CIS: Conditional Importance Sampling for Yield Optimization of Analog and SRAM Circuits

Yanfang Liu[1], Wei W. Xing[2*]
[1]*School of Integrated Circuit Science and Engineering, Beihang University, Beijing, China*
[2]*School of Mathematics and Statistics, University of Sheffield, S3 7RH, UK*
liuyanfang@buaa.edu.cn, wayne.xingle@gmail.com

2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC) | 979-8-3503-9354-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ASP-DAC58780.2024.10473819

***Abstract*—Yield optimization is one of the central challenges in submicrometer integrated circuit manufacture. Classic yield optimization methods rely on importance sampling (IS) to provide efficient and robust yield estimation for each individual design. Despite its success, such an approach is still computationally expensive due to the large number of calculations for many different designs. To resolve this challenge, we propose conditional importance sampling (CIS) that can approximate the optimal proposal distribution for any given design by leveraging the power of the modern deep-learning-based sampling method, conditional normalizing flow. More importantly, CIS generalizes well to unseen design and thus can deliver effective yield optimization with a small number of expensive simulations. To conduct yield optimization efficiently with consideration of creditable uncertainty, we propose a novel Important Sampling Bayesian Optimization (ISBO) using a deep-warped gradient-boosting regression (GBR). The proposed method is extensively evaluated against five state-of-the-art baselines; the results show that the proposed method delivers superior performance: a speedup of 1.10x-10.46x (4.45x on average) with even higher yield designs, an improvement of 1.1x-10x (4.44x on average) in consideration of the Optimality-Cost Ratio, and most importantly, excellent robustness and consistency in all our extensive experiments on analog and SRAM circuits.**

***Index Terms*—Yield Estimation, Yield Optimization, Importance Sampling, Conditional Normalizing Flow**

I. Introduction

With the advancements in integrated circuits technology, microelectronic devices have scaled down to the nanometer range, introducing significant process variance, e.g., doping fluctuation, intra-die mismatches, and threshold voltage variation. Consequently, circuit performance often deviates from nominal design and fails to meet design specifications, particularly in analog and mixed-signal CMOS circuits [1], [2].To address this issue, the design of robust nominal circuits becomes crucial. More specifically, we aim to satisfy electronic specifications while accommodating fabrication process variations, thereby framing the yield optimization problem. Yield optimization is highly challenging due to the need for a large number of simulations to estimate the yield for just a given design. Moreover, the availability of yield derivatives with respect to the design parameters is limited, further complicating the optimization process. An accurate and efficient yield estimation is crucial for successful yield optimization, and it has received considerable attention in the literature. The Monte Carlo (MC) method is the golden standard for yield estimation, known for its reliability and wide practical use. However, MC is highly computationally expensive, demanding tens of thousands of circuit simulations to achieve reasonable accuracy. For instance, achieving a yield of 99.9999% necessitates a minimum of 10^7 simulations, which is impractical in real-world scenarios.

Importance sampling (IS) based methods aim to enhance the efficiency of MC estimator by emphasizing sampling from the failure region. With the foundation established by the optimal mean shift vector (OMSV [3]), IS-based methods have emerged as a significant branch for yield estimation, due to their high efficiency, reliability, and robustness. In an effort to further enhance OMSV, adaptive importance sampling (AIS) is proposed in [4] which dynamically updates the shifted distribution as more samples are collected. To handle high-dimensional spaces, adaptive clustering sampling (ACS [5]) employs multi-cone clustering to sample from multiple regions and sequentially update the proposal distribution. Additionally, non-Gaussian adaptive importance sampling (NGAIS [6]) introduces a mixture of von Mises-Fisher distributions to replace Gaussian distribution, further improving AIS's performance.

Another important branch of yield estimation methods is surrogate-based yield estimation which employs a surrogate/meta-model to predict performance metrics based on variational parameters and design parameters. To inherit the powerful fitting capacity of polynomial chaos expansion (PCE) but to scale to the high-dimensional problems in yield optimization, [7] introduces a low-rank tensor approximation to efficiently approximate complete PCE to deliver an accurate prediction of the performance metrics.

Since our objective is yield optimization rather than precise yield estimation, allocating excessive computational resources for accurate estimation in designs with inherently low yields is not sensible. Inspired by this idea, [8] proposes a heuristic two-stage MC yield estimation and Bayesian optimization (BO) method based on weighted expected improvement for yield optimization (WEIBO). The framework is further enhanced by [9], which replaces the weighted acquisition function with a max-value entropy search that enables a more effective exploration of the design space (MESBO). Additionally, [10] combines a gradient-free optimizer with its yield estimated using a kernel density estimator to achieve efficient yield optimization (KDEBO). Despite their success, the separation of yield estimation and optimization entails a loss of accuracy and stability.

To resolve this issue, all sensitivity adversarial importance sampling (ASAIS [11]) eliminates the need for surrogate-based optimization by directly optimizing the OMSV using sensitivity analysis. However, this approach makes a strong assumption that the OMSV is enough to characterize the failure pattern, which is not the case in practice. In contrast, bayesian yield analysis and optimization with active learning

This work is supported by Fundamental Research Funds for the Central Universities; experiments are supported by Primarius Technologies Co.,Ltd.
* Corresponding author.

(BYA [12]) proposes a Gaussian process (GP)-based method that conducts yield estimation and optimization simultaneously to deliver both efficiency and effectiveness. However, like any GP-based estimation, this method has the risk of failure if the surrogate fitting is inaccurate.

