OPT: Optimal Proposal Transfer for Efficient Yield Optimization for Analog and SRAM Circuits

Yanfang Liu[1], Guohao Dai[2], Yuanqing Cheng[1], Wang Kang[1], and Wei W. Xing[1*]
[1] *School of Integrated Circuit Science and Engineering, Beihang University, Beijing, China*
[2] *College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen, China*
{liuyanfang,yuanqing,wang.kang,wxing} @buaa.edu.cn, daiguohao2019@email.szu.edu.cn

2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD) | 979-8-3503-2225-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCAD57390.2023.10323689

***Abstract*—Yield optimization is one of the central challenges in submicrometer integrated circuit manufacture. However, yield optimization is computationally expensive due to intensive yield estimation and intractable optimization processes. In this work, we first reinvent the state-of-the-art all sensitivity adversarial importance sampling (ASAIS) yield optimization from a Laplace approximation perspective, which also reveals its limitations and suggests improvements. We then generalize it with infinite components and discover the key ingredient in yield optimization to be an effective proposal distribution transfer (OPT) procedure, which is captured using conditional normalizing flow (CNF). To deliver a reliable yield optimization pipeline that accounts for the uncertainty due to the lack of data, we propose sequential ensemble, the first empirical uncertainty estimation that enables tractable Bayesian yield optimization without introducing an extra surrogate for the first time. We conduct extensive experiments against five state-of-the-art baselines and show that the proposed method delivers superior performance: a speedup of 1.01x-11.94x (5.57x on average) with higher yield designs, and most importantly, excellent robustness and consistency in all our experiments on analog and SRAM circuits.**

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

I. Introduction

As the technology of integrated circuits develops, microelectronic devices shrink their scale to nano-meter, which leads to severe process variance, e.g., doping fluctuation, intra-die mismatches, and threshold voltage variation. This will cause the performance of the circuits to deviate from the nominal design and even fail to meet the specifications, especially in the fields of analog and mixed-signal CMOS circuits [1], [2]. It is thus crucial to design nominal circuits that not only satisfy electronic specifications but are also robust against fabrication process variations, which forms the yield optimization problem. Yield optimization is challenging because it requires a large number of simulations to estimate the yield of a given design, and also the derivative of the yield w.r.t. the design parameters is not available.

A successful yield optimization generally requires an accurate and efficient yield estimation as an inner loop, which has been extensively studied in the literature. The golden standard for yield estimation is the Monte Carlo (MC) method, which is still widely used in practice due to its reliability. Nonetheless, MC is extremely inefficient as it requires tens of thousands of circuit simulations to achieve reasonable accuracy. For instance, a yield of 99.9999% requires a minimum of 10^7 simulations, which is infeasible in practice.

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

Importance sampling (IS) based methods aim to reduce the variance of the MC estimator by sampling more from the failure regions. With the foundation laid by optimal mean shift vector (OMSV [3]), IS-based methods have become an important branch for yield estimation, due to their high efficiency and, most importantly, reliability and robustness. To further improve OMSV, [4] proposes an adaptive importance sampling (AIS) to update the shifted distribution as more samples are collected. To deal with high-dimensional space, adaptive clustering sampling (ACS [5]) samples from multiple regions clustered by multi-cone clustering and sequentially updates its proposal distribution. AIS is further enhanced by [6] by introducing a mixture of von Mises-Fisher distributions to replace the standard normal distribution.

Another important branch of yield estimation methods is surrogate-based yield estimation, which builds a surrogate/meta model to predict the performance metric given any variational and design parameters. [7] puts forth a low-rank tensor approximation to the polynomial chaos expansion (PCE) to approximate the performance function. [8] instead uses the Gaussian process (GP) with features selection to deal with high-dimensional problems. Based on GP, [9] proposes an entropy reduction active learning for efficiency improvement.

With an efficient yield estimation in hand, one can now optimize the yield with gradient-free optimization such as Bayesian optimization (BO). Keep in mind that our goal is yield optimization instead of accurate yield estimation. It might not be wise to spend too much effort (computational budget) on an accurate yield estimation for an obviously low-yield design. Such a philosophy is also shared by [10], which proposes a heuristic two-stage MC yield estimation and BO for yield optimization (WEIBO). This framework is further improved by [11] by replacing the weighted acquisition function with a max-value entropy search to better explore the design parameters space (MESBO). In [12], the min-norm failure vector (MNFV) is proposed to optimize the yield by increasing the overall distance from the failure boundary based on IS-based yield estimation. [13] combines a gradient-free optimizer routine with its yield estimated on a kernel density estimator and BO (KDEBO).

Despite their efficiency and success, a combination of IS-based yield estimation and BO for yield optimization is ad-hoc and requires careful tuning of many intermediate hyperparameters, making it less attractive outside the research community. To address this issue, one can remove either component. All sensitivity adversarial importance sampling (ASAIS [14]) removes BO in the pipeline by directly optimizing the OMSV, obtained from OMSV yield estimation, using sensitivity analy-

sis. Due to its simplicity and reliability of IS, ASAIS is highly efficient and robust. The major limitation is that it can be inaccurate (see Fig. 1) due to the simple assumption embedded in OMSV. Other limitations include the lack of full knowledge sharing between similar designs and the large variance of the optimization process due to the lack of consideration of uncertainty.

These limitations are addressed by Bayesian yield analysis (BYA [9]), which removes the yield estimation and directly incorporates the design and variational parameters into a GP to predict performance metrics. The yield optimization and yield estimation are unified into a single framework and conducted jointly to achieve maximum efficiency with active learning. However, as the no free lunch theorem suggests, the surrogate model itself can become computationally expensive; also, the yield optimization quality can be compromised if the surrogate is not accurate enough, leading to misleading optimal design.

