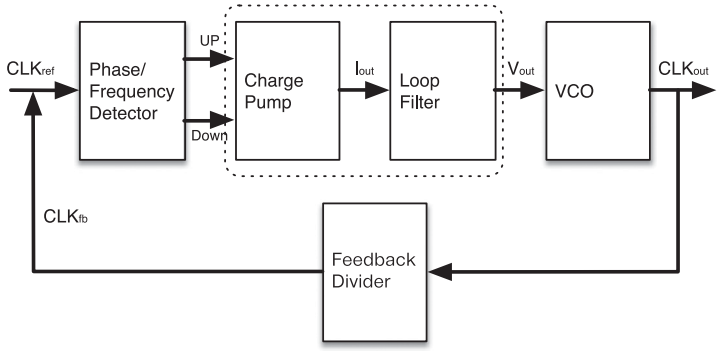


Fig. 13. A functional diagram of the PLL circuit.

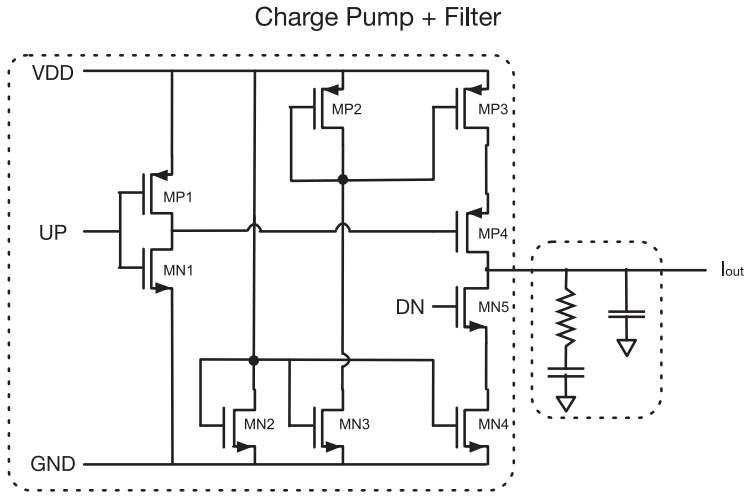


Fig. 14. Schematic representations of the charge pump and filter.

Table V. Comparison of the Accuracy and Efficiency for the Charge Pump Circuit

	Failure probability	**# Sim. runs**	**Speed-up(x)**	**Error (%)**
Monte Carlo(MC)	2.300E-04	1 million	–	–
Rare Event Microscope (REscope)	3.337E-04	4875	205.1	45.09
Recursive Statistical Blockade (RSB)	2.245-04	10432	95.9	2.39
Proposed method (EliteScope)	2.052-04	6263	159.7	10.78

performance distribution accurately. Similar to the 6T-SRAM case, we perform our algorithm twice with 97% percentile bound ($P(Y \geq t_1) = 0.03$, $P(Y \geq t_2) = 0.0009$).

The result is summarized in Table V. Our approach requires only 6263 Spice simulation runs for estimating the failure probability of a 3-sigma region with 10.78%

Table VI. Estimated In-Spec Guidance of Parameters for the Charge Pump Circuit

Rank	**Parameter @MOSFET**	**Initial condition (μ, σ)**	**In-spec Guidance [MIN, MAX]**
1	*ndep*@MN1	(2.8E+18,0.1)	[1.74E+18,3.78E+18]
2	*ndep*@MP2	(2.8E+18,0.1)	[1.68E+18,3.73E+18]
3	*ndep*@MP4	(2.8E+18,0.1)	[1.78E+18,3.81E+18]
4	*ndep*@MN5	(2.8E+18,0.1)	[1.78E+18,3.77E+18]
5	*ndep*@MP3	(2.8E+18,0.1)	[1.88E+18,3.80E+18]
Failure probability	0.0009($= t_2$)	**Estimation Error (%)**	**2.00**

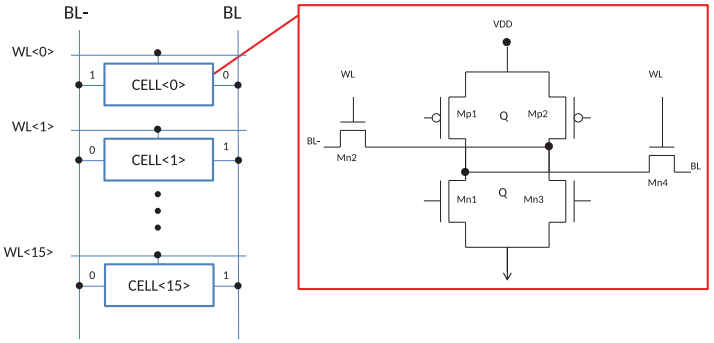


Fig. 15. The schematic of a 16b 6T-SRAM column.

relative error compared to the traditional MC method. Even though RSB achieves better accuracy with only a 2.39% error, it runs nearly 4000 more simulations than EliteScope. REscope requires 4875 simulation runs, but it produces significantly large errors compared to the MC method.

