# Efficient Importance Sampling for High-sigma Yield Analysis with Adaptive Online Surrogate Modeling

Jian Yao, Zuochang Ye and Yan Wang
Tsinghua National Laboratory for Information Science and Technology
Institute of Microelectronics, Tsinghua University, Beijing 100084, China
Email: jianyao.thu@gmail.com; {zuochang, wangy46}@tsinghua.edu.cn

***Abstract*—Massively repeated structures such as SRAM cells usually require extremely low failure rate. This brings on a challenging issue for Monte Carlo based statistical yield analysis, as huge amount of samples have to be drawn in order to observe one single failure. Fast Monte Carlo methods, e.g. importance sampling methods, are still quite expensive as the anticipated failure rate is very low. In this paper, a new method is proposed to tackle this issue. The key idea is to improve traditional importance sampling method with an efficient online surrogate model. The proposed method improves the performance for both stages in importance sampling, i.e. finding the distorted probability density function, and the distorted sampling. Experimental results show that the proposed method is 1e2X∼1e5X faster than the standard Monte Carlo approach and achieves 5X∼22X speedup over existing state-of-the-art techniques without sacrificing estimation accuracy.**

## I. Introduction

With technology feature size scaling towards the physical limit, variations due to uncertainties are becoming a growing concern in IC designs. This in general requires statistical methods such as Monte Carlo method for yield analysis. Statistical methods are usually time-consuming. To be worse, highly repeated cells such as SRAM cells usually require an extremely low failure rate ($10^{-6} \sim 10^{-8}$) in the per cell basis to ensure a reasonably moderate yield for the whole chip. For these high-sigma applications, standard Monte Carlo method requires huge amount of samples ($10^7 \sim 10^9$) to achieve the wanted accuracy, with each sample corresponding to a SPICE simulation run. The reason is that most of the samples fall into the feasible region, and only an extremely small fraction of samples are in the failure region.

Accelerated Monte Carlo methods are thus required. Most of existing acceleration methods are based on the importance sampling methods [1]–[3]. The key idea of importance sampling methods is to sample the process parameter space based on a distorted probability density function (PDF) to increase the probability of observing failures.

Importance sampling provides a framework towards solving the computational barrier. However, the performance of importance sampling depends highly on its implementation details, especially in how to define the distorted PDF. In addition, even with importance sampling, the required samples are still in the order of several thousands to tens of thousands, thus the computational cost spent on circuit simulation is still high.

In this paper, we propose a fast high-sigma yield analysis method based on the importance sampling framework. The key idea is to adaptively replace circuit simulation with online trained surrogate model. This surrogate model is used in both stages of importance sampling for reducing the computational cost, i.e. finding the distorted PDF, and the distorted sampling itself.

The rest of this paper is organized as follows. In section II, the relevant background materials will be presented. In section III, the proposed method will be discussed. Experimental results are given in section IV. Conclusions are given in the section V.

## II. Background

Suppose $X = [x_1, x_2, ..., x_D]^T$ is a $D$-dimensional random variable modeling process variations and its joint PDF can be defined as $p(X)$. Such random variables include the variations of threshold voltage $V_{th}$, oxide thickness $T_{ox}$, and gate length $L_{eff}$, etc. Typically, $X$ is a multivariate Normal distribution [1]. Without loss of generality, we further assume that the random variables $[x_1, x_2, ..., x_D]$ in the vector $X$ are mutually independent and standard Normal (i.e., with zero mean and unit variance):

$$p(X) = \prod_{i=1}^{D} \left[ \frac{1}{\sqrt{2\pi}} \cdot \exp\left( -\frac{1}{2} x_i^2 \right) \right] \tag{1}$$

any correlated random variables that are jointly Normal can be transformed to the independent random variables in equation (1) by principal component analysis (PCA) [4].

The failure rate $P_f$ of a circuit can be mathematically represented as

$$P_f = \int_{\Omega} p(X) \cdot dX \tag{2}$$

where $\Omega$ denotes the failure region, i.e., the subset of the variation space where the performances of interest (e.g., read noise margin, write delay for SRAM cells) do not meet the specification. The corresponding yield is $1 - P_f$.

This work was supported in part by the Major State Basic Research Development Program of China (973 Program) under Grants 2010CB327403 and 2011CBA00604 and in part by the National Natural Science Foundation of China under Grants 61176034 and 61106031.