To conduct an efficient yield optimization, we aim to equip the IS-based method with a design parameter-dependent proposal distribution such that the advantages of using a surrogate can be absorbed while the risk of poor fitting is mitigated. The novelty of this work includes:

1) As far as the authors are aware, CIS is the first IS-based yield estimation that leverages prior knowledge from similar designs as in the surrogate-based yield estimation methods to improve sampling efficiency significantly.
2) CIS is developed with an extensive study of the latest machine learning sampling methods.
3) To conduct effective yield optimization with creditable uncertainty, we propose a novel ISBO approach equipped with deep-warped gradient-boosting regression (GBR).
4) Based on our extensive experiments of different circuits, CIS achieves remarkable optimization in almost all cases consistently, with an average speedup of 4.45x (up to 10.46x), optimal performance improvement of 313x (up to 4,156x), and normalized Optimality-Cost Ratio (nOCR) improvement of 4.44x (up to 10x) compared with the state-of-the-art (SOTA) yield optimization methods.

II. Background

A. Problem Definition

Let $\mathbf{x} = [x^{(1)}, x^{(2)}, \cdots, x^{(d_x)}]^T \in \mathcal{X}$ represent a vector encompassing various design parameters, e.g., transistor widths and lengths, resistance values, capacitance values, and bias voltages and currents. The feasible design parameter space, denoted as $\mathcal{X}$, is defined by the circuit designer. The variational parameters, denoted as $\mathbf{v} = [v^{(1)}, v^{(2)}, \cdots, v^{(d_v)}]^T \in \mathcal{V}$, are considered to comprehensively capture the inherent random variations in the manufacturing process. After normalization, the variational parameters $\mathbf{v}$ are assumed to follow an independent Gaussian distribution, specifically characterized by the probability density function (PDF) $p(v^{(i)}) = \exp\left(-{v^{(i)}}^2/2\right)/\sqrt{2\pi}$. The qualified design refers to a circuit with the corresponding parameters $[\mathbf{x}, \mathbf{v}]$, and the circuit performance metric $\mathbf{y}$ can be expressed as a function $\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{v})$. The failure status of a circuit is denoted using the failure indicator $I(\mathbf{x}, \mathbf{v})$, where $I(\mathbf{x}, \mathbf{v})$ is 1 representing a failure design and 0 otherwise. Thus, the failure rate is

$$g(\mathbf{x}) = \int_{\mathcal{V}} I(\mathbf{x}, \mathbf{v}) p(\mathbf{v}) d\mathbf{v}, \tag{1}$$

The optimization problem is defined as finding $\mathbf{x}^*$ that minimizes the failure rate, i.e., $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} g(\mathbf{x})$. The challenge here is twofold. First, the computation of the failure rate $g(\mathbf{x})$ requires a large number of simulations to evaluate the integral Eq. (1), and second, the derivative $\nabla_{\mathbf{x}} g(\mathbf{x})$ is not available.

B. Monte Carlo and Importance Sampling Yield Estimation

Yield estimation, essentially the computation of $g(\mathbf{x})$, is typically performed using MC methods. This involves sampling M instances of $\mathbf{v}_i$ from the distribution $p(\mathbf{v})$ and evaluating the failure probability by calculating the ratio of failure samples to the total number of samples. The estimated failure rate $\hat{g}(\mathbf{x})$ can be approximated as $\hat{g}(\mathbf{x}) \approx \frac{1}{M}\sum_{i=1}^{M} I(\mathbf{x}, \mathbf{v}_i)$. Obtaining an estimate with $1-\varepsilon$ accuracy and $1-\delta$ confidence requires approximately $N \approx \frac{\log(1/\delta)}{\varepsilon^2 \hat{g}(\mathbf{x})}$ samples. For instance, achieving 90% accuracy ($\varepsilon = 0.1$) and 90% confidence ($\delta = 0.1$) would require around $N \approx 100/g(\mathbf{x})$ samples. Consequently, MC is infeasible in practice for small values of $g(\mathbf{x})$, such as 10^{-8} (which is a common requirement in SRAM circuits). Intuitively, this can be understood by considering that, on average, $1/g(\mathbf{x})$ samples are needed to get one failure sample.

Instead of sampling from the distribution $p(\mathbf{v})$, IS-based methods draw samples from a proposal distribution $q(\mathbf{v}|\mathbf{x})$. The circuit failure rate $g(\mathbf{x})$ can be estimated using IS as follows:

$$g(\mathbf{x}) = \int_V \frac{I(\mathbf{x}, \mathbf{v}) p(\mathbf{v})}{q(\mathbf{v}|\mathbf{x})} q(\mathbf{v}|\mathbf{x}) d\mathbf{v} \approx \frac{1}{N}\sum_{i=1}^{N} \frac{I(\mathbf{x}, \mathbf{v}_i) p(\mathbf{v}_i)}{q(\mathbf{v}_i|\mathbf{x})}, \tag{2}$$

where $\mathbf{v}_i$ is the sample drawn from $q(\mathbf{v}|\mathbf{x})$. By choosing an appropriate proposal distribution $q(\mathbf{v}|\mathbf{x})$, the variance of the estimator can be significantly reduced, meaning that fewer samples are needed to achieve the same level of accuracy.