Table VI shows that the proposed method makes the decision for in-spec conditions of process parameters with a 98% confidence level by managing only the first 5 ranked parameters of 81.

The tail distribution of *EliteScope* in a 3-sigma failure region is much closer to MC than REscope, as we can see in Figure 12.

4.4. 16b 6T-SRAM Column Failure Rate Diagnosis

To illustrate the scalability of the proposed method on large analog circuits, we perform a comparison on one large 16b 6T-SRAM column circuit (1b line), as shown in Figure 15. In this example, we treat the delay of discharging $\overline{BL}$ as the failure criterion. To mimic the worst-case scenario, in which the impact of leakage current can be maximized, logic 0 is stored in $cell < 0 >$ and the rest of the cells store logic 1. In the reading operation, only $cell < 0 >$'s word line is turned on while all other word lines are turned off. We choose the same process parameters used in one SRAM cell experiment. The model parameters are independent in different cells. As a result, we have 432 random variables (16 cells * 27 random variables) that make this case a good example for scalability study. We run 6,000 samples to capture the circuit behavior. The same failure criterion is set as $t_c = 2.3 \times 10^{-4}$, which is about the 3-sigma in terms of the circuit yield. The proposed method iterates twice with 97% percentile bound as a slope guard to separate the failure region from the initial distribution. Hence, threshold bounds t_1 and t_2 are calculated as $\mathrm{P}(\mathrm{Y} \geq t_1) = 0.03$ and $\mathrm{P}(\mathrm{Y} \geq t_2) = 0.0009$, respectively.

Table VII. Accuracy Improvement Comparisons with REscope and RSB for 16b 6T-SRAM Column Case

# of Tail Fitting Samples		248	1392	6346
Monte Carlo Reference (MC)	**FP**	2.3E-04	2.3E-04	2.3E-04
	Error(%)	0	0	0
Rare Event Microscope (REscope)	**FP**	11.4E-04	6.29E-04	2.61E-04
	Error(%)	395.65	173.48	13.48
Recursive Statistical Blockade (RSB)	**FP**	7.705E-04	5.569E-04	2.39E-04
	Error(%)	235.00	142.13	3.91
Proposed Method (EliteScope)	**FP**	6.47E-04	3.385E-04	2.33E-04
	Error(%)	181.30	47.17	1.30

We set the same number of tail fitting samples for REscope, RSB, and *EliteScope* (the actual samples used to fit the tail distribution) so that we can fairly compare their accuracy. The estimated failure probability and their errors obtained from the three methods are shown in Table VII. All results are compared against the results from the MC method with one million runs, which give the failure probability as 2.3×10^{-4}, the gold standard reference for all the other methods.

In Table VII, the first row indicates the number of tail-fitting samples each approach uses. For each column, two terms are given for each method. The first term is the absolute failure probability (**FP**) obtained by the different methods; the second term is the relative failure probability error rate against the MC method.

When a small number of tail-fitting samples are used (only a few hundred), all methods result in large errors since the small number of samples cannot build a reliable model for the tail distribution. By using more tail-fitting samples, overall performance will be naturally improved. But still, the proposed method presents good performance with the lowest estimation relative error among all approaches.

We note that, by applying the elite learning sampling selection, we select 1392 samples with higher effectiveness out of 6961 samples generated for tail distribution fitting. Furthermore, the selected elite samples are quite effective for capturing tail distribution precisely. Compared to the REscope method, the proposed method achieves $3.67X$ improvement in accuracy using the same simulation costs.

When 6364 tail-fitting samples are generated to fit the tail distribution, all methods obtain a better approximation to the gold standard reference (very close to t_c) while *EliteScope* achieves the lowest error – 1% error compared to the standard MC simulation. Note that all results are obtained based on GPD fitting in this case. In this case, *EliteScope* is about $10X$ more accurate than the REscope.