978-3-9815370-0-0/DATE13/©2013 EDAA

Alternatively, the failure rate in equation (2) can be defined as:

$$P_f = \int_{-\infty}^{\infty} I\left(X\right) \cdot p\left(X\right) \cdot dX \tag{3}$$

where $I(X)$ represents the indicator function:

$$I\left(X\right) = \left\{ \begin{array}{ll} 1 & X \in \Omega \\ 0 & X \notin \Omega \end{array} \right. \tag{4}$$

Basically, the failure rate $P_f$ can be estimated by the standard Monte Carlo analysis [5]. The key idea is to draw $N$ random samples $\{\widetilde{X}_1, \widetilde{X}_2, ..., \widetilde{X}_N\}$ from $p(X)$, and then obtain the circuit performances $\{\widetilde{f}_1, \widetilde{f}_2, ..., \widetilde{f}_N\}$ by SPICE-simulations to calculate indicator function $\{I(\widetilde{X}_1)$, $I(\widetilde{X}_2)$, ..., $I(\widetilde{X}_N)\}$. The failure rate can be calculated as

$$\widetilde{P}_f^{MC} = \frac{1}{N} \sum_{i=1}^{N} I\left(\widetilde{X}_i\right) \tag{5}$$

For high-sigma applications, the failure rate $P_f$ in equation (3) is extremely small and most random samples created in Monte Carlo analysis do not fall into the failure region $\Omega$. Hence, a large number of samples are needed by the Monte Carlo method to accurately estimate the failure rate. Note that each sampling point corresponds to a SPICE-simulation run, which is expensive. The aforementioned Monte Carlo method is extremely expensive, or even infeasible, when applied to the high-sigma yield analysis.

The importance sampling techniques are widely reported in accelerating the high-sigma yield analysis [1]–[3]. It aims at directly generating a large number of random samples in the failure region by using a distorted PDF $q(X)$. In this case, the failure rate can be expressed as:

$$P_f = \int_{-\infty}^{\infty} \frac{I\left(X\right) \cdot p\left(X\right)}{q\left(X\right)} \cdot q\left(X\right) \cdot dX \tag{6}$$

Typically, importance sampling is implemented in two steps. In the first step, the distorted PDF $q(X)$ should be generated. Then, in the second step, $M$ sampling points $\{\widetilde{X}_1, \widetilde{X}_2, ..., \widetilde{X}_M\}$ are drawn from $q(X)$ and the failure rate can be estimated by:

$$\widetilde{P}_f^{IS} = \frac{1}{M} \cdot \sum_{i=1}^{M} \frac{I\left(\widetilde{X}_i\right) \cdot p\left(\widetilde{X}_i\right)}{q\left(\widetilde{X}_i\right)} \tag{7}$$

The key of importance sampling methods is how to generate the distorted PDF $q(X)$. In theory, the optimal PDF $q^{opt}(X)$ is described in equation (8) [6], which only needs one sample in the $q^{opt}(X)$ to ensure the accuracy:

$$q^{opt}\left(X\right) = \frac{I\left(X\right) \cdot p\left(X\right)}{P_f} \tag{8}$$

Obtaining the optimal PDF is infeasible in practical applications, but we would notice from equation (8) that the sample should be generated from the most likely failure region as discussed in [1]. Most existing importance sampling methods [1]–[3] are mainly based on this idea. For example, in [2], the maximum probability density of the process condition subject to violating at least one specification will be selected as the optimal mean-shift vector. And then, the distorted PDF is based on this vector, which is illustrated in Fig. 1. In this two-dimensional example, the optimal mean-shift vector $X_{opt} = [x_{1,opt}, x_{2,opt}]^T$ is used to determine the most likely failure region and the distorted PDF $q(X)$ is based on it.