To bridge the gap between IS and surrogate methods, we propose a novel optimal proposal transfer (OPT) framework by 1) revealing the connection between IS and surrogate methods under a novel statistical perspective, 2) introducing conditional normalizing flow (CNF) based OPT, and 3) equipping OPT with BO. The novelty of this work is as follows,

1) We derive a novel statistical perspective for IS-based yield optimization, which reveals the limitations of the ASAIS and a link between IS and surrogate methods.
2) We propose a CNF-based OPT. It combines the advantages of IS and surrogate methods and provides efficient and robust yield optimization with effective knowledge transfer, which is validated through a joint yield estimation experiment on real circuits.
3) We propose sequential ensemble, the first empirical uncertainty quantification tailored for yield problem. Combined with OPT, it enables efficient Bayesian yield optimization without the need for an extra surrogate for the first time.
4) Based on our extensive experiments of different circuits, OPT achieves remarkable optimization in almost all cases consistently, with an average speedup of 5.57x (up to 11.94x) and optimal yield improvement of 705x (up to 6,367x) compared with the state-of-the-art (SOTA) yield optimization methods.

II. Background

A. Problem Definition

Let $\mathbf{x} = [x_1, x_2, \cdots, x_{d_x}]^T \in \mathcal{X}$ denote a vector containing all the design parameters, e.g., transistor widths and lengths, resistance values, capacitance values, and bias voltages and currents, whereas $\mathcal{X}$ indicates the feasible design parameters space with bounds specified by the circuit designers. The inevitable random variations of a manufacturing process are assumed fully captured by the variational parameters, denoted as $\mathbf{v} = [v_1, v_2, \cdots, v_{d_v}]^T \in \mathcal{V}$. After normalization, $\mathbf{v}$ is considered independent Gaussian distributed, i.e., $p(v_i) = \exp\left(-v_i^2/2\right)/\sqrt{2\pi}$. The circuit performance metrics $\mathbf{y}$ can be considered as a function $\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{v})$. When all metrics can meet the predefined criteria $\mathbf{y}^0$, e.g., $\mathbf{y} \leq \mathbf{y}^0$, the circuit with the corresponding parameters $[\mathbf{x}, \mathbf{v}]$ is considered a qualified design. For a certain design $\mathbf{x}$, the circuit yield $g(\mathbf{x})$ is defined,

$$g(\mathbf{x}) \triangleq \int_V I(\mathbf{x}, \mathbf{v}) p(\mathbf{v}) d\mathbf{v}, \tag{1}$$

where $I : \mathcal{X} \times \mathcal{V} \to \{0, 1\}$ is the indicator function of whether all performance metrics fail the predefined criteria. The yield optimization problem is then formulated as

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}}\, g(\mathbf{x}), \tag{2}$$

where $\mathbf{x}^*$ is the optimal design subject to the random variations $\mathbf{v}$. Yield optimization is equivalent to failure rate minimization, which we use in this work, i.e., $g(\mathbf{x})$ is the failure rate and $I(\mathbf{x}, \mathbf{v}) = 1$ if the design fails. 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

Estimation of $g(\mathbf{x})$ is known as yield estimation, which is commonly achieved using MC. It samples M $\mathbf{v}_i$ from $p(\mathbf{v})$ and evaluates the failure rate by the ratio of failure samples to total samples, $\hat{g}(\mathbf{x}) \approx \frac{1}{M}\sum_{i=1}^{M} I(\mathbf{x}, \mathbf{v}_i)$. To obtain an estimate of $1-\varepsilon$ accuracy with $1-\delta$ confidence, $N \approx \frac{\log(1/\delta)}{\varepsilon^2 \hat{g}(\mathbf{x})}$ is required. For a modest 90% accuracy ($\varepsilon = 0.1$) with 90% confidence ($\delta = 0.1$), we need $N \approx 100/g(\mathbf{x})$ samples, which is infeasible in practice for a small $g(\mathbf{x})$, says, 10^{-8}. We can also see this intuitively from the fact that it requires $1/g(\mathbf{x})$ samples on average just to observe a failure event.

Instead of drawing samples from $p(\mathbf{v})$, the IS methods draw samples from a proposal distribution $q(\mathbf{v})$ and estimate

$$g(\mathbf{x}) = \int_V \frac{I(\mathbf{x}, \mathbf{v}) p(\mathbf{v})}{q(\mathbf{v})} q(\mathbf{v}) d\mathbf{v} \approx \frac{1}{N}\sum_{i=1}^{N} \frac{I(\mathbf{x}, \mathbf{v}_i) p(\mathbf{v}_i)}{q(\mathbf{v}_i)}, \tag{3}$$

where $\mathbf{v}_i$ are samples drawn from $q(\mathbf{v})$. If $q(\mathbf{v})$ is chosen properly, the variance of the estimator can be reduced significantly, i.e., fewer samples are required to achieve the same accuracy.

One of the fundamental works in IS for yield estimation is OMSV [3], which shifts the mean of original distribution $p(\mathbf{v})$ to the smallest passing sample $\boldsymbol{\mu}^*$,

$$\boldsymbol{\mu}^* = \operatorname{argmin} ||\mathbf{v}||^2 \quad \text{s.t. } I(\mathbf{x}, \mathbf{v}) = 1, \tag{4}$$

where $||\mathbf{v}||^2 = \sum_{d=1}^{D} v_d^2$ is the Euclidean norm. OMSV only considers yield estimation, and thus $\mathbf{x}$ is assumed fixed.

C. Bayesian Yield Optimization

To conduct optimization on unknown $g(\mathbf{x})$, BO place a GP prior $\mathbf{g}(\mathbf{x})|\boldsymbol{\theta} \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'|\boldsymbol{\theta})\right)$, with mean $m(\mathbf{x})$ and covariance functions $k(\mathbf{x}, \mathbf{x}'|\boldsymbol{\theta})$. Based on the yield estimation of $g(\mathbf{x}_i)$ subject to some errors, the hyperparameters $\boldsymbol{\theta}$ are estimated by maximum likelihood estimate (MLE) of the likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$, a joint Gaussian distribution. Conditioning on $\mathbf{y}$, we can derive the predictive mean $\bar{g}(\mathbf{x})$ and variance $\hat{v}(\mathbf{x})$ for any $\mathbf{x}$. We can approach the optimal by exploring the areas with higher uncertainty towards the minimum

$$\underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmin}} \left(\bar{g}(\mathbf{x}) - \beta \hat{v}(\mathbf{x})\right), \tag{5}$$

where β tunes the balance between exploration and exploitation. This is known as the upper confidence bound (UCB) [15], which is simple and easy to implement yet powerful and effective. Other acquisition functions include max-value entropy search (MES) and predictive entropy search.