4.5. Classifier Computation Complexity and Performance Convergence Analysis

In our implementation part, we use the Nu-Support Vector Classification (*NuSVC*) as our classifier. It is a built-in classifier function in the *scikit − learn* Python machine-learning package. It is a nonlinear C-support vector machine classifier. The computation complexity of the SVM is typically more than quadratic ($O(n^2)$), where n is the number of training samples. Depending on the test case and parameter selection, the computation time spent on classifier training varies. The total computation time of *EliteScope* is mainly spent in two parts: (1) classifier training and (2) NGSPICE circuit simulation. In low-dimensional test cases, NGSPICE simulation is fast due to the simple circuit netlist. Thus, classifier training dominates the time cost since *EliteScope* would run fewer NGSPICE simulations (usually 15% to 30% of sample candidates) by applying the Elite Learning Sample Selection scheme. But for the 16b SRAM circuit, which is a high-dimensional variable case, both classifier training cost and NGSPICE simulation costs drastically increase. One reason is the nonlinear computational

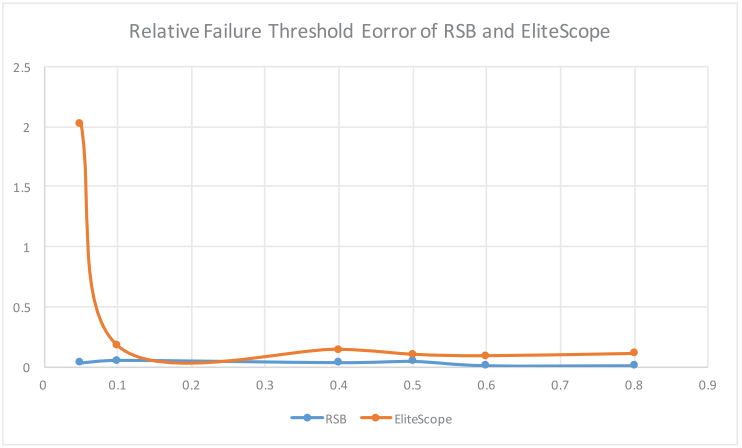


Fig. 16. Estimated Failure Threshold Error of RSB and EliteScope.

complexity of *NuSVC*. As we use more training samples to better capture the circuit behavior in high-dimensional space, more classifier training time is consumed. On the other hand, performing one single NGSPICE simulation costs 4s due to the large circuit size. The total NGSPICE simulation consumption time required by RSB is $3X$ that of classifier training time, while *EliteScope* further sorts out those samples that are more worthy to simulate. The proposed method can save over 50% of total computation time, which makes it even more time efficient in the high-dimensional case while keeping an acceptable accuracy level.

To further prove the feasibility of *EliteScope*, we repeatedly perform single-bit SRAM tests by using different values of selection ratio r. The failure region thresholds estimated by RSB and *EliteScope* are compared. We perform two separate tests for a given selection ratio and take the average of the failure threshold estimation values as the data in the figure.

Figure 16 shows the absolute estimation error of failure thresholds of RSB and *EliteScope*. Even though *EliteScope* encounters over 200% error when selection ratio r is set to be 0.05, it is acceptable since samples are still out of number and some highly weighted samples are not considered. After 10% of samples are used, the estimation error of *EliteScope* quickly converges to RSB but still exists due to its nature limitation. Figure 17 illustrates the relative error of *EliteScope* compared to the RSB method, with the selection range between 0.1 to 0.8. One observation is that the relative error of *EliteScope* does decrease as more samples are used, but only a maximum of 50% decrease of relative error at a cost of $5X$ simulation consumption occurs. Those low-weight samples provide a very limited contribution to estimating failure threshold. The result further supports that Elite Learning Sample Selection plays a smart role in selecting samples with great efficiency in failure region estimation.

5. CONCLUSIONS

In this article, we present a novel statistical diagnosis method for rare failure events. The proposed method introduced two new techniques to speed up the failure analysis while providing the in-spec guidance of process parameters. First, the proposed method applies the smart sample selection method to reduce the additional simulation cost during the iterative failure region locating process. Second, the new approach can provide safe design space of parameters, which can help design to improve the yield

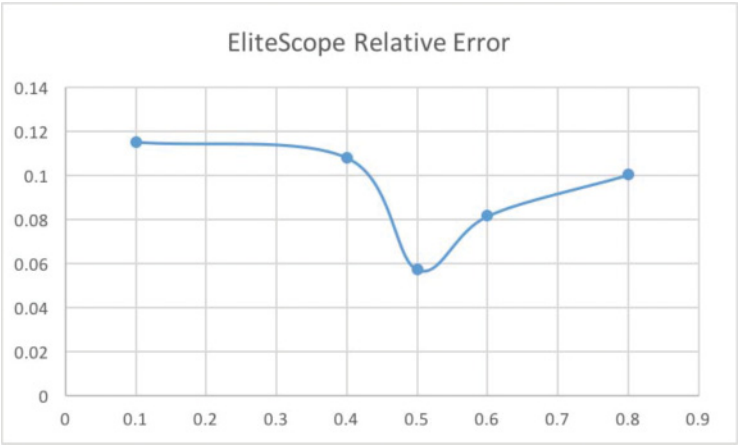


Fig. 17. Relative Failure Threshold Error of RSB and EliteScope.

and meet the target performance of design. Experiments on four circuit examples show that *EliteScope* achieves a significant improvement on failure region estimation in terms of accuracy and simulation cost over traditional approaches. The 16b 6T-SRAM column example also shows that the new method is scalable for handling large problems with a large number of process variables.