C. Yield Optimization Using Bayesian Optimization

Yield optimization is a more challenging task as it requires optimization of the unknown function of $g(\mathbf{x})$, which can be evaluated only at given design $\mathbf{x}$, and the direct derivative $\nabla_{\mathbf{x}} g(\mathbf{x})$ is unavailable. To conduct yield optimization, most previous work relies on surrogate-based optimization. BO is commonly employed to optimize the unknown function $g(\mathbf{x})$. BO assumes a GP prior $\mathbf{g}(\mathbf{x})|\boldsymbol{\theta} \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'|\boldsymbol{\theta}))$, where $m(\mathbf{x})$ denotes the mean function and $k(\mathbf{x}, \mathbf{x}'|\boldsymbol{\theta})$ represents the covariance function. The model parameters $\boldsymbol{\theta}$ of the GP are estimated using the maximum likelihood estimate (MLE) of the likelihood function, which is a joint Gaussian distribution based on several failure rate estimations of $g(\mathbf{x})$. $g(\mathbf{x})$ is then approximated with mean $\bar{g}(\mathbf{x})$ and variance $\hat{v}(\mathbf{x})$ for any given $\mathbf{x}$. This gives BO a unique advantage in optimizing $g(\mathbf{x})$ by considering the uncertainty. A simple optimization strategy is to propose the next candidate based on design $\mathbf{x}$ that minimizes $\bar{g}(\mathbf{x}) - \beta\hat{v}(\mathbf{x})$ where β plays a crucial role in balancing the trade-off between exploration and exploitation. This approach is commonly referred to as the upper confidence bound (UCB), known for its simplicity and effectiveness. Alternative acquisition functions, e.g., predictive entropy search, are also available for BO.

III. Research Methodology

A. Adaptive Important Sampling Yield Estimation

For the IS-based yield estimation, the key component is to have a proper proposal distribution, which is only available to be approximated when enough data is collected. Thus, almost adaptive important sampling methods try to design proposal distribution that asymptotically approximates the ideal one, which can be derived from Equation Eq. (2) by minimizing the approximate variance

$$q(\mathbf{v}|\mathbf{x}) = \operatorname*{argmin}_{q} \mathbb{E}_q \left[w^2(\mathbf{v}|\mathbf{x}) \left(I(\mathbf{x}, \mathbf{v}) - g(\mathbf{x}) \right)^2 \right], \tag{3}$$

where $w(\mathbf{v}|\mathbf{x}) = p(\mathbf{v})/q(\mathbf{v}|\mathbf{x})$ represents the $\mathbf{x}$ dependent importance weight. By applying the Lagrange multiplier rule for the calculus of variations, we obtain the optimal proposal distribution, $q(\mathbf{v}|\mathbf{x}) = \frac{p(\mathbf{v})I(\mathbf{x},\mathbf{v})}{g(\mathbf{x})}$, which is proportional to the distribution of failure events. Because $I(\mathbf{x}, \mathbf{v})$ is unknown, we need to give $q(\mathbf{v}|\mathbf{x})$ some assumed form and update it as more data, i.e., simulations, are executed.

For example, OMSV [3], the ground-breaking work for IS-based yield estimation, designs a simple proposal distribution $q(\mathbf{v}|\mathbf{x}) = p(\mathbf{v}|\boldsymbol{\mu}^*, \mathbf{1})$, where $\boldsymbol{\mu}^*$ is updated by $\boldsymbol{\mu}^* = \text{argmin}_i \, ||\mathbf{v}_i||^2 \quad \text{s.t. } I(\mathbf{x}, \mathbf{v}_i) = 1$, where $i = 1, \cdots, N_e$ are all existing failure samples for $\mathbf{x}$. The proposal distribution is updated using the existing failure samples closest to the failure origin. To enhance OMSV, AIS [4] proposes a Gaussian mixture form: $q(\mathbf{v}|\mathbf{x}) = \frac{1}{N_s}\sum_{i=1}^{N_s} p_i(\mathbf{v}|\boldsymbol{\mu}_i, \Sigma_i)$, where $\boldsymbol{\mu}_i = \{\mathbf{v}_i\}_{i=1}^{N_s}$ represents N_s failure samples. To put forth a model with a larger capacity, ACS [5] introduces $q(\mathbf{v}|\mathbf{x})$ with clusters: $q(\mathbf{v}|\mathbf{x}) = \frac{1}{\sum_{i=1}^{M}\lambda_i}\sum_{j=1}^{J}(\sum_{\mathbf{v}_i \in C_j}\lambda_i)h_j(\mathbf{v})$, where i and j represent the failure samples and clusters, respectively. The failure samples $\{\mathbf{x}, \mathbf{v}_i\}_{i=1}^{M}$ are assigned to clusters $\{C_j\}_{j=1}^{J}$; N_j is the number of failure samples of cluster C_j satisfying $\sum_{j=1}^{J} N_j = M$. And $h_j(\mathbf{v})$ represents the cluster sampling distribution in each cluster, which is a mixture of Gaussian distribution, and λ_i represents the probability density for all failure samples. ACS generates M samples from $q(\mathbf{v}|\mathbf{x})$ and re-cluster them to update the $q(\mathbf{v}|\mathbf{x})$. A similar work is NGAIS [6], which uses a non-Gaussian distribution: von Mises-Fisher (vMF) mixture distribution as the proposal distribution. The proposal distribution is $q(\mathbf{v}|\mathbf{x}) = \sum_{l=1}^{L}\alpha_l v_l(\mathbf{v} \mid \mu_l, \kappa_l)$, where L is the number of vMF distributions and α_l is the corresponding normalized weight function. $v_l(\mathbf{v}|\mu_l, \kappa_l)$ represents a single-modal vMF distribution where μ_l is the unit mean direction vector and the concentration parameter k_l is a measure of the degree of directional dispersion. NGAIS adopts the MLE based on the Expectation-Maximization (EM) algorithm framework to conduct parameter estimation. New samples are generated from the current $q(\mathbf{v}|\mathbf{x})$ and validated using real simulations, which then update $q(\mathbf{v}|\mathbf{x})$ with more knowledge.

