FUSIS: Fusing Surrogate Models and Importance Sampling for Efficient Yield Estimation

Yanfang Liu
School of Integrated Circuit Science and Engineering, Beihang University
Beijing, China
liuyanfang@buaa.edu.cn

Wei W. Xing*
School of Mathematical and Physical Science University of Sheffield
S3 7RH, UK
w.xing@sheffield.ac.uk

***Abstract*—As process nodes continue to shrink, yield estimation has become increasingly critical in modern circuit design. Traditional approaches face significant challenges: surrogate-based methods often struggle with robustness and accuracy, whereas importance sampling (IS)-based methods suffer from high simulation costs. To address these challenges simultaneously, we propose FUSIS, a unified framework that combines the strengths of surrogate-based and IS-based approaches. Unlike conventional surrogate-based methods that directly replace SPICE simulations for performance predictions, FUSIS employs a Deep Kernel support vector machine (SVM) as an approximation of the indicator function, which is further utilized to construct a quasi-optimal proposal distribution for IS to accelerate convergence. To further mitigate yield estimation bias caused by surrogate inaccuracies, we introduce a novel correction factor to adjust the IS-based yield estimation. Experiments conducted on SRAM and analog circuits demonstrate that FUSIS significantly improves accuracy by up to 24.84% (8.67% on average) while achieving up to 29.54× (10.30× on average) speedup in efficiency compared to seven state-of-the-art methods.**

***Index Terms*—Yield Estimation, Importance Sampling, Surrogate Model, Deep Kernel SVM**

I. Introduction

As integrated circuit technology advances, microelectronic devices are continuously shrinking to submicrometer scales, making random process variations—such as intra-die mismatches, doping fluctuations, and threshold voltage shifts—critical factors in circuit design. In modern circuit designs, especially with highly replicated structures like SRAM cell arrays, addressing yield concerns is vital. Accurate and efficient yield estimation methods are essential for assessing failure rates under specific process variations.

Monte Carlo (MC) simulation, the industry-standard baseline for yield estimation, involves running SPICE (Simulation Program with Integrated Circuit Emphasis) simulations with parameters drawn from the process variation distribution millions of times and counting failures to obtain a precise estimation. However, MC is computationally intensive and becomes impractical for scenarios where the failure rate is as low as 10^{-5}, which is common in a 45nm SRAM cell array.

To enhance yield estimation efficiency, importance sampling (IS)-based methods have gained prominence. Rather than sampling from the default normal distribution, IS-based methods leverage a proposal distribution to improve efficiency. One effective strategy involves shifting the sampling center of the normal distribution according to the Optimal Mean Shift Vector (OMSV), which is typically identified by the Minimum Norm failure sample, as utilized by Minimum Norm Importance Sampling (MNIS) [1]. To address the challenge of multiple failure regions, Hyperspherical Clustering and Sampling (HSCS) employs clustering techniques to identify these regions and assigns an OMSV to each, thereby constructing a mixture of Gaussian distributions [2]. To overcome the limitations of static distributions, Adaptive Importance Sampling (AIS) introduces a dynamically updated sampling distribution, significantly enhancing the accuracy of yield estimation [3]. Combining the strengths of HSCS and AIS, Adaptive Clustering and Sampling (ACS) further improves estimation efficiency by constructing a dynamically updated weighted mixture of Gaussian distributions [4]. Due to the complexity of high-dimensional parameter spaces, which cannot be adequately captured by specific distributions, Optimal Manifold Importance Sampling (OPTIMIS) employs a generative model known as normalizing flow to fit the optimal proposal distribution [5]. Despite their success, IS-based methods remain computationally expensive during sampling for estimation.

Another critical approach to enhancing yield estimation efficiency involves surrogate-based methods, which construct data-driven surrogate models to approximate circuit performance functions that typically require SPICE simulations, thereby significantly reducing simulation costs. In particular, [6] employs a Gaussian process (GP) to model the underlying performance functions and applies an entropy reduction strategy within an active learning framework. Building on this, Absolute Shrinkage Deep Kernel Learning (ASDK) replaces the GP with a nonlinear-correlated deep kernel method, which also includes feature selection to identify essential features for targeted analysis [7]. Additionally, [8] adopts a low-rank tensor approximation (LRTA) to efficiently approximate the performance function. Despite their successes, surrogate-based methods have not gained widespread acceptance due to their intrinsic instability and substantial data requirements. Additionally, surrogate-based methods are susceptible to challenges associated with highly nonlinear optimization during training. Without meticulous management, this can lead to flawed surrogate models and subsequently inaccurate yield estimations, which are intolerable in industry.

The strengths and weaknesses of IS-based and surrogate-based methods are both quite pronounced. Bayesian Optimized Importance Sampling (BOIS) combines surrogate models with IS, using the surrogate model to replace SPICE simulations, thereby reducing the simulation cost in the IS-based estimation process [9], [10]. However, as mentioned earlier, if the sur-

*Corresponding author.

978-3-9826741-0-0/DATE25/© 2025 EDAA

rogate model is inaccurate, it can lead directly to estimation errors. Recently, [11] proposes a two-stage meta-model IS based on the support vector machine (SVM) for rare event estimation. While effective in certain contexts, this method struggles to handle highly nonlinear problems.