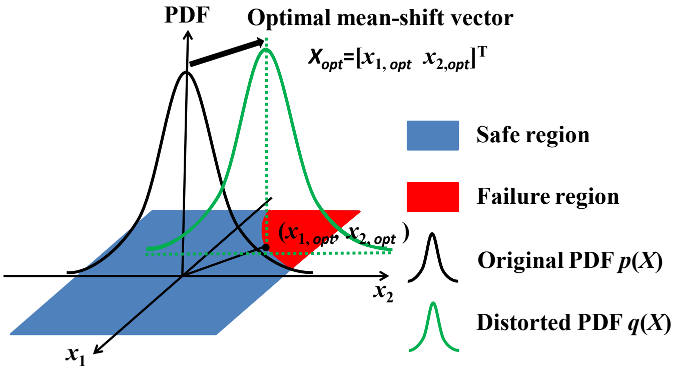

PDF
Optimal mean-shift vector
$X_{opt}$=[$x_{1,opt}$ $x_{2,opt}$]$^T$
Safe region
Failure region
($x_{1,opt}$, $x_{2,opt}$ )
Original PDF $p(X)$
Distorted PDF $q(X)$
$x_2$
$x_1$

Fig. 1. The two-dimensional example to illustrate the importance sampling.

The importance sampling technique reduces the computational cost in high-sigma yield analysis. However, the samples (i.e., the SPICE-simulations) required in both steps of importance sampling are still expensive [1]–[3], and the time-consuming SPICE-simulation is by far the bottleneck of the whole analysis.

## III. Proposed Method

The proposed method follows the basic importance sampling framework introduced in Section II, i.e., to first find the optimal mean-shift vector and then, perform sampling based on the distorted PDF parameterized with the optimal mean-shift vector.

### A. Finding optimal mean-shift vector with optimization

According to equation (1), the maximum probability density occurs where the distance to the origin is minimized [7]. Thus finding the optimal mean-shift vector can be translated to solving the following optimization problem:

$$\begin{array}{rl} \min & \|X\| \\ \text{subject to}: & X^- < X < X^+ \\ & I\left(X\right) = 0 \end{array} \tag{9}$$

where $\|\bullet\|$ is the 2-norm. The choice of optimization boundary $X^{\pm}$ is based on the idea that the sampling points are infeasible out of the boundary (e.g., $X^{\pm} = \pm 8$sigma for one-dimensional problem, which has the probability density of $10^{-15}$).

The problem (9) is essentially a global optimization problem. The best class of algorithms for solving such is the population-based optimization algorithm, which is illustrated in Fig. 2.

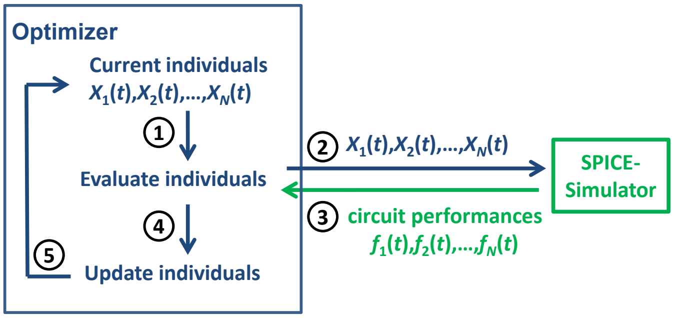


Fig. 2. Illustration of optimization to obtain the optimal mean-shift vector.

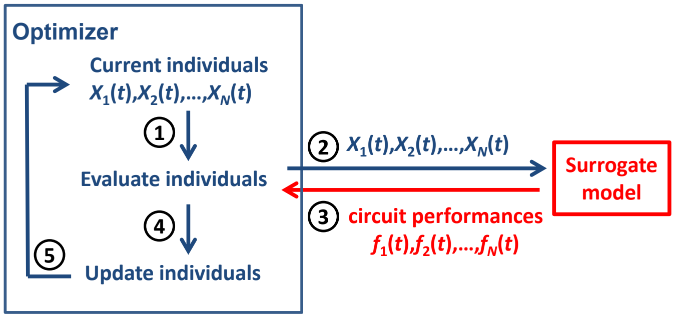


Fig. 3. Illustration of optimization combined with surrogate models.

At each generation (iteration) $t$, suppose $[X_1(t), X_2(t), ..., X_N(t)]$ are the current individuals, where $X_i(t)$ is a sample in process parameter $X$, and $N$ is the number of individuals in the population.

Firstly, each individual will be evaluated by the SPICE-simulation to calculate the circuit performance $f_i(t)$ $(i = 1, 2, ..., N)$ and corresponding indicator function $I(X_i(t))$ $(i = 1, 2, ..., N)$ according to equation (4). Then, the population will be updated based on the adopted population-based optimization algorithm. The next iteration is then performed until the optimal individual (i.e., the optimal mean-shift vector) satisfies the convergence condition.