B. Modern Deep Learning Sampling

Despite the success of these SOTA methods, all of them are limited by the lack of flexibility (for different problems) and the requirement of domain knowledge to design a proper form. To harness the power of modern deep learning to approximate the optimal proposal distribution $q^*(\mathbf{v}|\mathbf{x})$, we conduct an extensive study, covering Generative Adversarial Network (GAN), Denoising Diffusion Probabilistic Models (DDPM), Variational Autoencoder (VAE) and normalizing flow (NF) to find the best candidate. GAN approximates the data distribution indirectly with a sampling procedure and an adversarial training; DDPM uses the diffusion process to transform simple data to complex distribution of particularly images. VAE maximizes the evidence lower bound and uses the idea of dimension reduction to approximate the target data distribution. All these methods have demonstrated their great potential for sampling from unknown and complex distributions. However, they do not produce a direct estimation of data probability density, which is essential when computing the importance weight in IS-based yield. In contrast, NF can explicitly model the PDF of the data through invertible transformations, enabling efficient sampling along with explicit calculation of probability, making it particularly suitable for IS-based yield estimation [13].

The idea of NF is to model a distribution $q(\mathbf{v})$ as a change of variables from the base distribution $p(\mathbf{z})$ (e.g., Gaussian distribution) using a series of invertible transformations which are parameterized f_k: $\mathbf{v} = f_K \circ \ldots \circ f_2 \circ f_1(\mathbf{z}, \eta)$, where η is the model parameters. The PDF of the distribution $q(\mathbf{v})$ can be calculated using the change of variables formula, which involves taking the determinant of the Jacobian matrix of each transformation: $q(\mathbf{v}) = p(\mathbf{z})\prod_{k=1}^{K} |\det(\frac{\partial f_k(\mathbf{z};\eta)}{\partial \mathbf{z}})|$.

C. Conditional Normalizing Flow Yield Estimation

In this work, our main focus is on the yield optimization, which requires yield estimation for multiple $\mathbf{x}$ iteratively. Thus, we modify NF with a Conditional parameter $\mathbf{x}$ such that the connection between different designs can be considered. We specify proposal distribution $q(\mathbf{v}|\mathbf{x})$ as a Conditional Normalizing Flow (CNF) [14]. The difference from NF is that CNF transforms samples from a simple base distribution $p(\mathbf{z})$ with design parameter $\mathbf{x}$,

$$q(\mathbf{v}|\mathbf{x}) = p(\mathbf{z})\prod_{k=1}^{K} |\det(\frac{\partial f_k(\mathbf{z};\eta,\mathbf{x})}{\partial \mathbf{z}})|. \tag{4}$$

To train the CNF, we aim to maximize the log-likelihood, which can be computed by summing the log-probability densities from each transformation as follows: $\log q(\mathbf{v}|\mathbf{x}) = \log p(\mathbf{z}) + \sum_{k=1}^{K} \log \left|\frac{\partial f_k(\mathbf{z};\eta,\mathbf{x})}{\partial \mathbf{z}}\right|$. Through the application of invertible transformations utilizing Spline Coupling Flows conditioned on $\mathbf{x}$, CNF can effectively learn intricate conditional optimal proposal distribution $q(\mathbf{v}|\mathbf{x})$ as more data are collected, which are based on sampling from $q(\mathbf{v}|\mathbf{x})$ and putting the samples through the simulations. The difference is that the design $\mathbf{x}$ is now considered in a uniform model like in the joint surrogate-based method [12].

D. Uncertainty Quantification for CNF Yield Estimation

Traditional BO relies on GP approximation of $g(\mathbf{x})$ with approximated yield estimation samples $\hat{g}(\mathbf{x}_i)$. Despite that GP can quantify the uncertainty of $g(\mathbf{x})$, the uncertainty quantification is for the lack of collocation points $\hat{g}(\mathbf{x}_i)$, whereas the essence uncertainty caused by the CNF is not well considered.

To resolve this issue, we first evaluate the uncertainty of $\hat{g}(\mathbf{x})$ caused by the proposal distribution CNF using the deep ensemble method. More specifically, the yield estimation is based on R different CNFs, which are trained in parallel based on random initializations to produce mean estimation $\bar{g}(\mathbf{x}_i) = \frac{1}{R}\sum_{r=1}^{R}\hat{g}_r(\mathbf{x}_i)$ and variance $\hat{v}(\mathbf{x}_i) = (\frac{1}{R}\sum_{r=1}^{R}(\hat{g}_r(\mathbf{x}_i) - \bar{g}(\mathbf{x}_i))^2)^{1/2}$, where $\hat{g}_r(\mathbf{x}_i)$ is yield estimation based on rth CNFs for $\mathbf{x}_i$.