In this paper, we present FUSIS, a unified framework that leverages the advantages of surrogate-based and IS-based methods to achieve accurate and efficient yield estimation. Unlike conventional surrogate-based approaches that directly replace SPICE simulations, FUSIS introduces Deep Kernel SVM as a surrogate indicator function. This, in combination with the known process variation distribution, enables the construction of a quasi-optimal proposal distribution for IS, significantly accelerating the convergence of the yield estimation. Furthermore, to address the potential bias introduced by inaccuracies in the SVM surrogate, we propose a novel correction factor that effectively calibrates the IS-based yield estimation. In summary, the novelty of this work includes:

1) **FUSIS**: A comprehensive framework that unifies the strengths of surrogate-based and IS-based methods to achieve improved performance in yield estimation.
2) **Deep Kernel SVM**: Employed as an indicator function approximation to effectively handle highly nonlinear problems.
3) **Quasi-optimal Proposal Distribution**: Constructed using SVM indicator function approximations, acting as the proposal distribution for IS to accelerate convergence.
4) **Novel Correction Factor**: Developed to calibrate the yield estimation bias introduced by potential inaccuracies in the SVM surrogate, ensuring estimation accuracy.
5) **Extensive Validation**: Demonstrated the superiority of FUSIS on multiple SRAM and analog circuits through comprehensive experiments, ablation studies, and robustness tests, showing up to a 24.84% improvement in accuracy (8.67% on average) and up to 29.54× (10.30× on average) speedup in efficiency when compared to seven state-of-the-art (SOTA) yield methods.

II. Background

A. Problem Definition

Consider the variation variables $\mathbf{x} = [x^{(1)}, x^{(2)}, \cdots, x^{(D)}]^T \in \mathcal{X}$, where $\mathcal{X}$ is the parameter space encompassing these variations. Generally, $\mathcal{X}$ is a high-dimensional space of dimension D, with each component of $\mathbf{x}$ representing a distinct manufacturing-related parameter that impacts a circuit, such as the length or width of PMOS and NMOS transistors. For the purposes of our analysis, we assume that the elements of $\mathbf{x}$ are statistically independent and follow a Gaussian distribution: $p(\mathbf{x}) = (2\pi)^{-\frac{D}{2}} \exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)$. Given $\mathbf{x}$, we can assess the performance of the circuit, denoted by $\mathbf{y}$ (e.g., metrics like memory read/write time and amplifier gain), through SPICE simulation. This relationship can be expressed as $\mathbf{y} = \mathbf{f}(\mathbf{x})$, where $\mathbf{f}(\cdot)$ denotes the SPICE simulator. A design is considered successful if $\mathbf{y}$ meets all predefined criteria $\mathbf{t}$ (e.g., $y^{(k)} \leq t^{(k)}$ for $k = 1, \cdots, K$); otherwise, it is regarded as a failure. To indicate failure, we define an indicator function $I(\mathbf{x})$, where $I(\mathbf{x})$ is 1 representing a failure design and 0 otherwise. The true failure rate P_f is then given by: $P_f = \int_{\mathcal{X}} I(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$.

B. Monte Carlo Yield Estimation

Directly calculating the yield is impractical due to the unknown function $I(\mathbf{x})$. One common method to estimate the failure rate is through MC simulation. This method involves sampling $\mathbf{x}_i$ from the distribution $p(\mathbf{x})$ and then computing the failure rate as the proportion of failures. Formally, this is expressed as: $\hat{P}_f = \frac{1}{N}\sum_{i=1}^{N} I(\mathbf{x}_i)$, where $\mathbf{x}_i$ represents the i-th sample from $p(\mathbf{x})$, and N is the total number of samples. To achieve an estimate with an accuracy of $1-\varepsilon$ and a confidence level of $1-\delta$, the required number of samples can be approximated by: $N \approx \frac{\log(1/\delta)}{\varepsilon^2 \hat{P}_f}$. For instance, to obtain an estimate with 90% accuracy ($\varepsilon = 0.1$) and 90% confidence ($\delta = 0.1$), the number of samples needed is roughly: $N \approx \frac{100}{\hat{P}_f}$. This sample size becomes impractically large for very small values of $\hat{P}_f$, such as 10^{-5}. Intuitively, this means that, on average, $1/\hat{P}_f$ samples are necessary to get one failure event.

C. Importance Sampling Yield Estimation

Unlike MC that samples directly from the distribution $p(\mathbf{x})$, IS-based methods use a proposal distribution $q(\mathbf{x})$ to draw samples and estimate the failure rate by

$$P_f = \int_{\mathcal{X}} \frac{I(\mathbf{x})p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} \approx \frac{1}{N}\sum_{i=1}^{N} \frac{I(\mathbf{x}_i)p(\mathbf{x}_i)}{q(\mathbf{x}_i)}, \tag{1}$$

where $\mathbf{x}_i$ are samples drawn from $q(\mathbf{x})$, and the integral is approximated as in MC. Equation (1) is proven to be more efficient than traditional MC, provided that the proposal distribution $q(\mathbf{x})$ is carefully chosen. According to [5], the optimal proposal distribution is given by

$$q^*(\mathbf{x}) = I(\mathbf{x})p(\mathbf{x})/P_f. \tag{2}$$

D. Support Vector Machine

SVM [12] is a supervised machine learning algorithm widely used for classification and regression. SVM finds the optimal hyperplane that maximally separates different classes in the feature space. Given a set of training samples $\{(\mathbf{x}_i, I(\mathbf{x}_i))\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $I(\mathbf{x}_i) \in \{0, 1\}$, the optimization problem is formulated as $\operatorname{argmin}_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$, subject to

$$I(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n, \tag{3}$$

where $\mathbf{w}$ is the normal vector to the hyperplane, b is the bias term, ξ_i are slack variables for handling non-separable cases, and C is a regularization parameter. To extend to non-linear problems, SVM employs kernels such as radial basis function (RBF) or sigmoid kernel to map the input features to higher-dimensional spaces and compute the correlation implicitly through kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$.