In general, there are a lot of population-based optimization methods can be used, e.g., genetic algorithm [8] and particle swarm algorithm [9]. In this work, we use the differential evolution algorithm [10] as the optimizer to solve this optimization problem. The algorithm tries to find the globally optimal solution by utilizing the distribution of individuals in the search space and optimal individual as search direction.

The key problem of population-based optimization is that usually a large amount of individuals are required in order to obtain a reasonably good solution, and each individual needs one time of SPICE-simulation in each iteration. As there are always several iterations to obtain the optimal mean-shift vector, the total computational cost is still expensive.

### B. Offline surrogate model to accelerate optimization

The key to reduce the computational cost is to reduce the number of time-consuming SPICE-simulations. To accelerate the optimization, a general idea is to first train surrogate models to approximate the correlation between circuit performances and process parameters. And then, the optimization can be cooperated with the generated surrogate models so that there is no SPICE-simulation used in the optimization, which is illustrated in Fig. 3.

There are quite a few existing modeling techniques, e.g., response surface modeling [11] and artificial neural network modeling [12], which can be used to generate the surrogate model. In this work, we use radial basis function network [13] as the surrogate model to approximate the correlation between circuit performances ($f$) and process parameters ($X$), which can be expressed in following equation[1]:

$$f(X) = \sum_{i=1}^{c} \omega_i H_i = \sum_{i=1}^{c} \omega_i \exp\left\{ \frac{-\|X - M_i\|^2}{2\sigma_i^2} \right\} \qquad (10)$$

where $c$ is the number of training samples; $\omega_i$, $M_i$ and $\sigma_i$ are coefficients of $i$th Gaussian function $H_i$.

With several training samples $\{\widetilde{X}_1, \widetilde{f}_1\}$, $\{\widetilde{X}_2, \widetilde{f}_2\}$, ..., $\{\widetilde{X}_K, \widetilde{f}_K\}$ obtained by SPICE-simulations, the coefficients in equation (10) can be determined by orthogonal least squares learning algorithm [14], and the radial basis function network model can be built.

As generating each training sample needs one time of SPICE simulation, it would be preferable if the number of training samples can be as small as possible. In practice, we found that the trade-off between model accuracy and number of samples used for training the model is usually poor. Thus either the accuracy is poor or the required number of SPICE simulations is very large.

Meanwhile, the training samples are averaged in the search space $X$ to train a model before it is used in the optimization. It is inefficient when combined with the optimization. As the purpose of optimization is to find the optimal individual, more training samples should be used in the region where should be explored (i.e., the region around the optimal mean-shift vector).

### C. Optimization with online training surrogate model

The idea of online surrogate model is inspired by the observation that during the optimization, the requirement for model accuracy varies. The model accuracy should be high enough where it is close to the optimal mean-shift vector $X_{opt}$, and the model accuracy can be relatively low where it is far from $X_{opt}$. This procedure can be made more adaptively, as described in the following.

Before the optimization, a small number of training samples are generated to train an initial coarse surrogate model. As the optimization relies on a good initial guess, to improve the convergence, we use the Latin-hypercube sampling method [5]

[1] For simplicity of our presentation, $f$ represents any performance that should be considered. It is understood that our method can be applied to multiple constraints scenario by training corresponding model for each performance.