III. Proposed Approach

A. Optimal Proposal Distribution in IS

The crucial step in yield optimization is an accurate and efficient estimation of $g(\mathbf{x})$, which entails a good $\mathbf{x}$ dependent proposal distribution $q(\mathbf{v}|\mathbf{x})$. From Eq. (3), we can see that the optimal proposal distribution $q^*(\mathbf{v}|\mathbf{x})$ is the one that minimizes the approximate variance given by the Delta method, i.e.,

$$q^*(\mathbf{v}|\mathbf{x}) = \underset{q}{\operatorname{argmin}}\, \mathbb{E}_q \left[w^2(\mathbf{v}|\mathbf{x}) \left(I(\mathbf{x},\mathbf{v}) - g(\mathbf{x}) \right)^2 \right], \quad (6)$$

where $w(\mathbf{v}|\mathbf{x}) = p(\mathbf{v})/q(\mathbf{v}|\mathbf{x})$ is the $\mathbf{x}$ dependent importance weight. Utilizing Lagrange multiplier rule for calculus of variations, the optimal proposal distribution is given by

$$q^*(\mathbf{v}|\mathbf{x}) = p(\mathbf{v})I(\mathbf{x},\mathbf{v})/g(\mathbf{x}). \quad (7)$$

To derive a tractable solution, we can take a Laplace approximation of $q^*(\mathbf{v}|\mathbf{x})$, which is a Gaussian distribution centered at $\hat{\boldsymbol{\mu}}(\mathbf{x})$ with covariance $\mathbf{S}(\mathbf{x})$, both of which are $\mathbf{x}$ dependent. The center $\hat{\boldsymbol{\mu}}(\mathbf{x})$ is the mode of $p(\mathbf{v})I(\mathbf{x},\mathbf{v})$, which can be obtained by solving the following optimization problem

$$\hat{\boldsymbol{\mu}}(\mathbf{x}) = \underset{\mathbf{v}}{\operatorname{argmax}}\ \log p(\mathbf{v})I(\mathbf{x},\mathbf{v}). \quad (8)$$

Since $p(\mathbf{v})$ is a standard normal distribution and monotonically decreases with $||\mathbf{v}||^2$ and $I(\mathbf{x},\mathbf{v}) = \{0,1\}$, the maximization in Eq. (8) is equivalent to the minimization of $||\mathbf{v}||^2$, i.e.,

$$\hat{\boldsymbol{\mu}}(\mathbf{x}) = \underset{\mathbf{v}}{\operatorname{argmin}}\ ||\mathbf{v}||^2 \quad \text{s.t.} \quad I(\mathbf{x},\mathbf{v}) = 1. \quad (9)$$

Thus, for a given $\mathbf{x}$, $\hat{\boldsymbol{\mu}}$ is obtained for the smallest $\mathbf{v}$ that satisfies $I(\mathbf{x},\mathbf{v}) = 1$, exactly the solution in OMSV of Eq. (4). Once we have $\hat{\boldsymbol{\mu}}(\mathbf{x})$, we can solve $\mathbf{S}(\mathbf{x})$ by solving the following optimization problem

$$\mathbf{S}(\mathbf{x}) = -\nabla^2_{\mathbf{v}} \log \left(p(\mathbf{v})I(\mathbf{x},\mathbf{v}) \right) \Big|_{\mathbf{v}=\hat{\boldsymbol{\mu}}(\mathbf{x})}, \quad (10)$$

which is ill defined since $I(\mathbf{x},\mathbf{v})$ is not differentiable at $\mathbf{v} = \hat{\boldsymbol{\mu}}(\mathbf{x})$. But if we take derivative at side where $I(\mathbf{x},\mathbf{v}) = 1$, we can get the analytical solution

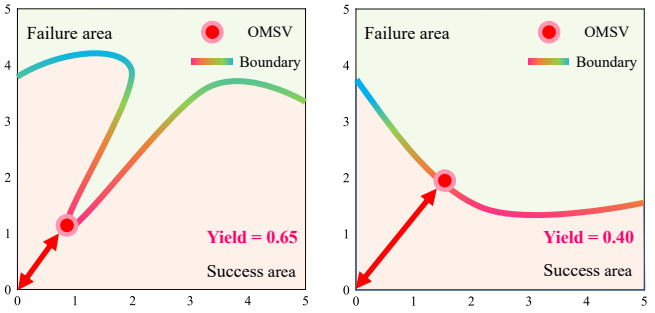
$$\mathbf{S}(\mathbf{x}) = \mathbf{I}, \quad (11)$$

where $\mathbf{I}$ is the identical matrix. Based on the assumption that the Laplace approximation $q^*(\mathbf{v}|\mathbf{x})$ is sufficiently close to the true distribution $p(\mathbf{v}|\mathbf{x})$, the yield will be proportional to the OMSV $\hat{\mu}(\mathbf{x})$ and the yield optimization in Eq. (2) becomes

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmax}}\ ||\hat{\mu}(\mathbf{x})||^2. \quad (12)$$

This is a novel statistical framework providing a theoretical foundation for ASAIS, which has shown great success in yield optimization with excellent efficiency and accuracy.

More importantly, this framework allows us to see the main issues of ASAIS—it seeks only the closest single failure region and ignores other failure regions, leading to inferior performance in practice. For instance, in Fig. 1, ASAIS will



(a) (b)

Fig. 1: An illustrating example of OMSV and why ASAIS fails. Design (a) has a smaller OMSV but a higher yield than (b). However, ASAIS determines the yield based on the length of the OMSV and will choose (b) as a higher yield design.

choose the wrong design that has a lower yield due to the non-convex failure region. Such an issue will become severe as the dimensions increase. We will also show this phenomenon in practical experiments later. Another issue of ASAIS is that the OMSV $\hat{\mu}(\mathbf{x})$ is computed independently for each $\mathbf{x}$, without knowledge sharing between similar designs.