III. Proposed Approach

We unify surrogate-based and IS-based methods into a comprehensive framework, FUSIS, by leveraging their strengths. Unlike traditional surrogate models that directly replace SPICE simulations, the core of FUSIS is to use the surrogate model

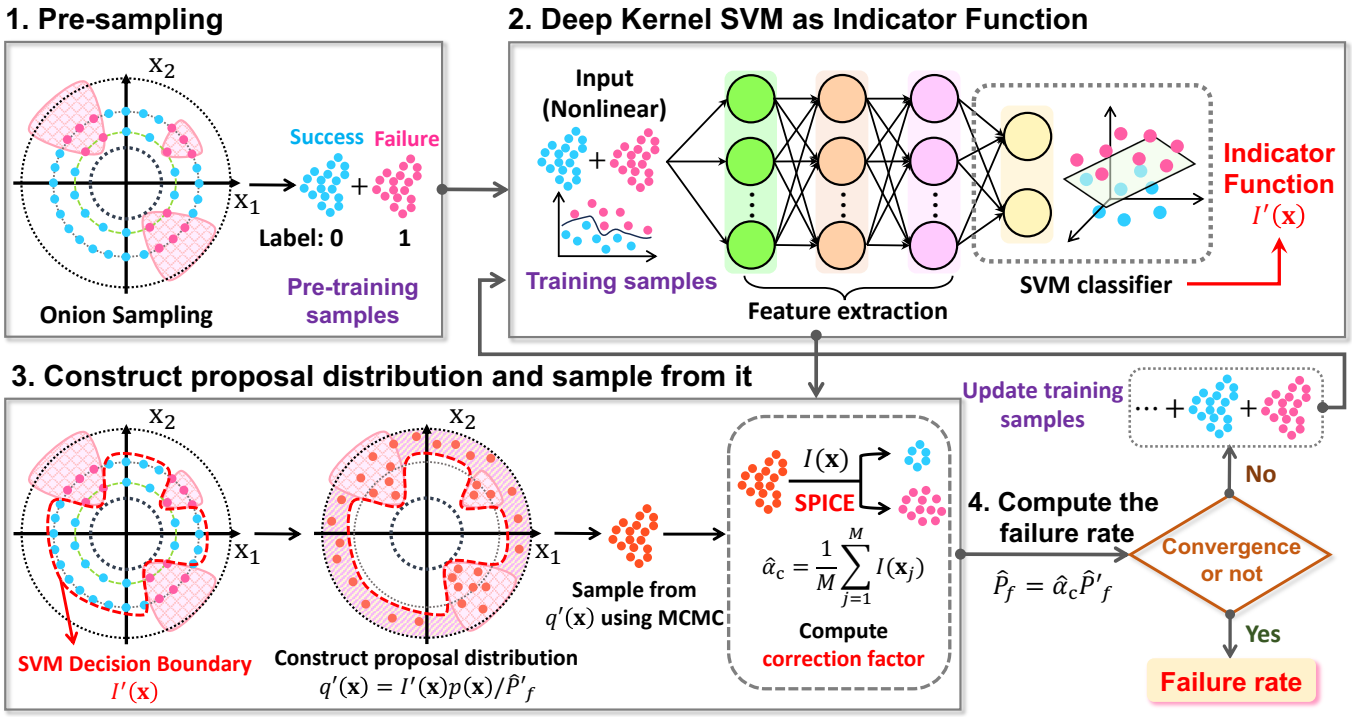

Fig. 1: The overall flow of FUSIS: (1) Perform pre-sampling using Onion Sampling [5] to extract pre-training samples (failure and non-failure samples.) (2) Train a Deep Kernel SVM as the indicator function $I'(\mathbf{x})$. (3) Construct the quasi-optimal proposal distribution $q'(\mathbf{x})$ using Eq. (7) and sample from it using MCMC to compute the correction factor. (4) Calculate the failure rate using Eq. (10). If convergence is not achieved, return to (2); otherwise, output the failure rate.

to assist in constructing the optimal proposal distribution for IS, accelerating convergence. Additionally, we introduce a correction factor to address the surrogate model's inaccuracies. The overall flow of FUSIS is shown in Fig. 1.

A. Deep Kernel SVM as Indicator Function Approximation

We first introduce the Deep Kernel SVM as the surrogate indicator function $I'(\mathbf{x})$, which combines the feature extraction capabilities of deep neural networks (DNNs) with the robust classification of SVMs, providing an effective approach for handling highly nonlinear problems. The Deep Kernel SVM leverages a DNN to extract complex features from input samples $\mathbf{x}_i \in \mathbb{R}^d$. The DNN maps these inputs to a higher-dimensional feature space through multiple layers of nonlinear transformations, denoted as $\phi_{NN}(\mathbf{x}_i)$. This transformation captures intricate data patterns essential for accurate yield estimation.