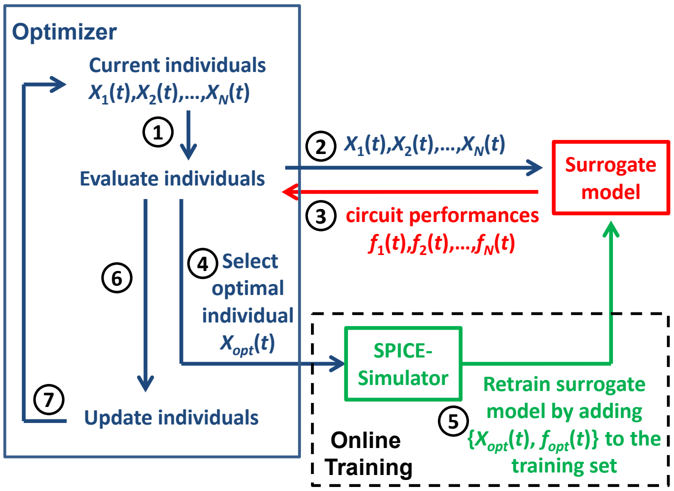

Optimizer
Current individuals
$X_1(t), X_2(t), \ldots, X_N(t)$
1
Evaluate individuals
2 $X_1(t), X_2(t), \ldots, X_N(t)$
Surrogate model
3 circuit performances
$f_1(t), f_2(t), \ldots, f_N(t)$
4 Select optimal individual $X_{opt}(t)$
6
7
Update individuals
SPICE-Simulator
Retrain surrogate model by adding $\{X_{opt}(t), f_{opt}(t)\}$ to the training set
5
Online Training

Fig. 4. Illustration of optimization with online training surrogate model.

with uniform distribution in the search space $X$ to generate the initial training samples.

Then, each iteration of optimization begins combined with training the surrogate model as illustrated in Fig 4. At each iteration $t$, the current individuals will first be evaluated by the current surrogate model. Among these individuals, the failure individual with minimum 2-norm is selected as the current optimal individual $X_{opt}$ and this individual is evaluated by SPICE-simulation. Next, $X_{opt}$ will be added to the training set and the surrogate model will be trained and renewed, which will be used in the next iteration. At the same time, the optimizer will update the population based on the optimization algorithm.

Notice there is only one SPICE-simulation in each iteration (for the optimal individual in each iteration), the cost of the optimization can thus be highly reduced. Meanwhile, as only the current optimal individual in each iteration is used to train the surrogate model, the accuracy of the model will be sufficient in the region around the optimal mean-shift vector. The efficiency is therefore achieved as most of training samples are used in the region which should be explored, while offline training surrogate model cannot achieve it.

### D. Sampling from the distorted PDF

After the optimal mean-shift vector $X_{opt}$ is generated, the final trained surrogate model is also generated at the same time, which can be used in this sampling stage to replace the time-consuming SPICE-simulation.

The distorted PDF is based on the generated mean-shift vector $X_{opt}$ from the optimization. To ensure a good proportion of samples that are infeasible (i.e., outside of specifications), the sampling algorithm shifts the mean of $p(X)$ onto $X_{opt}$, to get $p(X - X_{opt})$. To keep the theoretical maximum number of samples low, some samples are drawn from $p(X)$ [15]. Therefore the distorted PDF used in this method is:

$$q(X) = \alpha p(X) + (1 - \alpha) p(X - X_{opt}) \tag{11}$$

where $0 \leq \alpha \leq 1$.

For each sample, the indictor function $I(X)$ should be evaluated. In this method, we use the following strategy to calculate the $I(X)$:

- When the circuit performances predicted by the final trained surrogate model are far from the specification (e.g., read noise margin (RNM) [16] of the SRAM cell is far from 0), $I(X)$ will be evaluated from the surrogate model.
- When the circuit performances predicted by the surrogate model are near the specification, the SPICE-simulation is used to obtain circuit performance and corresponding $I(X)$.

This means that when the value predicted by surrogate model is close to failure boundary, the predicted value is suspected and the SPICE-simulation should be used. We will show in section IV that even if there is no SPICE-simulations used in calculating the $I(X)$, the model accuracy is enough (>98.5%) to ensure an accurate yield evaluation result. Only a few SPICE-simulations (6% in total samples) can further enhance the sampling accuracy up to 100% and thus 100% accuracy in yield calculation.

### E. General framework of the proposed method

Based on the aforementioned techniques, the overall algorithm of the proposed method can now be constructed. The detailed description is in Algorithm 1.