B. Optimal Proposal Distribution

To resolve these challenges, we increase the number of OMSVs to infinite to achieve a better coverage of failure regions. Let us equip an $\mathbf{x}$ dependent infinite mixture of Gaussian to our proposal distribution

$$q(\mathbf{v}|\mathbf{x}) = \sum_{i=1}^{M} \alpha_i \mathcal{N}(\mathbf{v} - \boldsymbol{\mu}_i(\mathbf{x}), s\mathbf{I}), \quad (13)$$

where α_i is the weight and $\mathbf{M} \to \infty$. Now the simple Laplace approximation is not sufficient to optimize $q(\mathbf{v}|\mathbf{x})$. Instead, we introduce more advanced variational inference techniques to approximate the optimal proposal distribution, i.e., minimization of the KL divergence between the mixture distribution and the optimal proposal distribution $\mathrm{KL}(q^*(\mathbf{v}|\mathbf{x})||q(\mathbf{v}|\mathbf{x})) =$

$$\mathbb{E}_{q^*(\mathbf{v}|\mathbf{x})} \left[\log q^*(\mathbf{v}|\mathbf{x}) \right] - \mathbb{E}_{q^*(\mathbf{v}|\mathbf{x})} \left[\log q(\mathbf{v}|\mathbf{x}) \right], \quad (14)$$

where $\mathbb{E}_{q^*(\mathbf{v}|\mathbf{x})} \left[\log q^*(\mathbf{v}|\mathbf{x}) \right]$ is a constant of the entropy of the optimal proposal distribution. Minimization of the KL divergence is equivalent to maximizing $\mathbb{E}_{q^*(\mathbf{v}|\mathbf{x})} \left[\log q(\mathbf{v}|\mathbf{x}) \right]$, which is the lower bound of the optimal solution. We now aim to optimize

$$\underset{\{\boldsymbol{\mu}_i, \alpha_i\}_{i=1}^M}{\operatorname{argmax}} \int \frac{p(\mathbf{v})I(\mathbf{x},\mathbf{v})}{g(\mathbf{x})} \log \left(\sum_{i=1}^{M} \alpha_i \mathcal{N}(\mathbf{v} - \boldsymbol{\mu}_i, s\mathbf{I}|\mathbf{x}) \right) \mathrm{d}\mathbf{v}. \quad (15)$$

The complete solution to Eq. (15) might seem complicated at first glance. But we can see that to get the maximum, the main volumes of the Gaussian components (corresponding to a large α_i) should be placed beyond the failure boundaries $B(\mathbf{v}|\mathbf{x}) = \partial I(\mathbf{x},\mathbf{v})/\partial v \neq 0$. Thus, optimization of the proposal distribution is equivalent to implicitly finding the whole failure boundary (vs. an optimal shift vector in ASAIS) in the variational parameters space.

This finding reveals the connection between IS and surrogate-based methods (e.g., BYA [9])—both methods seek

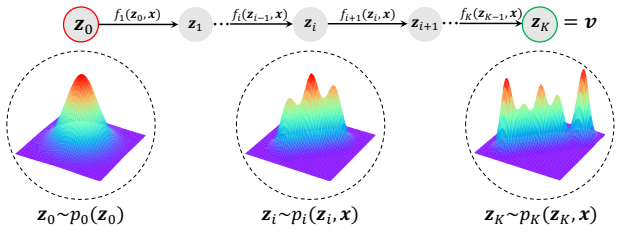


Fig. 2: Conditional Normalizing Flow

the failure boundaries $B(\mathbf{v}|\mathbf{x})$ to success. The surrogate does this explicitly by approximating $I(\mathbf{v}|\mathbf{x})$, leading to higher efficiency but lack of robustness—if $I(\mathbf{v}|\mathbf{x})$ is not accurate enough, the yield estimation and optimization will be misleading. In contrast, optimal IS does this implicitly by updating the proposal distribution $q(\mathbf{v}|\mathbf{x})$ with random samples and might lack some efficiency, but it will not be trapped in local minima and will always converge. It is thus a more reliable method for practical use. The key to successful yield optimization is to combine both, i.e., embedding the $B(\mathbf{v}|\mathbf{x})$ knowledge in the proposal distribution.

C. Conditional Normalizing Flows As Proposal Distribution

To this end, we replace the mixture of Gaussian distributions of Eq. (13) with the SOTA deep learning-based distribution approximation, conditional normalizing flow [16] to approximate $q^*(\mathbf{v}|\mathbf{x})$. CNF is a modification of the generic normalizing flow (NF) [17], which has shown great success in modeling complex distributions by harnessing the power of modern deep learning and massively parallel computing hardware (e.g., GPU). It is capable of approximating any complex distributions conditioned on some parameters (in our case, the design parameters $\mathbf{x}$).

The key idea of CNF is to introduce a conditional network that modulates the parameters of the invertible transformations based on the given conditions. More specifically, CNF defines a series of invertible transformations (e.g., affine transformations and non-linear functions) $f_1, f_2, ..., f_k$, each of which is conditioned on $\mathbf{x}$ (see Fig. 2 for an illustration). In our case, we aim to approximate the optimal proposal distribution $q^*(\mathbf{v}|\mathbf{x})$, and we choose Neural Spline Flows (NSF) [17] as our invertible transformations. The conditional density $q(\mathbf{v}|\mathbf{x})$ is obtained by applying *changes of variables* for a simple base distribution $p(\mathbf{z})$,

$$\mathbf{v} \triangleq \mathbf{z}_K = f_K \circ f_{K-1} \circ \cdots \circ f_1 \left(\mathbf{z}_0; \gamma, \mathbf{x}\right), \tag{16}$$

where $\mathbf{z}_0$ is a sample from $p(\mathbf{z})$, $\mathbf{z}_K$ is the final output after the flow transformations, γ are parameters controlling the flow, and $\mathbf{x}$ is the conditional variable.

Training for CNF is straightforward by MLE on samples generated from the target distribution. In our case, these are samples that satisfy $I(\mathbf{v}_i, \mathbf{x}_j) = 1$. The Jacobian of each flow transformation f_k is used to compute the log-likelihood:

$$\mathcal{L} = \log p\left(\mathbf{z}_0\right) - \sum_{k=1}^{K} \log \left| \frac{\partial f_k\left(\mathbf{z}_{k-1}; \gamma, \mathbf{x}\right)}{\partial \mathbf{z}_{k-1}} \right|. \tag{17}$$

The parameters γ are updated using stochastic gradient descent, with the gradients easily computed via chain rules.