Once the features are extracted, the SVM finds the optimal hyperplane to separate failure and success cases in this transformed feature space. The decision function of the SVM, $F(\mathbf{x}) = \mathbf{w}\cdot\phi_{NN}(\mathbf{x})+b$. is used to define the surrogate indicator function $I'(\mathbf{x})$, which determines whether the circuit design has failed:

$$I'(\mathbf{x}) = \begin{cases} 1, & \text{if } F(\mathbf{x}) < 0 \\ 0, & \text{if } F(\mathbf{x}) \geq 0 \end{cases}. \tag{4}$$

B. Efficient Sampling From $q'(\mathbf{x})$ Using MCMC

With the indicator function $I'(\mathbf{x})$ derived from the Deep Kernel SVM, the failure rate can be estimated as:

$$P'_f = \int_{\mathcal{X}} I'(\mathbf{x})p(\mathbf{x})d\mathbf{x} \approx \frac{1}{N}\sum_{i=1}^{N} I'(\mathbf{x}_i). \tag{5}$$

However, due to estimation bias arising from the inaccuracy of the SVM surrogate (see Fig. 1), P'_f cannot serve as the final failure rate. IS-based estimation is known for its stability and convergence. Therefore, we use the SVM surrogate to assist in constructing a quasi-optimal proposal distribution $q'(\mathbf{x})$ for IS, as per Eq. (2), to accelerate the IS estimation:

$$q'(\mathbf{x}) = I'(\mathbf{x})p(\mathbf{x})/P'_f. \tag{6}$$

Due to the presence of P'_f in the denominator, direct sampling from $q'(\mathbf{x})$ is challenging. To address this, we use the Metropolis-Hastings algorithm [13], a widely used method in Markov Chain Monte Carlo (MCMC) [14] for generating a sequence of samples from a probability distribution that is difficult to sample directly. The algorithm begins with an initial sample $\mathbf{x}_0$ and generates candidate samples $\mathbf{x}'$ from a proposal distribution $g(\mathbf{x}|\mathbf{x}_t)$, typically modeled as a Gaussian distribution centered at the current sample: $g(\mathbf{x}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t, \sigma^2)$. The acceptance ratio α is computed as:

$$\alpha = \min\left(1, \frac{q'(\mathbf{x}')g(\mathbf{x}_t|\mathbf{x}')}{q'(\mathbf{x}_t)g(\mathbf{x}'|\mathbf{x}_t)}\right). \tag{7}$$

The candidate sample $\mathbf{x}'$ is accepted with probability α. If accepted, $\mathbf{x}_{t+1} = \mathbf{x}'$; otherwise, $\mathbf{x}_{t+1} = \mathbf{x}_t$. This process is iterated to generate a sequence of samples.

By using $I'(\mathbf{x})$, the construction of the quasi-optimal proposal distribution $q'(\mathbf{x})$ leverages the strengths of Deep Kernel SVM in handling complex, nonlinear distributions, thereby enhancing the efficiency and accuracy of failure rate estimation.

C. IS-based Yield Estimation Correction Factor

After obtaining the quasi-optimal proposal distribution $q'(\mathbf{x})$ by Eq. (7) and efficiently sampling from it, the failure rate

estimation can be rearranged as:

$$P_f = \int_{\mathcal{X}} \frac{I(\mathbf{x})p(\mathbf{x})}{q'(\mathbf{x})} q'(\mathbf{x})d\mathbf{x} = P'_f \int_{\mathcal{X}} \frac{I(\mathbf{x})}{I'(\mathbf{x})} q'(\mathbf{x})d\mathbf{x} = \alpha_c P'_f, \tag{8}$$

which reveals the connection between the golden standard P_f and the IS-based estimation P'_f. We introduce a correction factor α_c to calibrate the IS-based estimation by accounting for inaccuracies in $I'(\mathbf{x})$, the surrogate indicator function estimated by the SVM. The correction factor α_c can be conveniently approximated as:

$$\alpha_c = \int_{\mathcal{X}} \frac{I(\mathbf{x})}{I'(\mathbf{x})} q'(\mathbf{x})d\mathbf{x} \approx \frac{1}{M} \sum_{j=1}^{M} \frac{I(\mathbf{x}_j)}{I'(\mathbf{x}_j)}, \tag{9}$$

where $\{\mathbf{x}_j\}_{j=1}^M$ are samples drawn from $q'(\mathbf{x})$. Thus, $I'(\mathbf{x}_j) = 1$ for failure samples identified by the SVM, the correction factor $\hat{\alpha}_c$ simplifies to: $\hat{\alpha}_c = \frac{1}{M}\sum_{j=1}^{M} I(\mathbf{x}_j)$. Thus, incorporating the correction factor $\hat{\alpha}_c$, the estimated failure rate is calibrated:

$$\hat{P}_f = \hat{\alpha}_c \hat{P}'_f = \frac{1}{MN} \sum_{j=1}^{M} I(\mathbf{x}_j) \sum_{i=1}^{N} I'(\mathbf{x}_i). \tag{10}$$

In summary, FUSIS leverages the indicator function from the Deep Kernel SVM to construct a quasi-optimal proposal distribution for IS, enabling accurate identification of failure regions in the parameter space. By efficiently sampling from this distribution and applying a correction factor (its impact on the accuracy has been validated through ablation experiments in IV-E.), we significantly improve yield estimation accuracy. This approach not only mitigates the inaccuracies of the SVM surrogate but also enhances the overall efficiency of IS-based estimation. The full algorithm is summarized in Algorithm 1.