**Algorithm 1** Framework of fast yield analysis method.

```
Step 0: Generate the initial training set by Latin-hypercube sampling and train an initial surrogate model. Meanwhile, generate the initial optimization population.
Step 1: The circuit performance of current population are calculated based on the surrogate model.
Step 2: Add the current optimal individual to the training set. Obtain the corresponding circuit performance by SPICE-simulation.
Step 3: Retrain the surrogate model.
Step 4: Perform optimization to update the optimization population.
Step 5: Check whether the stopping criterion is met. If yes, obtain the optimal mean-shift vector and final trained surrogate model, and go to step 6. Otherwise, go to step 1.
Step 6: Generate distorted PDF as described in equation (11).
Step 7: Sample from the distorted PDF using the strategy discussed in section III-D.
Step 8: Obtain the failure rate and corresponding yield of the analyzed circuit by equation (7).
```

## IV. Experimental Results

In this section, a 6-T SRAM cell designed with a commercial 40 nm process is used to demonstrate the accuracy and efficiency of the proposed method. Fig. 5 shows the schematic of the SRAM cell. The local $V_{th}$ mismatch of each transistor is considered as the process variables. All process variables are mutually independent and standard Normal. The circuit performance is chosen as the read noise margin (RNM) which characterizes the stability of the SRAM cell [16]. When RNM is less than zero, the data retention failure happens. Note that the same algorithm can be applied to other variation sources (e.g., $L_{eff}$ and $T_{ox}$) and other circuit performances (e.g. write delay and access time) as well. Four specific designed SRAM cells with different failure rates $\{\mathrm{SRAM}_1, \mathrm{SRAM}_2, \mathrm{SRAM}_3, \mathrm{SRAM}_4\}$ are used as the experimental examples. The examples run on a PC with Intel 2.16GHz CPU.

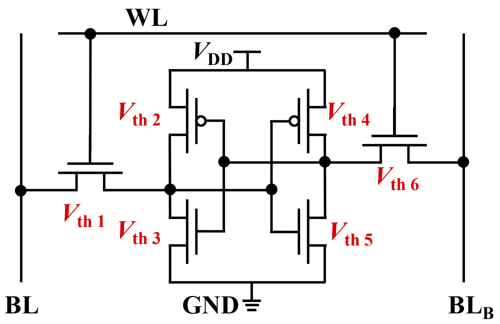


Fig. 5. The circuit schematic of the 6-T SRAM cell.

### A. Validate the accuracy of the proposed method

First, we evaluate the accuracy of the final trained surrogate model as it will be used in the second step. After the optimal mean-shift vector $X_{opt}$ and final model are generated in the first step for the four designed SRAM cells, $10^5$ test samples are randomly generated in the $q(X)$ as expressed in equation (11). The indictor functions $I(X)$ of the test samples are calculated by both the model and SPICE-simulation to evaluate the accuracy of the surrogate model, defined as the percentage of samples accurately predicted, which is presented in the Table I.

TABLE I
MODEL AND METHOD ACCURACY IN FOUR DESIGNED CIRCUITS

| SRAM cell | Model accuracy | Percentage of added SPICE-simulations | Method accuracy |
|---|---|---|---|
| $\mathrm{SRAM}_1$ | 98.90% | 6.42% | 100% |
| $\mathrm{SRAM}_2$ | 98.95% | 6.31% | 100% |
| $\mathrm{SRAM}_3$ | 98.69% | 6.33% | 100% |
| $\mathrm{SRAM}_4$ | 98.98% | 6.29% | 100% |

Based on the strategy discussed in section III-D, when the predicted absolute value of RNM is smaller than 5e-4, SPICE-simulation is used to calculate the $I(X)$. The percentage of added SPICE-simulations in $10^5$ samples and the corresponding method accuracy are also presented in Table I. The satisfied accuracy can be observed.

To show the accuracy clearly, we further compare the proposed method with the standard Monte Carlo (MC) Method [5] on the four SRAM examples for calculation the failure rate, which is shown in Fig 6 (a). The failure rate estimations from the two methods closely match each other, which again validates the estimation accuracy of our proposed method.

### B. Validate the efficiency of the proposed method

To show the efficiency of the proposed method, besides the Monte Carlo method, the exiting acceleration method: mixture importance sampling (MIS) [3], minimum-norm importance sampling (MNIS) [2], Gibbs sampling (GS) [1] are used to analyze the SRAM cells. The convergence condition is selected as the Figure-Of-Merit (FOM) [2] equals to 0.1 (i.e., 90% accuracy level with 90% confidence interval).