Through the forward and inverse transformations, we can generate high-quality samples from approximated $q^*(\mathbf{v}|\mathbf{x})$, validate these samples, estimate the yield, and update the CNF to get better approximate to $q^*(\mathbf{v}|\mathbf{x})$. Another major advantage of CNF is that the underlying knowledge of $I(\mathbf{x}, \mathbf{v})$ is implicitly encoded in the conditional distribution $q^*(\mathbf{v}|\mathbf{x})$, which suggests that the knowledge between different designs can be learned by CNF. Meaning that an optimal proposal learned from one design can be used to approximate the optimal proposal for another design. We will show the effectiveness of this knowledge transfer in a joint yield estimation task in later experiments.

D. Gradient-based OPT Yield Optimization

Another huge advantage of the CNF is that the derivative w.r.t $\mathbf{x}$ is directly available using chain rules. This allows us to directly optimize the yield by gradient-based methods without the need for another surrogate model (e.g., in WEIBO and MESBO) or a simplification of our target function (e.g., in ASAIS). Given a trained CNF $q(\mathbf{v}|\mathbf{x})$, the yield $g(\mathbf{x})$ can be estimated by

$$\hat{g}(\mathbf{x}) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{I(\mathbf{x}, \mathbf{v}_i) p(\mathbf{v}_i)}{q(\mathbf{v}_i|\mathbf{x})} \approx \frac{1}{N} \sum_{i=1}^{N} \frac{p(\mathbf{v}_i)}{q(\mathbf{v}_i|\mathbf{x})}. \tag{18}$$

We can make this approximation because the CNF is trained to approximate the optimal proposal. If that is the case, for all samples $\mathbf{v}_i$ from $q(\mathbf{v}|\mathbf{x})$, $I(\mathbf{x}, \mathbf{v}_i)$ is always 1 and the approximation is exact. In practice, that is not the case and the approximation is biased. However, that is sufficient to guide us to move to the next design points for yield optimization. We do not expect our method to reach the optimal design in one step. At the next design point $\mathbf{x}'$, $q(\mathbf{v}|\mathbf{x}')$ will propose more samples, which are then passed to a SPICE-based indication $I(\mathbf{x}', \mathbf{v})$ for validation, and the failure samples are collected to update the CNF about the failure boundary for design $\mathbf{x}'$ and, most importantly, through the knowledge transfer, about the whole design parameters space. As more data is collected, the CNF refines itself and further approximates the optimal proposal $q^*(\mathbf{v}|\mathbf{x})$, leading to a better estimation of $g(\mathbf{x})$ and a more accurate gradient of the yield w.r.t $\mathbf{x}$. Based on the estimation of Eq. (18), the gradient of the estimated yield w.r.t $\mathbf{x}$ is given by

$$\nabla_{\mathbf{x}} \hat{g}(\mathbf{x}) = -\frac{1}{N} \sum_{i=1}^{N} \frac{p(\mathbf{v}_i)\, \nabla_{\mathbf{x}} \log q(\mathbf{v}_i|\mathbf{x})}{\left(\mathbf{q}(\mathbf{v}_i|\mathbf{x})\right)^2}, \tag{19}$$

where $\nabla_{\mathbf{x}} \log q(\mathbf{v}_i|\mathbf{x})$ is the gradient of the CNF w.r.t $\mathbf{x}$ and is easily computed using chain rules. The gradient-based OPT yield optimization is summarized in Algorithm 1.

E. Uncertainty Quantification Using Sequential Ensemble

Despite that the gradient-based OPT yield optimization is able to find the optimal design $\mathbf{x}^*$ given sufficient computational budget, the uncertainty of the yield estimation $\hat{g}(\mathbf{x})$ introduced by the CNF and lack of data is not well quantified, leading to a large variance of the optimization.

A commonly used method for quantifying the uncertainty of deep learning is the deep ensemble, which aggregates the predictions of an ensemble of models that are trained using different random initializations or hyperparameters. Despite its success, the deep ensemble method does not apply to our problem well because it accounts for uncertainty due to the

Algorithm 1 Gradient-based OPT Yield Optimization

Require: SPICE-based Indication $I(\mathbf{v}, \mathbf{x})$, N, N_{iter}

Generate a random design $\mathbf{x}_0$ and N random samples $\mathbf{v}_j$
Pass $\{\mathbf{v}_j, \mathbf{x}_0\}_{j=1}^{N}$ to SPICE-based indication $I(\mathbf{v}_j, \mathbf{x}_0)$ to get failure samples $\mathcal{D} = \{v_j, \mathbf{x}_0\}_{j=1}^{N'}(N' \leq N)$
for i = 1 to N_{iter} **do**
  Update CNF $q(\mathbf{v}|\mathbf{x})$ with dataset $\mathcal{D}$
  Optimize estimated yield $\hat{g}(\mathbf{x})$ of Eq. (18) with its gradient in Eq. (19) to get optimal design $\mathbf{x}_i^*$
  Generate N samples $\mathbf{v}_j$ from $q(\mathbf{v}|\mathbf{x}_i^*)$.
  Pass $\mathbf{v}_j$ to the circuit SPICE-based indication $I(\mathbf{v}_j, \mathbf{x}_i^*)$ to get failure samples $\mathcal{D}_i = \{v_j, \mathbf{x}_i^*\}_{j=1}^{N'}(N' \leq N)$
  Update dateset $\mathcal{D} = \mathcal{D} \cup \mathcal{D}_i$
end for
return $\mathbf{x}_i^*$

model initializations or hyperparameters instead of the lack of data, which is the main source of uncertainty in yield.