E. Important Sampling Bayesian Optimization

With the uncertainty of $\hat{g}(\mathbf{x}_i)$, we can now perform yield optimization based on BO. More specifically, Since any BO eventually optimizes a deterministic acquisition function, we propose to directly optimize the acquisition function e.g., $UCB(\mathbf{x}) = \bar{g}(\mathbf{x}) - \beta\hat{v}(\mathbf{x})$, using a deterministic yet powerful model such as GBR. This way, the uncertainty is considered, and some errors are supposed to be canceled out. We call this important sampling Bayesian optimization (ISBO). GBR is

a highly effective and extensively utilized machine learning technique for complex regression tasks, e.g., finance and healthcare, where domain knowledge cannot be easily utilized. It works by iteratively constructing multiple weak learners and combining them with weights to approximate the UCB function using colocation samples $UCB(\mathbf{x}_i)$

$$UCB(\mathbf{x}) \approx \sum_i \gamma_i \cdot \mathbf{f}_{DT}(\mathbf{x}, \varepsilon_i), \tag{5}$$

where $\mathbf{f}_{DT}(\mathbf{x}, \varepsilon_i)$ is a decision tree regression with parameters ε_i; γ_i is the weight. For efficient optimization of the approximated $UCB(\mathbf{x})$, we adopt a gradient-free global optimization approach. To further improve practicality and avoid local optimum, we propose Dynamic Optimization with Restricted Neighborhoods (DORN). DORN first divides the design parameter space $\mathcal{X}$ into several equal subspaces and executes a random search optimization algorithm within each subspace to select the optimal subspace and abandon the other subspaces. This process is repeated until the maximum number of iterations is reached. Compared to gradient-based optimization algorithms, DORN avoids falling into local optima and is characterized by its simplicity and efficiency. The main steps of CIS yield optimization are summarized in Algorithm 1.

When conducting the yield optimization, the geometry of $UCB(\mathbf{x})$ can impact the optimization significantly. A smoother geometry is the key to reducing this influence. Thus, we leverage input-warping functions to stretch the geometry locally to deliver a smoother response surface that is easier to optimize. The idea is to use the Beta CDF input warping function [15]

$$w(x^{(i)}) = 1 - (1 - (x^{(i)})^a)^b \tag{6}$$

to parameterize $x^{(i)}$. Here, $a > 0$ and $b > 0$ represent the concentration parameters learned during the training process. Unlike the original work, we discover that multiple layers of wrapping can further improve the final performance possibly due to its capacity to smooth more local regions. Some experimental findings are shown in the later experiment section.

Algorithm 1 CIS Yield Optimization

Require: SPICE-based Indication $I(\mathbf{v}, \mathbf{x})$, R, N_{iter}
1: Pre-sampling: get failure samples $\mathcal{D} = \{\{\mathbf{v}_s, \mathbf{x}_k\}_{s=1}^{N_\mathbf{v}}\}_{k=1}^{N_\mathbf{x}}$ by Onion Sampling [13]
2: Initialization: obtain initial training samples $\mathcal{D}_g$ for GBR using deep ensemble with CNF IS-based yield estimation
3: **for** i = 1 to N_{iter} **do**
4: Fit GBR with $\mathcal{D}_g$ and optimize UCB using DORN to get optimal design $\mathbf{x}_i^*$
5: **for** r=1 to R **do**
6: Update CNF $q_r(\mathbf{v}|\mathbf{x}_i^*)$ with dataset $\mathcal{D}$ and estimate $\hat{g}_r(\mathbf{x}_i^*)$
7: Generate Q samples $\mathbf{v}_s'$ from $q_r(\mathbf{v}|\mathbf{x}_i^*)$.
8: Pass $\mathbf{v}_s'$ to SPICE-based indication $I(\mathbf{v}_s', \mathbf{x}_i^*)$ to get failure samples $\mathcal{D}' = \{\mathbf{v}_s', \mathbf{x}_i^*\}_{s=1}^{N_\mathbf{v}'}$ and update dataset $\mathcal{D} = \mathcal{D} \cup \mathcal{D}'$
9: **end for**
10: Compute $\bar{g}(\mathbf{x}_i^*)$ and $\hat{v}(\mathbf{x}_i^*)$ to Get $UCB(\mathbf{x}_i^*)$
11: Get $\mathcal{D}_g' = \{\mathbf{x}_i^*, UCB(\mathbf{x}_i^*)\}$ and update $\mathcal{D}_g = \mathcal{D}_g \cup \mathcal{D}_g'$
12: **end for**
13: **return** $\mathbf{x}_i^*$

IV. Experimental Results

In this work, we comprehensively evaluate the accuracy and efficiency of our proposed method, named *CIS*, on three circuits: an operational transconductance amplifier (OTA), a 6T-SRAM, and an adder circuit. To establish a solid basis for comparison, we implement five SOTA yield optimization methods: WEIBO [2], MESBO [9], ASAIS [11], KDEBO [10], and BYA [12]. To ensure the robustness of the methods, we introduce two distinct circuit specifications, referred to as Case 1 and Case 2, for each circuit in the yield optimization experiments. We validate the optimal designs using MC simulations, with 4e7 and 1e6 simulations conducted in Case 1 and Case 2, respectively. In order to better demonstrate the optimization efficiency of each method, we introduce a novel evaluation criterion called nOCR. OCR is defined as the ratio of final yield divided by the number of simulations. Normalization means dividing all OCR by a baseline OCR, which is the OCR of CIS in this work. To ensure fairness, each algorithm is executed 5 times with different seeds, and the final optimization experimental results are based on the average of the results of these 5 seeds as shown in Table I.

In the experiments, CNF adopts an 8-layer flow of transformations, with each layer containing a 2-layer multi-layer perceptron (MLP), which has 10 times the dimension of $\mathbf{v}$ hidden units and uses the ReLU activation function. The optimization processes use the Adam optimizer, with 500 iterations for CNF model update and UCB optimization. When fitting GBR, we use 3 iterations, each proposing 10 samples to update CNF and get $\hat{v}(\mathbf{x})$ and $\bar{g}(\mathbf{x})$. In addition, 3-layer input warping function is applied. The baseline methods are implemented using their default settings, and in some cases where certain methods lack generalization, we fine-tune hyperparameters to achieve better performance for different circuits. Note that CIS does not require any adjustments for any experiments. All experiments are conducted on a Linux system with an AMD 7950x CPU and 32GB RAM.