Algorithm 1 FUSIS Algorithm

Require: SPICE-based indicator function $I(\mathbf{x})$

1: Use Onion Sampling [5] to draw initial sample set $\mathcal{D} = \{(\mathbf{x}_i, I(\mathbf{x}_i))\}_{i=1}^n$
2: **repeat**
3: Update iteration $t = t + 1$
4: Use Deep Kernel SVM to approximate the indicator function $I'_t(\mathbf{x})$ and estimate $\hat{P}'^t_f$ according to Eq. (5)
5: Construct the quasi-optimal proposal distribution $q'_t(\mathbf{x})$ using Eq. (7)
6: Use Metropolis-Hastings sampling to draw M samples from $q'_t(\mathbf{x})$ and calculate the correction factor $\hat{\alpha}^t_c = \frac{1}{M}\sum_{j=1}^{M} I(\mathbf{x}_j)$.
7: Estimate the failure rate $\hat{P}_f = \frac{1}{t}\sum_{l=1}^{t} \hat{\alpha}^t_c \hat{P}'^t_f$
8: Update sample collection $\mathcal{D}$ with new samples
9: **until** Figure of Merit (FOM): $\frac{std(\hat{P}_f)}{\hat{P}_f} < 0.1$
10: **return** Failure rate $\hat{P}_f$

IV. Experimental Results

In this section, we thoroughly evaluate the accuracy and efficiency of our proposed method, FUSIS, for yield estimation on four benchmark circuits: a 6T-SRAM bit cell, an operational transconductance amplifier (OTA), a 6-bit 6T-SRAM array, and a 1093-dimensional SRAM column circuit. To ensure a comprehensive comparison, we implement seven SOTA methods as baselines: MNIS [1], HSCS [2], AIS [3], ACS [4], LRTA [8], ASDK [7], and OPTIMIS [5]. We use MC simulations as the gold standard for estimating the true failure rate. Additionally, we employ the Figure of Merit (FoM), denoted as ρ, which is calculated as $\rho = \frac{\mathrm{std}(\hat{P}_f)}{\hat{P}_f}$, where $\mathrm{std}(\hat{P}_f)$ represents the standard deviation of the estimated failure rate. Following the approach in [1], [2], we set $\rho = 0.1$ as the termination criterion for all methods. To assess performance, speedup is determined by $\frac{\#Sim_{MC}}{\#Sim}$, and the relative error rate is calculated as $(\hat{P}_f - \hat{P}_{f_{MC}})/\hat{P}_{f_{MC}}$.

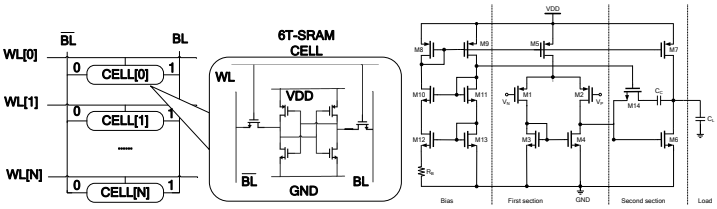

Fig. 2: Simplified schematic for 6T-SRAM and OTA Circuits.

In the experimental setup, for Deep Kernel SVM, we implement a feature extractor using a three-layer DNN. The structure consists of: the first layer expanding the input dimension by $4\times$, the second layer reducing it to $2\times$ the input dimension, and the third layer restoring it to the original dimension. Each layer uses ReLU activation [15], with a 50% Dropout applied after the first two layers to prevent overfitting. The feature extractor is trained for 1000 epochs, optimized using mean squared error (MSE) loss and the Adam optimizer [16]. After extracting features, we employ the SVM with a RBF kernel for classification, using GridSearchCV to optimize the regularization parameter C and the kernel parameter γ. For each method, we perform ten experiments using different random seeds, ensuring consistency in seed usage across all methods. The final failure rate is obtained by averaging the results of these ten experiments. Furthermore, we select the best-performing outcome from the ten random seed experiments for each method to visualize the iterative estimation of the failure rate and its FoM. Baseline methods are configured using their default settings, with hyperparameters fine-tuned where necessary to enhance performance. All experiments are conducted on a Windows system equipped with an AMD 7950X CPU and 32GB RAM.

A. 6T-SRAM Bit Cell

The 6T-SRAM bit cell, depicted on the left side of Fig. 2, is implemented using a 45nm CMOS process and comprises six transistors. Each transistor has three independent random variables: threshold voltage, mobility, and gate oxide thickness. These variables play a crucial role in affecting yield across all variation parameters, resulting in a total of 18 independent random variables for the circuit. Our experiments focus on the delay time of SRAM read/write operations as the key performance metric. The yield estimation results are summarized in Table I, while the failure rate convergence and FoM evaluation are illustrated in Fig. 3. We can see that FUSIS provides the most accurate estimation with the fewest simulations. In terms of accuracy, FUSIS achieves a relative error rate as low as