Inspired by the figure of merit (FOM) concept that is used to determine the convergence of a yield estimation, we propose a sequential ensemble method to quantify the uncertainty of the yield estimation $\hat{g}(\mathbf{x})$. The key idea of the sequential ensemble is to keep the latest R models of the CNF and use them to estimate the uncertainty due to the lack of data as in FOM. More specifically, at j iteration of the optimization, a copy of the CNF, $q_j(\mathbf{v}|\mathbf{x})$ is saved before its update. Then, the uncertainty of the yield estimation $\hat{g}(\mathbf{x})$ is estimated as the standard deviation $\hat{v}(\mathbf{x})$ of the yield estimation $\hat{g}(\mathbf{x})$ using these R models, i.e.,

$$\hat{v}(\mathbf{x}) = (\frac{1}{R}\sum_{r=j-R}^{j}(\hat{g}_r(\mathbf{x}) - \bar{g}(\mathbf{x}))^2)^{1/2}, \tag{20}$$

where $\hat{g}_r(\mathbf{x})$ is the yield estimation based on CNF at r iteration based on Eq. (18) and $\bar{g}(\mathbf{x})$ is the mean of the yield estimation using the R models.

$$\bar{g}(\mathbf{x}) = \frac{1}{NR}\sum_{i=1}^{N}\sum_{r=j-R}^{j}\frac{p(\mathbf{v}_i)}{q_r(\mathbf{v}_i|\mathbf{x})}. \tag{21}$$

F. Bayesian OPT Yield Optimization

Once we obtain the mean and standard deviation by the sequential ensemble, we can perform the yield optimization based on BO of Eq. (5). More specifically, for yield optimization, instead of directly optimizing the yield estimation model, we optimize the UCB acquisition function,

$$a_{UCB}(\mathbf{x}) = \bar{g}(\mathbf{x}) - \beta\hat{v}(\mathbf{x}), \tag{22}$$

which automatically balances the tradeoff between exploration and exploitation for a better design and significantly reduces the variance of optimization. Note that the second term's sign is minus because $\bar{g}(\mathbf{x})$ is the failure rate (1-yield) and thus the optimization is a minimization of Eq. (22).

Now the optimization of yield becomes the minimization of the UCB acquisition function Eq. (22) iteratively until we find the optimal design $\mathbf{x}^*$. The gradient of the UCB acquisition function is easily calculated as both $\bar{g}(\mathbf{x})$ and $\hat{v}(\mathbf{x})$ are linear combinations of the $q_r(\mathbf{v}_i|\mathbf{x})$, whose gradient is given by Eq. (19). The summary of the Bayesian OPT yield optimization is summarized in Algorithm 2, which is applied in our experiments.

Algorithm 2 Bayesian OPT Yield Optimization

Require: SPICE-based Indication $I(\mathbf{v}, \mathbf{x})$, N, N_{iter}

Generate a random design $\mathbf{x}_0$ and N random samples $\mathbf{v}_j$
Pass $\{\mathbf{v}_j, \mathbf{x}_0\}_{j=1}^{N}$ to SPICE-based indication $I(\mathbf{v}_j, \mathbf{x}_0)$ to get failure samples $\mathcal{D} = \{v_j, \mathbf{x}_0\}_{j=1}^{N'}(N' \leq N)$
Train 5 CNF $\{q_j(\mathbf{v}|\mathbf{x})\}_{j=-4}^{0}$ with dataset $\mathcal{D}$/5
for i = 1 to N_{iter} **do**
  Update CNF $q(\mathbf{v}|\mathbf{x})$ with dataset $\mathcal{D}$
  Get $\bar{g}(\mathbf{x}_{i-1})$ and $v(\mathbf{x}_{i-1})$ from previous five trained CNF $\{q_j(\mathbf{v}|\mathbf{x})\}_{j=-5}^{i}$ according to Eq. (21) and Eq. (20)
  Optimize acquisition of Eq. (22) with its gradient to get next design $\mathbf{x}_i^*$
  Generate N samples $\mathbf{v}_j$ from $q(\mathbf{v}|\mathbf{x}_i^*)$.
  Pass $\mathbf{v}_j$ to the circuit SPICE-based indication $I(\mathbf{v}_j, \mathbf{x}_i^*)$ to get failure samples $\mathcal{D}_i = \{v_j, \mathbf{x}_i^*\}_{j=1}^{N'}(N' \leq N)$
  Update dataset $\mathcal{D} = \mathcal{D} \cup \mathcal{D}_i$
end for
return $\mathbf{x}_i^*$

IV. Experimental Results

We assess the performance of OPT in terms of accuracy and efficiency on a set of benchmark circuits, including an operational transconductance amplifier (OTA), a 6T-SRAM, and an adder circuit. Five SOTA yield optimization methods, namely, WEIBO [2], MESBO [11], ASAIS [14], KDEBO [13], and, BYA [9] are implemented for a thorough comparison. To assess the robustness of each method, we introduce two distinct circuit specifications, namely higher and lower (referred to as Case 1 and Case 2), for each circuit in our yield optimization experiments. The optimal design is validated using MC with 4e7 and 1e6 simulations in Case 1 and Case 2, respectively. In Case 1, if only one failure event is detected among the 4e7 simulations, its failure rate will be 2.5e-8. To ensure a fair and meaningful comparison, each algorithm runs 10 times with different seeds to reduce random fluctuations.

For all experiments, the CNF uses 8 functional transformations composited by the same 2-layer multi-layer perceptron (MLP). In each MLP, the number of hidden units is 10 times the number of the process variation parameters for any yield problems. ReLU activation function [18] is adopted. The Adam optimizer [19] is employed for all optimization processes. The update of the CNF uses 500 iterations whereas the optimization of the acquisition is 200 iterations. For yield optimization, we use 10 iterations, each of which proposes 30 samples. The initialization depends on the variational dimension of the problem. The baseline methods are implemented using their (default) settings as suggested in their respective papers. Because some methods do not generalize well, we also fine-tune some hyperparameters for them for different circuits to achieve better performance. In contrast, OPT does not require any hyperparameters tuning for any of our experiments. All experiments are conducted on a workstation equipped with AMD 7950x CPU and 32GB RAM.