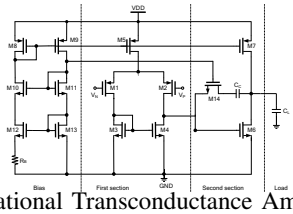

Fig. 1: Operational Transconductance Amplifier Circuit

A. Operational Transconductance Amplifier Circuit

The operational transconductance amplifier (OTA) circuit (shown in Fig. 1) is fabricated using a 180 nm CMOS process with 14 transistors. Three design parameters are contained: the transistor widths of M5, M7, and M13. Additionally, each transistor has four process variation parameters, namely oxide thickness, threshold voltage, and length and width variations due to process deviation. In our experiments, we evaluate the performance of interest, namely, the quiescent current I_Q at a temperature of $27°C$.

As shown in Table I, in Case 1, CIS achieves a significant optimization performance improvement of 0.01%-3.3% with a speedup of 2.94x-5.86x compared to all baseline methods, which shows the stability and robustness of CIS. In case 2, MESBO outperforms CIS in terms of standard deviation (std) results, but it incurs 3.73x higher simulations than CIS. In contrast, with a speedup of 2.71x-6.11x, CIS demonstrates

TABLE I: Yield optimization report for the OTA, 6T-SRAM and adder circuit

	Circuits	OTA				6T-SRAM				Adder			
Case	Method	Yield	Std	#Sim	nOCR	Fail Rate	Std	#Sim	nOCR	Fail Rate	Std	#Sim	nOCR
1	WEIBO	99.56%	0.39%	5377	0.25x	2.42e-6	1.78e-6	2559	0.42x	1.30e-6	4.20e-7	4033	0.39x
	MESBO	99.63%	0.45%	4691	0.29x	8.50e-8	6.25e-8	4586	0.23x	7.00e-8	4.00e-8	8018	0.19x
	ASAIS	96.67%	4.04%	2651	0.50x	6.50e-8	9.00e-8	1192	0.89x	5.00e-8	2.24e-8	2422	0.64x
	KDEBO	99.96%	0.09%	8000	0.17x	1.87e-4	2.26e-4	7000	0.15x	6.00e-8	2.00e-8	10000	0.16x
	BYA	99.94%	0.06%	6600	0.21x	7.00e-8	4.36e-8	11000	0.10x	4.50e-8	**1.00e-8**	11000	0.14x
	Proposed	**99.97%**	**0.02%**	**1365**	**1.00x**	**4.50e-8**	**1.00e-8**	**1065**	**1.00x**	**4.00e-8**	1.22e-8	**1555**	**1.00x**
2	WEIBO	99.78%	0.13%	4223	0.31x	1.42e-4	1.25e-4	1897	0.40x	1.78e-5	7.48e-7	3515	0.39x
	MESBO	99.84%	**4.00e-3%**	4880	0.27x	2.62e-5	3.95e-5	2252	0.34x	1.72e-5	2.48e-6	6150	0.22x
	ASAIS	99.43%	0.22%	3546	0.37x	7.80e-6	8.93e-6	804	0.95x	1.72e-5	5.71e-6	2413	0.57x
	KDEBO	99.69%	0.12%	8000	0.16x	1.10e-2	2.39e-3	5720	0.13x	3.94e-5	4.65e-5	8000	0.17x
	BYA	99.83%	7.48e-3%	6600	0.20x	1.08e-5	4.12e-6	8000	0.10x	1.72e-5	**2.45e-7**	8000	0.17x
	Proposed	**99.87%**	0.01%	**1310**	**1.00x**	**5.60e-6**	**2.80e-6**	**765**	**1.00x**	**1.36e-5**	5.28e-6	**1383**	**1.00x**

a notable performance improvement of 0.04%-0.90%. More importantly, in both Case 1 and Case 2, CIS performs best in the nOCR metric, which achieves an average improvement of 4.10x (up to 6.25x) compared to all baseline methods.

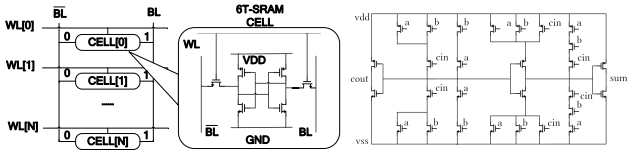

Fig. 2: Simplified schematic for 6T-SRAM and Adder Circuits

B. 6T-SRAM Circuit

The SRAM bit-cell (see the left figure of Fig. 2 for a simplified schematic representation) is fabricated using a 45nm CMOS process and consists of six transistors. Each transistor has three independent random variables: threshold voltage, mobility, and gate oxide width, composing 18 independent variational parameters for each SRAM cell. For optimization, we focus on the width and length of an individual tran sistor as the design parameters. In our experiments, the performance metric of interest is the optimization of the write delay.

In Table I, CIS demonstrates the best performance with the fewest simulations in both Case 1 and Case 2. Compared to the baseline methods, CIS stands out with a remarkable performance improvement of 1.56x-4,156x with a speedup of 1.12x-10.32x in Case 1 and a performance improvement of 1.39x-1,964x with a speedup of 1.05x-10.46x in Case 2. Furthermore, CIS outperforms all baseline methods with a perfect nOCR metric in both case1 and case2 experiments, exhibiting an average improvement of 4.87 x (up to 10x).