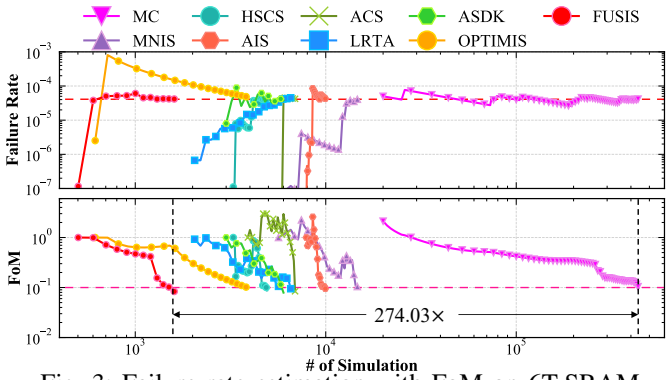

Fig. 3: Failure rate estimation with FoM on 6T-SRAM

TABLE I: Yield Estimation Results on 6T-SRAM

Model	Fail. Rate	Rel. Err.	# Sim	Speedup
MC	4.99e-5	-	406240	1×
MNIS	4.81e-5	3.61%	10030	40.50×
HSCS	4.86e-5	2.61%	4152	97.84×
AIS	4.85e-5	2.81%	9702	41.87×
ACS	4.70e-5	5.81%	9620	42.23×
LRTA	4.86e-5	2.61%	6130	66.27×
ASDK	4.85e-5	2.81%	6640	61.18×
OPTIMIS	4.93e-5	1.18%	3916	103.74×
FUSIS	**4.95e-5**	**0.80%**	**1602**	**253.58**×

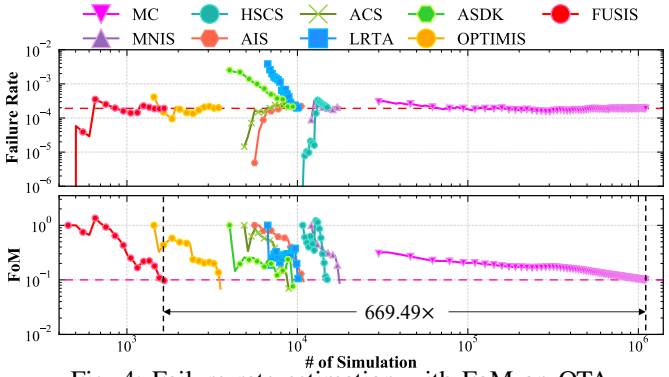

Fig. 4: Failure rate estimation with FoM on OTA

TABLE II: Yield Estimation Results on OTA

Model	Fail. Rate	Rel. Err.	# Sim	Speedup
MC	1.89e-4	-	1102000	1×
MNIS	1.64e-4	11.94%	21065	52.3×
HSCS	1.70e-4	10.15%	17950	61.39×
AIS	1.74e-4	8.18%	11178	98.59×
ACS	1.78e-4	5.59%	11053	99.70×
LRTA	2.04e-4	7.94%	10100	109.11×
ASDK	2.14e-4	11.68%	9600	114.79×
OPTIMIS	1.92e-4	1.57%	4126	267.09×
FUSIS	**1.90e-4**	**0.79%**	**1727**	**638.10**×

0.80%, improving by 0.38% to 5.01% over other baselines. For efficiency, FUSIS offers up to 253.58× speedup over MC, and 2.44× to 6.26× speedup over other baselines.

B. Operational Transconductance Amplifier

The OTA circuit, shown on the right side of Fig. 2, consists of 14 transistors. Each transistor has four process variation parameters: oxide thickness, threshold voltage, and deviations in length and width due to process variations. This results in a total of 56 independent random variables for the circuit. Our

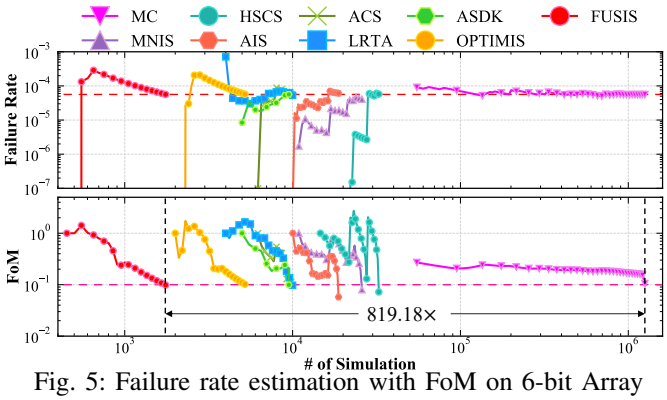

Fig. 5: Failure rate estimation with FoM on 6-bit Array

TABLE III: Yield Estimation Results on 6-bit Array

Model	Fail. Rate	Rel. Err.	# Sim	Speedup
MC	5.62e-5	-	1417500	1×
MNIS	4.94e-5	12.03%	45174	31.38×
HSCS	4.21e-5	25.09%	47090	30.10×
AIS	4.37e-5	22.19%	15996	88.62×
ACS	4.92e-5	12.44%	14060	100.82×
LRTA	5.96e-5	6.05%	12300	115.24×
ASDK	5.87e-5	4.44%	12500	113.40×
OPTIMIS	5.66e-5	0.71%	5300	267.45×
FUSIS	**5.61e-5**	**0.25%**	**1737**	**816.02**×

experiments focus on the quiescent current I_Q at $27°C$ as the performance metric. The yield estimation results are presented in Table II, and the progression of failure rate convergence along with the FoM evaluation is depicted in Fig. 4.

The results show that FUSIS consistently delivers highly accurate estimations with fewer simulations for the analog circuit. In terms of accuracy, FUSIS achieves a relative error rate as low as 0.79%, improving by 0.78% to 11.15% compared to baselines. In efficiency, FUSIS reaches up to 638.10× speedup over MC and 2.39× to 12.20× over other baselines. These findings underscore the robustness of FUSIS across different circuit complexities.