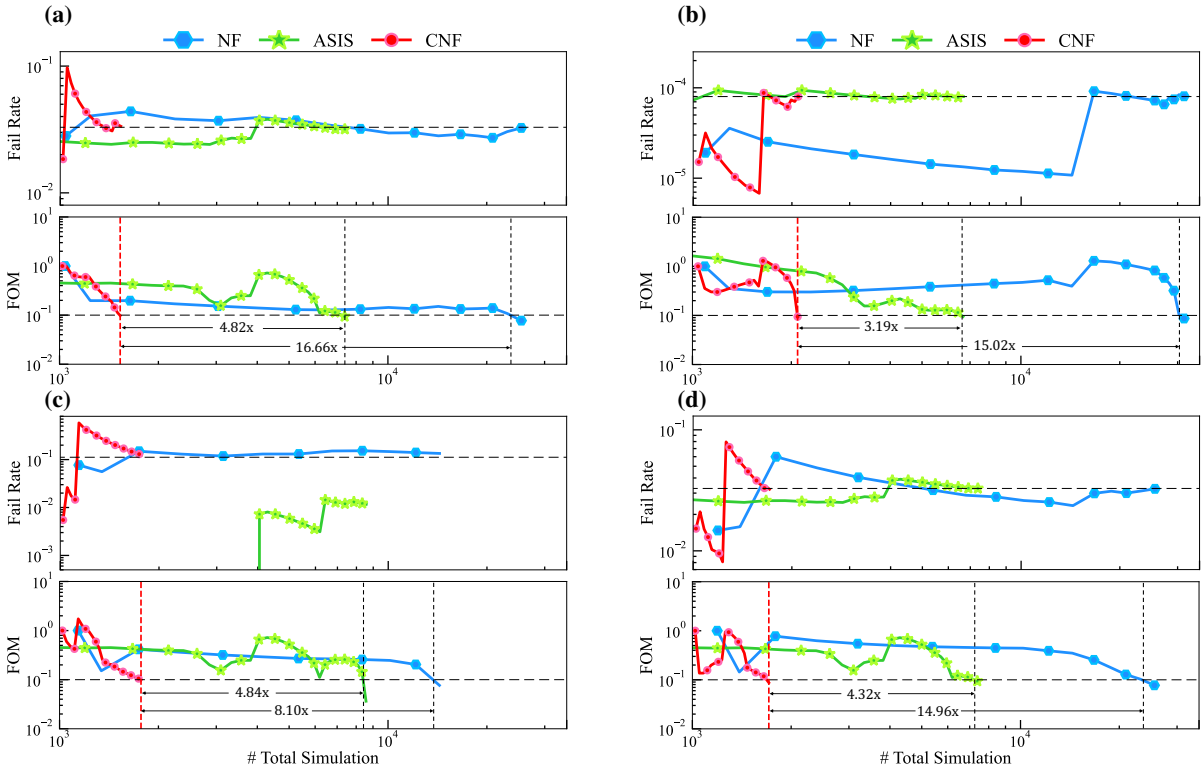


Fig. 3: Validation of knowledge transfer via joint yield estimation for four designs at corners

A. Validation of OPT Thought Joint Yield Estimation

Before we move to yield optimization, we first validate the capacity of knowledge transfer between different designs in an adder circuit (details described later) using OPT. The success of this experiment will reveal OPT's foundation for its superior performance in yield optimization. In this experiment, we conduct yield estimation for the four designs at the corners of the design parameters space at the same time. We call this joint yield estimation. For implementation, we update each model based on the different designs alternatively, each time with a small number of simulations. We use normal NF, which can be only used individually for each design, as a reference without knowledge transfer, and All-Sensitivity Importance Sampling (ASIS) [14], which has the capacity to share knowledge between different designs but is based on the OMSV, as a comparison.

The ground-truth log failure rate, yield estimation, and FOM as more simulations are conducted are shown in Fig. 3. The FOM $\rho = \mathrm{std}(P_f)/P_f$ (where $\mathrm{std}(P_f)$ is the standard deviation of estimated failure rate) is used as the stopping criterion for all methods with $\rho = 0.1$ (indicating at least 90% accurate with 90% confidence interval) as in many previous works, e.g., [3], [20], [21]. We can learn that knowledge transfer enables both ASIS and OPT to conduct joint yield estimation simultaneously for all four distinct designs, resulting in a significant reduction in simulation costs compared to NF. Because OPT is based on the truth optimal proposal distribution, it converges to the true value quickly and accurately. In contrast, based on OMSV, ASIS shows a slow convergence rate and sometimes cannot converge to the true value for designs with a high failure rate, as shown in Fig. 3c. In these four different designs, OPT achieves an average speedup of 13.69x and 4.29x compared to NF and ASIS, respectively.

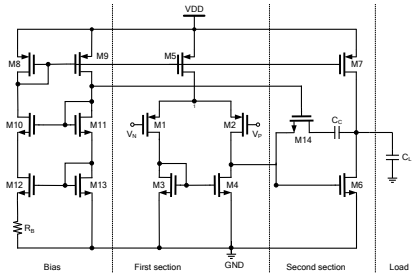


Fig. 4: Operational Transconductance Amplifier Circuit

TABLE I: Yield optimization report for the OTA circuit

Case	Method		WEIBO	MESBO	ASAIS	KDEBO	BYA	Proposed
1	Yield	Best	**99.99%**	**99.99%**	99.96%	**99.99%**	**99.99%**	**99.99%**
		Worst	99.06%	99.06%	99.84%	99.71%	99.60%	**99.93%**
		Mean	99.68%	99.66%	99.90%	99.96%	99.94%	**99.97%**
		Std	0.35%	0.42%	0.04%	0.09%	0.12%	**0.02%**
	#Sim	Best	3352	1500	2169	8000	6600	**890**
		Worst	9684	5900	5302	8000	6600	**890**
		Mean	5055	4380	3192	8000	6600	**890**
2	Yield	Best	99.84%	99.85%	99.87%	99.85%	99.84%	**99.90%**
		Worst	99.52%	**99.83%**	99.27%	99.48%	99.82%	99.79%
		Mean	99.73%	99.84%	99.45%	99.69%	99.83%	**99.87%**
		Std	0.15%	**3.91e-3%**	0.24%	0.10%	7.07e-3%	0.04%
	#Sim	Best	2942	3570	3515	8000	6600	**670**
		Worst	7465	5900	3550	8000	6600	**670**
		Mean	4294	5111	3546	8000	6600	**670**

B. Operational Transconductance Amplifier Circuit

The OTA circuit (shown in Fig. 4) is implemented in a 180 nm CMOS process which consists of 14 transistors. The circuit contains three design variables: the transistor widths