C. Adder Circuit

The adder circuit (shown in the right figure of Fig. 2) consists of 28 MOS transistors, each with three consistent variational parameters, totaling 84 variational parameters. Our main design focus involves two design parameters: the width and length of each transistor. We meticulously evaluate the time-to-threshold (TT) performance within a defined temperature range of $27°C$ while determining the yield by simulating the transient response until the sum of outputs matches a predefined threshold voltage.

In this evaluation, BYA shows favorable std performance, but it incurs simulations up to 7x higher than CIS. Remarkably, CIS achieves the best performance with the minimum number of simulations. In Case 1, CIS achieves an impressive performance improvement of 1.13x-32.50x, with a speedup of 1.56x-7.07x. In Case 2, CIS achieves a substantial performance improvement of 1.26x-2.90x, along with a speedup of 1.74x-5.78x. Additionally, CIS achieves an excellent nOCR metric in both Case 1 and Case 2 compared to all baseline methods, showcasing an average improvement of 4.34x (up to 7.14x).

D. Resource-based Comparison

To showcase the optimization efficiency of each method in our experiments, we initially conduct ten optimization runs with a unified stopping criterion. However, since each method has distinct stopping criteria, we aim to provide a more comprehensive demonstration of optimization performance. To achieve this, we adopt a classic experimental approach: each method is no longer restricted to a fixed number of runs but instead given the same simulation resources for yield optimization. We conduct Case 1 experiments on the previously mentioned three circuits and record the optimization outcomes for each method at 5,000 and 10,000 simulations. Additionally, we perform 5 runs for each method with different random seeds. The experimental results are summarized in Table II.

For the OTA circuit, when using 5,000 simulations, CIS achieves the best optimization performance, outperforming other baseline methods by 0.03%-3.32% in terms of mean results. With 10,000 simulations, CIS continues to perform remarkably well, except for ASAIS, which has a smaller std result. The optimization improvement of CIS ranges from 0.01% to 0.4%. For the classic 6T-SRAM circuit, CIS stands out as the best performer in all cases, achieving a remarkable performance improvement of 1.07x-269,143x with 5,000 simulations and 1.01x-6,233x with 10,000 simulations. Lastly, for the adder circuit, CIS remains the top-performing method among all methods. It demonstrates a remarkable optimization improvement of 1.11x-63,000x with 5,000 simulations, and a 1.40x-50x enhancement with 10,000 simulations.

E. ISBO vs BO(GP)

Furthermore, we conduct a comparative experiment in Case 1 for OTA circuits using ISBO and BO with GP, respectively, and the results are shown in Table III. Due to the relatively high dimensionality of OTA circuits and the sparsity of samples in high-dimensional space, the kernel function estimation of GP becomes unstable, resulting in significantly inferior

TABLE II: Yield optimization for the three circuits with different numbers of total simulations

	Circuits	OTA (Yield)				6T-SRAM (Fail Rate)				Adder (Fail Rate)			
#Sim	Method	Best	Worst	Mean	Std	Best	Worst	Mean	Std	Best	Worst	Mean	Std
5K	WEIBO	99.94%	99.09%	99.56%	0.40%	3.50e-7	5.75e-6	2.42e-6	1.78e-6	7.50e-7	2.02e-6	1.30e-6	4.20e-7
	MESBO	**99.99%**	99.07%	99.63%	0.45%	**2.50e-8**	2.00e-7	8.50e-8	6.25e-8	5.00e-8	1.24e-3	2.48e-4	4.89e-4
	ASAIS	99.97%	91.65%	96.65%	4.06%	**2.50e-8**	**5.00e-8**	3.75e-8	1.25e-8	**2.50e-8**	**5.00e-8**	4.44e-8	1.04e-8
	KDEBO	**99.99%**	99.50%	99.80%	0.22%	**2.50e-8**	9.24e-2	9.42e-3	2.72e-2	5.00e-8	1.87e-2	2.52e-3	5.64e-3
	BYA	99.98%	99.87%	99.93%	0.06%	**2.50e-8**	1.50e-7	1.40e-7	5.83e-8	**2.50e-8**	**5.00e-8**	4.50e-8	1.00e-8
	Proposed	**99.99%**	**99.94%**	**99.97%**	**0.02%**	**2.50e-8**	**5.00e-8**	**3.50e-8**	**1.22e-8**	**2.50e-8**	**5.00e-8**	**4.00e-8**	**1.22e-8**
10K	WEIBO	99.94%	99.09%	99.58%	0.39%	**2.50e-8**	5.75e-6	1.22e-6	1.71e-6	5.00e-8	2.00e-6	1.58e-6	7.61e-7
	MESBO	**99.99%**	99.93%	99.97%	0.02%	**2.50e-8**	**5.00e-8**	3.03e-8	1.02e-8	5.00e-8	1.50e-7	7.00e-8	4.00e-8
	ASAIS	99.98%	99.96%	99.97%	**7.05e-3%**	**2.50e-8**	**5.00e-8**	3.33e-8	1.18e-8	**2.50e-8**	**5.00e-8**	4.44e-8	1.04e-8
	KDEBO	**99.99%**	99.92%	99.96%	0.09%	**2.50e-8**	5.19e-4	1.87e-4	2.26e-4	5.00e-8	1.00e-7	6.00e-8	2.00e-8
	BYA	**99.99%**	99.87%	99.94%	0.06%	**2.50e-8**	1.50e-7	7.00e-8	4.36e-8	**2.50e-8**	**5.00e-8**	4.50e-8	1.00e-8
	Proposed	**99.99%**	**99.97%**	**99.98%**	7.48e-3%	**2.50e-8**	**5.00e-8**	**3.00e-8**	**1.00e-8**	**2.50e-8**	**5.00e-8**	**3.17e-8**	**9.72e-9**