REFERENCES

K. Agarwal and S. Nassif. 2006. Statistical analysis of SRAM cell stability. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC'06)*. 57–62.

B. H. Calhoun, Yu Cao, Xin Li, Ken Mai, L. T. Pileggi, R. A. Rutenbar, and Kenneth L. Shepard. 2008. Digital circuit design challenges and opportunities in the era of nanoscale CMOS. *Proceedings of the IEEE* 96, 2, 343–365.

F. Gong, X. Liu, H. Yu, S. X. D. Tan, and L. He. 2012. A fast non-Monte-Carlo yield analysis and optimization by stochastic orthogonal polynomials. *ACM Transactions on Design Automation of Electronic Systems* 17, 1, 10:1–10:23.

T. Hastie. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

D. E. Hocevar, M. R. Lightner, and Timothy N. Trick. 1983. A study of variance reduction techniques for estimating circuit yields. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 2, 3, 180–192.

J. R. Hosking. 1985. Algorithm as 215: Maximum-likelihood estimation of the parameters of the generalized extreme-value distribution. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 34, 3, 301–310.

Stephen Joe and Frances Y. Kuo. 2003. Remark on algorithm 659: Implementing Sobol's quasirandom sequence generator. *ACM Transactions on Mathematical Software* 29, 1, 49–57.

H. Masuda, S. Ohkawa, A. Kurokawa, and M. Aoki. 2005. Challenge: Variability characterization and modeling for 65- to 90-nm processes. In *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC'05)*. 593–599.

D. Montgomery. 2013. *Design and Analysis of Experiments*. Wiley, Hoboken, NJ.

S. Mukhopadhyay, H. Mahmoodi, and K. Roy. 2004. Statistical design and optimization of SRAM cell for yield enhancement. In *Proceedings of the International Conference on Computer Aided Design (ICCAD'04)*. 10–13.

S. Nassif, R. Kanj, and R. Joshi. 2006. Mixture important sampling and its application to the analysis of SRAM designs in the presence of rare failure events. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC'06)*. 69–72.

Paolo Nenzi and V. Holger. 2010. Ngspice users manual. *Version 25plus*.

M. Qazi, M. Tikekar, L. Dolecek, D. Shah, and A. Chandrakasan. 2010. Loop flattening and spherical sampling: Highly efficient model reduction techniques for SRAM yield analysis. In *Proceedings of Design, Automation and Test In Europe (DATE'10)*. 801–806.

Ruijing Shen, Sheldon X.-D. Tan, Hai Wang, and Jinjun Xiong. 2012b. Fast statistical full-chip leakage analysis for nanometer VLSI systems. *ACM Transactions on Design Automation of Electronic Systems* 17, 4, Article 51, 19 pages. DOI: http://dx.doi.org/10.1145/2348839.2348855

Ruijing Shen, Sheldon X.-D. Tan, and Hao Yu. 2012a. *Statistical Performance Analysis and Modeling Techniques for Nanometer VLSI Designs*. Springer Publishing.

A. Singhee and R. A. Rutenbar. 2009. Statistical blockade: Very fast statistical simulation and modeling of rare circuit events and its application to memory design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28, 8, 1176–1189.

A. Singhee, Jiajing Wang, B. H. Calhoun, and R. A. Rutenbar. 2008. Recursive statistical blockade: An enhanced technique for rare event simulation with application to SRAM circuit design. In *International Conference on VLSI Design*. 131–136.

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 International Conference on Computer Aided Design (ICCAD'14)*. 324–331.

S. Sun, X. Li, H. Liu, K. Luo, and B. Gu. 2013. Fast Statistical Analysis of Rare Circuit Failure Events via Scaled-Sigma Sampling for High-Dimensional Variation Space. In *Proceedings of the International Conference on Computer Aided Design (ICCAD'13)*. 478–485.

J. R. Wallis and J. R. Hosking. 1987. Parameter and quantile estimation for the generalized Pareto distribution. *Technometrics* 29, 3, 339–349.

Wei Wu, Fang Gong, Gengsheng Chen, and Lei He. 2014a. A fast and provably bounded failure analysis of memory circuits in high dimensions. In *Proceedings of the Asia South Pacific Design Automation Conference (ASPDAC'14)*. 424–429.

Wei Wu, Wenyao Xu, Rahul Krishnan, Yen-Lung Chen, and Lei He. 2014b. REscope: High-dimensional statistical circuit simulation towards full failure region coverage. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC'14)*. 1–6.

Received June 2015; revised November 2015; accepted January 2016