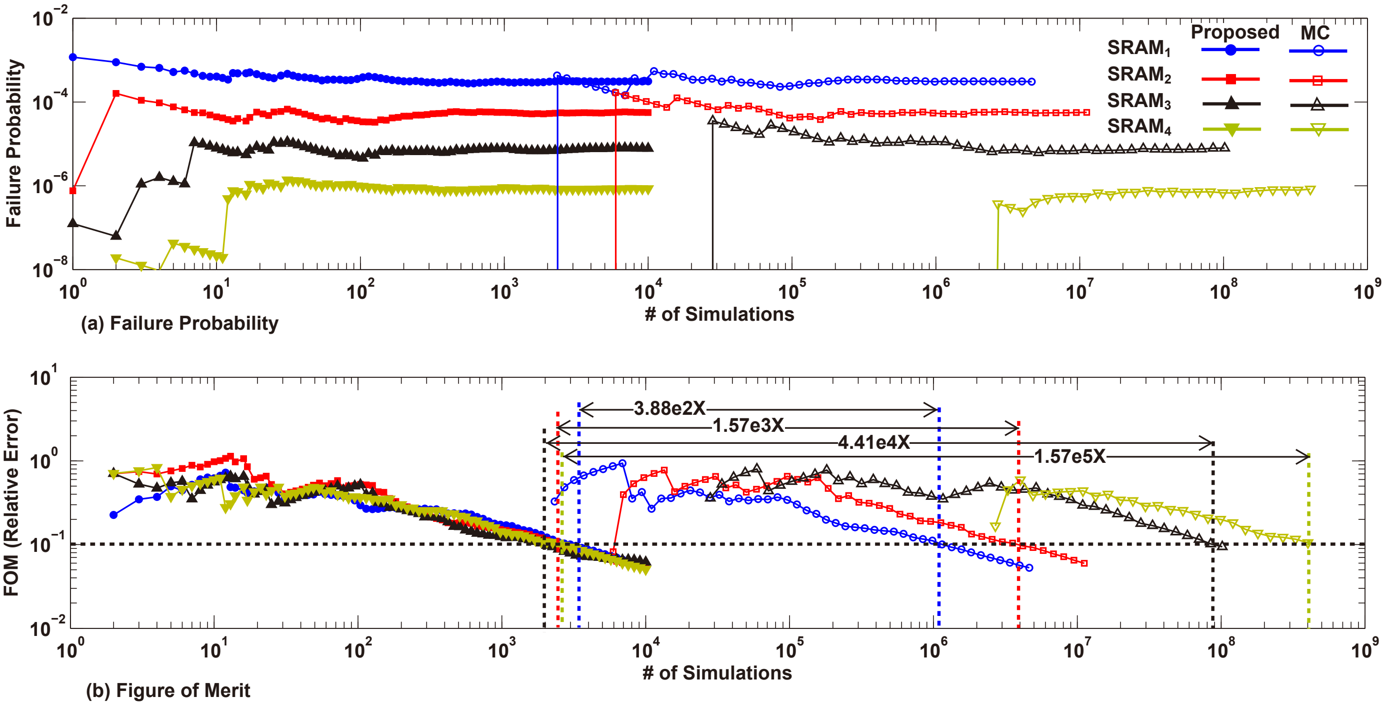
From Fig 6 (b), the required samples of the proposed method are 1e2X∼1e5X less than the standard Monte Carlo method. Meanwhile, the required samples of Monte Carlo increase as the failure rate is smaller, while the proposed method is independent with the specific circuits. It implies that the proposed method has the potential and ability to solve the high-sigma problems.

Table II shows the computational cost of mixture importance sampling (MIS) [3], minimum-norm importance sampling (MNIS) [2], Gibbs sampling (GS) [1] and the proposed method for $\mathrm{SRAM}_4$. An 5X∼22X accelerating rate of the proposed method compared with other exiting methods can be found and similar accelerating rates are also shown in other SRAM examples $\{\mathrm{SRAM}_1, \mathrm{SRAM}_2, \mathrm{SRAM}_3\}$.

We can find that although the required samples for the proposed method are similar or even more than the other exiting methods, most of the samples can be generated from the surrogate model in both steps of the importance sampling. The surrogate model is generally order of magnitude faster than SPICE-simulation as the surrogate model can be thought as the ”Black Box” only describing the surface relations between circuit performance and process variations. The computational cost of SPICE-simulation is by far the bottleneck of the whole analysis. The key of the efficiency of the proposed method is that the SPICE-simulation required for the importance sampling is sufficiently reduced.