TABLE III: Yield Optimization for OTA

Model	Best	Worst	Mean	Std
ISBO	**99.99%**	**99.94%**	**99.97%**	**0.02%**
BO(GP)	98.31%	98.19%	98.25%	0.04%

TABLE IV: Yield Optimization Statistical Results

	OTA		6T-SRAM		Adder	
Layer	Single	Multiple	Single	Multiple	Single	Multiple
Best	98.95%	**99.99%**	5.00e-8	**2.50e-8**	5.00e-8	**2.50e-8**
Worst	98.81%	**99.94%**	2.00e-6	**5.00e-8**	5.27e-4	**5.00e-8**
Mean	98.89%	**99.97%**	1.02e-6	**4.50e-8**	1.06e-4	**4.00e-8**
Std	0.05%	**0.02%**	8.72e-7	**1.00e-8**	2.10e-4	**1.22e-8**

optimization performance compared to GBR, which is more suitable for fitting in high-dimensional space. Specifically, the optimization performance of ISBO for yield optimization demonstrates an average improvement of 1.72% compared with BO.

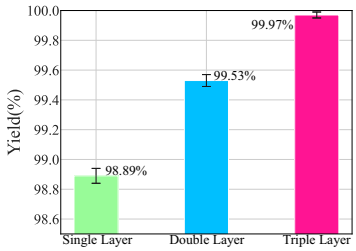

Fig. 3: CIS with different Warping layers on OTA circuit

F. Ablation Study

Finally, we also test the impact of stacking warping function layers, both single and multiple, on the optimization performance of CIS. The yield optimization experiments are conducted on three circuits in Case 1 and the results are summarized in Table IV (yield for OTA and fail rate for others). We can see that stacking multiple layers of warping functions consistently outperforms using a single layer. This improvement is 1.08% for the OTA circuit and 441x-2,650x for 6T-SRAM and adder circuits, based on the mean results. As shown in Fig. 3, the more layers, the better the optimization.

V. Conclusion

We propose CIS, a novel yield optimization framework equipped with CNF and a novel ISBO approach. The exceptional performance of CIS is firmly validated through a comprehensive series of experiments on real-world circuit benchmarks, accompanied by meticulous ablation studies. The limitation includes the discrete optimization for GBR in ISBO and its scalability to dimension, which should be addressed with other advanced optimization schemes with gradients.

References

[1] B. Liu, F. V. Fernández, and G. G. Gielen, "Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational intelligence techniques," IEEE TCAD, vol. 30, no. 6, pp. 793–805, 2011.

[2] M. Wang, W. Lv, F. Yang, C. Yan, W. Cai, D. Zhou, and X. Zeng, "Efficient yield optimization for analog and sram circuits via gaussian process regression and adaptive yield estimation," IEEE TCAD, vol. 37, no. 10, pp. 1929–1942, 2017.

[3] L. Dolecek, M. Qazi, D. Shah, and A. Chandrakasan, "Breaking the simulation barrier: Sram evaluation through norm minimization," in Proc. ICCAD. IEEE, 2008, pp. 322–329.

[4] X. Shi, F. Liu, J. Yang, and L. He, "A fast and robust failure analysis of memory circuits using adaptive importance sampling method," in Proc. DAC. IEEE, 2018, pp. 1–6.

[5] 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 Proc. ISPD, 2019, pp. 139–146.

[6] X. Shi, H. Yan, C. Li, J. Chen, L. Shi, and L. He, "A non-gaussian adaptive importance sampling method for high-dimensional and multi-failure-region yield analysis," in Proc. ICCAD, 2020, pp. 1–8.

[7] 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 Proc. DAC, 2019, pp. 1–6.

[8] M. Wang, F. Yang, C. Yan, X. Zeng, and X. Hu, "Efficient bayesian yield optimization approach for analog and sram circuits," in Proc. DAC. IEEE, 2017, pp. 1–6.

[9] S. Zhang, F. Yang, D. Zhou, and X. Zeng, "Bayesian methods for the yield optimization of analog and sram circuits," in Proc. ASP-DAC.

[10] D. D. Weller, M. Hefenbrock, M. Beigl, and M. B. Tahoori, "Fast and efficient high-sigma yield analysis and optimization using kernel density estimation on a bayesian optimized failure rate model," IEEE TCAD, vol. 41, no. 3, pp. 695–708, 2022.

[11] W. Hu, Z. Wang, S. Yin, Z. Ye, and Y. Wang, "Sensitivity importance sampling yield analysis and optimization for high sigma failure rate estimation," in Proc. DAC, 2021, pp. 895–900.

[12] S. Yin, X. Jin, L. Shi, K. Wang, and W. W. Xing, "Efficient bayesian yield analysis and optimization with active learning," in Proc. DAC, 2022, pp. 1195–1200.

[13] Y. Liu, G. Dai, and W. W. Xing, "Seeking the yield barrier: High-dimensional sram evaluation through optimal manifold," 2023. [Online]. Available: https://arxiv.org/abs/2307.15773

[14] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, "Neural spline flows," Advances in neural information processing systems, vol. 32, 2019.

[15] J. Snoek, K. Swersky, R. Zemel, and R. Adams, "Input warping for bayesian optimization of non-stationary functions," in Proc. IMLS. PMLR, 2014, pp. 1674–1682.