C. SRAM Array Circuit

Building on the successful validation of FUSIS in the 6T-SRAM bit cell experiments, we extend its application to two complex SRAM array circuits: a 6-bit 6T-SRAM array and a 1093-dimensional SRAM column circuit.

1) 6-bit 6T-SRAM Array: This circuit consists of six 6T-SRAM bit cells and includes 108 variational parameters, accounting for the effects of peripheral circuits to enhance the accuracy of failure rate estimation. The results are shown in Table III, with the failure rate convergence and FoM evaluation illustrated in Fig. 5. FUSIS continues to perform exceptionally well on the higher-dimensional circuit. In terms of accuracy, FUSIS achieves a relative error rate as low as 0.25%, improving accuracy by 0.46% to 24.84% over baselines. In terms of efficiency, FUSIS provides up to 816.02× speedup over MC, and 3.05× to 27.11× speedup over other baselines. These results demonstrate its effectiveness in handling the complexity of advanced SRAM architectures.

2) 1093-Dimensional SRAM Column: We further increase the complexity of the problem by incorporating a detailed BSIM4 model, accounting for 1093 variation parameters. The

TABLE IV: Yield Estimation Results on 1093-D Column

Model	Fail. Rate	Rel. Err.	# Sim	Speedup
MC	4.80e-5	-	1650000	1×
MNIS	4.21e-5	12.32%	81000	20.37×
HSCS	1.53e-6	N/A	-	-
AIS	3.79e-5	21.04%	27900	59.14×
ACS	4.30e-5	10.42%	24000	68.75×
LRTA	3.33e-5	N/A	-	-
ASDK	N/C	-	-	-
OPTIMIS	4.95e-5	3.13%	4832	341.47×
FUSIS	**4.73e-5**	**1.46%**	**2737**	**602.85×**

"N/C" stands for "Not Converged." And "N/A" stands for "Not Applicable," meaning that the relative error rate of this method exceeded 30%, making it unable to provide an accurate estimation.

results are presented in Table IV. HSCS, LRTA, and ASDK are not suitable for such high-dimensional circuits, as they may lead to significant estimation errors or even convergence failures. In contrast, FUSIS is highly effective in handling high-dimensional circuits, achieving a relative error rate as low as 1.46%. Moreover, it offers a speedup improvement of 1.77× to 29.54× in efficiency compared to other baselines.

TABLE V: The Comparison of Computational Time

CPU Hours	MNIS	HSCS	AIS	ACS	LRTA	ASDK	OPT.	FUSIS
6T-SRAM	4.8	2.0	4.6	4.6	3.1	3.2	2.0	**0.8**
OTA	50.5	43.0	26.8	26.5	24.3	25.0	10.0	**4.4**
6-bit Array	270.9	283.2	95.9	84.3	73.9	75.7	32.4	**10.6**

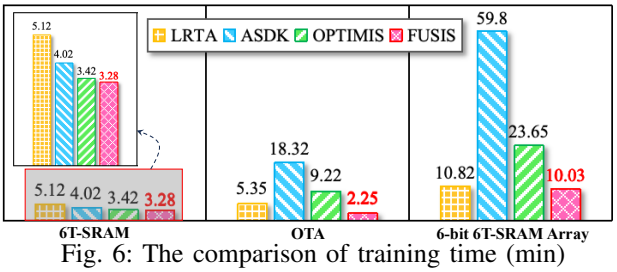

Fig. 6: The comparison of training time (min)

D. Computational Time Study

The computational time comparison for the experiments is shown in Table V, where FUSIS demonstrates a significant efficiency advantage, with average speedups of 4.34×, 6.69×, and 12.35× for the 6T-SRAM, OTA, and 6-bit 6T-SRAM array circuits, respectively. Additionally, the training time of FUSIS is compared to three SOTA methods (LRTA, ASDK, and OPTIMIS), which involve training machine learning models. As shown in Fig. 6, FUSIS achieves average speedups of 1.28×, 4.87×, and 3.13× across the three circuits. These results highlight the superior computational and training efficiency of FUSIS.

E. Ablation Study on the Impact of the Correction Factor

We conduct ablation experiments on all benchmark circuits to assess the impact of the correction factor on the accuracy of failure rate estimation, as shown in Table VI (the second column is the true failure rate). Fig. 7 shows a comparison of the convergence accuracy during the estimation iterations of FUSIS, with and without the correction factor. From Table VI and Fig. 7, it is evident that without the correction factor—relying solely on the SVM surrogate model—accurate estimations

TABLE VI: Ablation Study on the Correction Factor

With correction		**No**		**Yes**	
Metric	True	Fail. Rate	Rel. Err.	Fail. Rate	Rel. Err.
6T-SRAM	4.99e-5	5.92e-4	N/A	4.95e-5	**0.80%**
OTA	1.89e-4	2.87e-2	N/A	1.90e-4	**0.79%**
6-bit Array	5.62e-5	2.70e-4	N/A	5.61e-5	**0.25%**
1093-D Column	4.80e-5	3.52e-4	N/A	4.73e-5	**1.46%**

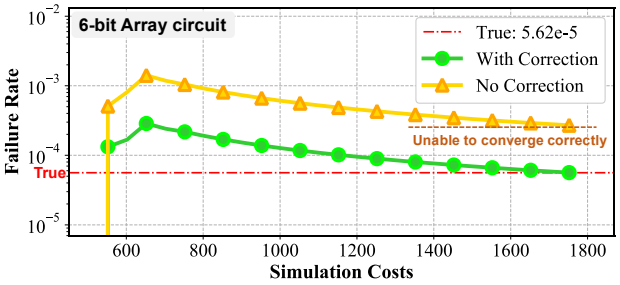

Fig. 7: Ablation study of FUSIS in estimation iterations

are not achieved. However, incorporating the correction factor significantly improves accuracy, with an average relative error of just 0.61%. This demonstrates that the correction factor effectively mitigates the estimation bias caused by surrogate model inaccuracies.