## V. CONCLUSION

We propose a new fast high-sigma yield analysis method based on the importance sampling framework. The key idea of the proposed method is to accelerate the analysis by a novel online surrogate model. Population-based optimization algorithm is used as the core optimizer to find the optimal mean-shift vector of the distorted probability density function. During the optimization, a surrogate model is trained online with the optimization to highly reduce the number of SPICE-simulation. The trained model can also be used to accelerate the sampling. Experimental results have shown that the proposed method is 1e2X∼1e5X faster than the standard Monte Carlo approach and achieves 5∼22X speedup over



(a) Failure Probability

(b) Figure of Merit

Fig. 6. The estimated failure rate and FOM as a function of the number of samples by the proposed method (the second step) and standard Monte Carlo method for the four SRAM examples.

TABLE II
THE COMPUTATIONAL EFFICIENCY OF THE FOUR METHODS IN SRAM4

| Method | The first step | | | The second step | | | Totally cost | Speedup |
|---|---|---|---|---|---|---|---|---|
| | Samples | SPICE cost | Surrogate model cost | Samples | SPICE cost | Surrogate model cost | | |
| MIS [3] | 6600 | 6600×1.35s | 0 | 7465 | 4740×1.35s | 0 | 255.15min | 1X |
| MNIS [2] | 2400 | 2400×1.35s | 0 | 2501 | 1588×1.35s | 0 | 89.73min | 2.84X |
| GS [1] | 2385 | 2385×1.35s | 0 | 584 | 371×1.35s | 0 | 62.01min | 4.11X |
| Proposed | 2800 | 325×1.35s | 2475×8.24ms | 2473 | 159×1.35s | 2317×8.24ms | 11.55min | 22.09X |

other existing state-of-the-art importance sampling techniques without sacrificing the estimation accuracy.

## REFERENCES

[1] C. Dong and X. Li, "Efficient sram failure rate prediction via gibbs sampling," in *ACM/IEEE Design Automation Conference*, Jun. 2011, pp. 200 –205.

[2] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan, "Breaking the simulation barrier: Sram evaluation through norm minimization," in *IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2008, pp. 322 –329.

[3] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of sram designs in the presence of rare failure events," in *ACM/IEEE Design Automation Conference*, Jun. 2006, pp. 69 –72.

[4] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processe*. McGraw-Hill, 2001.

[5] X. Li, J. Le, and L. T. Pileggi, "Statistical performance modeling and optimization," *Foundations and Trends in Electronic Design Automation*, vol. 1, no. 4, pp. 331 – 480, Apr. 2006.

[6] G. Fishman, *A First Course in Monte Carlo*. Duxbury Press, 2005.

[7] T. McConaghy and P. Drennan, "Variation-aware custom ic design: Improving pvt and monte carlo analysis for design performance and parametric yield," in *Solido White Paper*, 2011.

[8] K. F. Man, K. S. Tang, and S. Kwong, *Genetic Algorithms: Concepts and Designs*. Springer, 1999.

[9] J. Sun, C.-H. Lai, and X.-J. Wu, *Particle Swarm Optimisation: Classical and Quantum Perspectives*. CRC Press, 2011.

[10] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution. A Practical Approach to Global Optimization*. Berlin, Germany: Springer, 2005.

[11] A. I. Khuri, *Response Surface Methodology And Related Topics*. World Scientific, 2006.

[12] B. Yegnanarayana, *Artificial Neural Networks*. PHI Learning Pvt. Ltd., 2004.

[13] S.-F. Su, C.-C. Chuang, C. Tao, J.-T. Jeng, and C.-C. Hsiao, "Radial basis function networks with linear interval regression weights for symbolic interval data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 1, pp. 69 –80, Feb. 2012.

[14] L. M. Surhone, M. T. Timpledon, and S. F. Marseken, *Radial Basis Function Network*. VDM Verlag Dr. Mueller e.K., 2010.

[15] A. Singhee and R. Rutenbar, "Statistical blockade: A novel method for very fast monte carlo simulation of rare circuit events, and its application," in *Design, Automation Test in Europe Conference Exhibition*, Apr. 2007, pp. 1 –6.

[16] K. Agarwal and S. Nassif, "The impact of random device variation on sram cell stability in sub-90-nm cmos technologies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 1, pp. 86 –97, Jan. 2008.