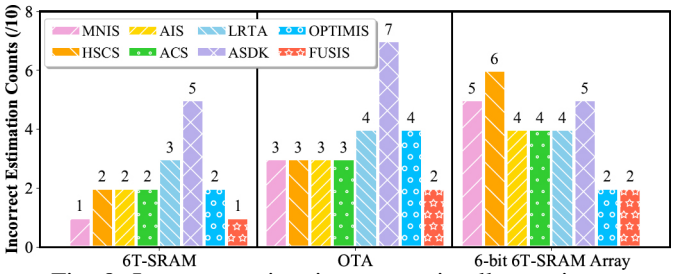

Fig. 8: Incorrect estimation counts in all experiments

F. Robustness Study

To address the industry's key concern of robustness, we conduct a comprehensive study on three benchmark circuits for all methods. Each method is tested with the same set of ten random seeds, and the number of incorrect estimations—defined as those with a relative error rate exceeding 30%—is recorded in Fig. 8. The results show that FUSIS consistently demonstrates superior stability across all circuits. In the 6T-SRAM experiment, both FUSIS and MNIS fail only once out of ten runs, while in the OTA and 6-bit 6T-SRAM array circuits, FUSIS fails only twice. This underscores the effectiveness of SVM surrogate calibration.

V. Conclusion

In this paper, we propose FUSIS, a unified framework that combines surrogate-based and IS-based methods for accurate and efficient yield estimation. By introducing Deep Kernel SVM and a novel correction factor, FUSIS accelerates convergence and mitigates estimation bias, delivering significant improvements in both accuracy and efficiency compared to conventional methods. Extensive experiments on multiple benchmark circuits demonstrate FUSIS's superior performance in yield estimation over seven SOTA methods, with notable gains in both accuracy and computational efficiency.

REFERENCES

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

[2] W. Wu, S. Bodapati, and L. He, "Hyperspherical clustering and sampling for rare event analysis with multiple failure region coverage," in *Proceedings of the 2016 on International Symposium on Physical Design*, ser. ISPD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 153–160. [Online]. Available: https://doi.org/10.1145/2872334.2872360

[3] X. Shi, F. Liu, J. Yang, and L. He, "A fast and robust failure analysis of memory circuits using adaptive importance sampling method," in *Proceedings of the 55th Annual Design Automation Conference*, ser. DAC '18, 2018. [Online]. Available: https://doi.org/10.1145/3195970.3195972

[4] X. Shi, H. Yan, J. Wang, X. Xu, F. Liu, L. Shi, and L. He, "Adaptive clustering and sampling for high-dimensional and multi-failure-region sram yield analysis," in *Proceedings of the 2019 International Symposium on Physical Design*, ser. ISPD '19, 2019, p. 139–146. [Online]. Available: https://doi.org/10.1145/3299902.3309748

[5] Y. Liu, G. Dai, and W. W. Xing, "Seeking the yield barrier: High-dimensional sram evaluation through optimal manifold," in *2023 60th ACM/IEEE Design Automation Conference (DAC)*, 2023, pp. 1–6.

[6] S. Yin, X. Jin, L. Shi, K. Wang, and W. W. Xing, "Efficient bayesian yield analysis and optimization with active learning," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, ser. DAC '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 1195–1200. [Online]. Available: https://doi.org/10.1145/3489517.3530607

[7] S. Yin, G. Dai, and W. W. Xing, "High-dimensional yield estimation using shrinkage deep features and maximization of integral entropy reduction," in *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, ser. ASPDAC '23, New York, NY, USA, 2023, p. 283–289. [Online]. Available: https://doi.org/10.1145/3566097.3567907

[8] X. Shi, H. Yan, Q. Huang, J. Zhang, L. Shi, and L. He, "Meta-model based high-dimensional yield analysis using low-rank tensor approximation," in *Proceedings of the 56th Annual Design Automation Conference 2019*, ser. DAC '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3316781.3317863

[9] D. D. Weller, M. Hefenbrock, M. S. Golanbari, M. Beigl, and M. B. Tahoori, "Bayesian optimized importance sampling for high sigma failure rate estimation," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019, pp. 1667–1672.

[10] M. Hefenbrock, D. D. Weller, M. Beigl, and M. B. Tahoori, "Fast and accurate high-sigma failure rate estimation through extended bayesian optimized importance sampling," in *2020 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2020, pp. 103–108.

[11] C. Ling and Z. Lu, "Support vector machine-based importance sampling for rare event estimation," *Structural and Multidisciplinary Optimization*, vol. 63, pp. 1609–1631, 2021.

[12] S. Amari and S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Neural Networks*, vol. 12, no. 6, pp. 783–789, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608099000325

[13] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *The american statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[14] P. Dellaportas and G. O. Roberts, "An introduction to mcmc," in *Spatial statistics and computational methods*. Springer, 2003, pp. 1–41.

[15] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 315–323. [Online]. Available: https://proceedings.mlr.press/v15/glorot11a.html

[16] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.