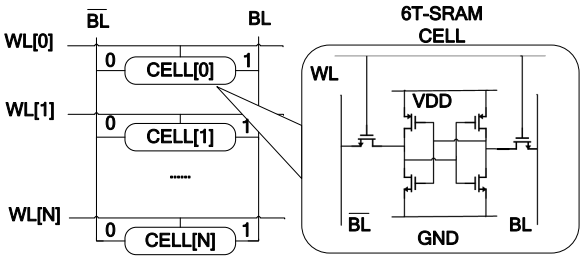


Fig. 5: The structure of SRAM column circuit

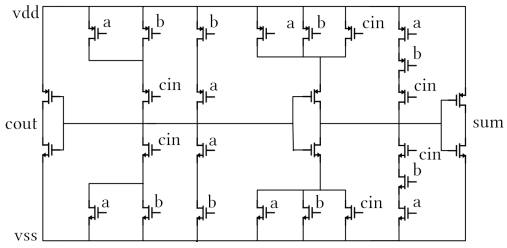


Fig. 6: The structure of Adder circuit

TABLE II: Yield optimization report for the SRAM circuit

Case	Method		WEIBO	MESBO	ASAIS	KDEBO	BYA	Proposed
1	Fail Rate	Best	**2.50e-8**	**2.50e-8**	**2.50e-8**	**2.50e-8**	**2.50e-8**	**2.50e-8**
		Worst	3.86e-4	2.70e-6	2.50e-7	5.19e-4	1.50e-7	**5.00e-8**
		Mean	3.96e-5	4.23e-5	6.50e-8	1.91e-4	5.00e-8	**3.00e-8**
		Std	1.15e-4	8.10e-7	6.82e-8	2.32e-4	3.54e-8	**1.00e-8**
	#Sim	Best	1161	2880	1186	7000	11000	**1020**
		Worst	2951	6180	1200	7000	11000	**1020**
		Mean	3614	4064	1192	7000	11000	**1020**
2	Fail Rate	Best	2.00e-6	2.00e-6	**1.00e-6**	1.24e-3	5.00e-6	2.00e-6
		Worst	5.20e-4	1.21e-4	4.50e-5	1.72e-2	1.16e-4	**6.00e-6**
		Mean	1.60e-4	3.22e-5	1.75e-5	1.02e-2	2.17e-5	**4.00e-6**
		Std	1.56e-4	4.16e-5	1.55e-5	4.19e-3	3.16e-5	**1.20e-6**
	#Sim	Best	901	1580	803	2600	8000	**800**
		Worst	3151	3440	807	6500	8000	**800**
		Mean	1815	2395	804	6110	8000	**800**

TABLE III: Yield optimization report for the adder circuit

Case	Method		WEIBO	MESBO	ASAIS	KDEBO	BYA	Proposed
1	Fail Rate	Best	7.50e-7	5.00e-8	**2.50e-8**	5.00e-8	4.00e-8	**2.50e-8**
		Worst	2.08e-5	1.50e-7	7.50e-8	1.30e-6	**5.00e-8**	**5.00e-8**
		Mean	6.80e-6	6.00e-8	4.75e-8	3.45e-7	5.50e-8	**4.05e-8**
		Std	8.61e-6	3.00e-8	2.08e-8	4.54e-7	**5.00e-9**	9.78e-9
	#Sim	Best	2121	4070	2417	10000	11000	**1475**
		Worst	4861	11200	2427	10000	11000	**1475**
		Mean	3626	8460	2422	10000	11000	**1475**
2	Fail Rate	Best	1.50e-5	1.03e-5	**9.00e-6**	1.10e-5	1.70e-5	**9.00e-6**
		Worst	1.90e-5	2.00e-5	2.60e-5	1.46e-4	**1.75e-5**	1.80e-5
		Mean	1.74e-5	1.70e-5	1.83e-5	5.24e-5	1.72e-5	**1.33e-5**
		Std	1.36e-6	2.83e-6	5.88e-6	5.40e-5	**2.29e-7**	2.90e-6
	#Sim	Best	2681	4220	2412	8000	8000	**1310**
		Worst	4391	7870	2420	8000	8000	**1310**
		Mean	3536	5687	2415	8000	8000	**1310**

of M5, M7, and M13. Additionally, each transistor has four process variation parameters, namely oxide thickness, threshold voltage, and variations in transistor length and width due to process deviation. In our experiments, the performance of interest is the quiescent current I_Q at $27°C$.

The yield optimization experimental results in Case 1 and Case 2 are provided in Table I. Most methods achieve high-yield results for their best performance in Case 1, indicating their capacity with proper settings. Nonetheless, the mean performance shows their lack of stability and robustness. OPT achieves a performance improvement of $0.01\% - 0.31\%$ with a speedup of 5.96x-10.88x over the baseline methods. In the more challenging Case 2, the best performance of the competitors is inferior to OPT by a significant margin. MESBO outperforms OPT slightly in Case 2 for the worst case but with about 8x more simulations. In contrast, OPT outperforms all baseline methods with a significant margin in almost all cases. Based on the mean performance in Case 1 and Case 2, OPT achieves an average speedup of 5.29x-11.94x over the baseline methods for a performance improvement of $0.03\% - 0.42\%$.

C. 6T-SRAM Circuit

The SRAM bit-cell (with a simplified schematic of an SRAM column presented in Fig. 5) is implemented in a 45nm CMOS process. It consists of six transistors, each of which has three independent random variables, namely, threshold voltage, mobility, and gate oxide width. This creates 18 independent variational parameters for each SRAM cell. The width and length of a single transistor are specified as the design variables to be optimized. In our experiments, we focus on optimizing the write delay as the performance metric of interest. The yield optimization experimental results are shown in the Table II.

In the easier Case 1, all methods manage to produce the same best results. However, if we look at the mean performance over 10 experiments, OPT demonstrates remarkable improvement over the baseline methods with a performance improvement of 1.66x-6,367x and a speedup of 1.17x-10.78x. In the more challenging Case 2, ASAIS does outperform OPT with its best performance. Nonetheless, looking at the mean performance of yield optimization, OPT achieves a performance improvement of 4.38x-2,550x with a speedup of 1.01x-10x over the baseline methods. AIAIS is the second best method with a very close number of total simulations to OPT. Due to the reliance on a single OMSV, it fails to consistently achieve the best performance.

D. Adder Circuit

The adder circuit (Fig. 6) uses a total of 28 MOS transistors, each of which is subject to the same three variational parameters, introducing a total of 84 variational parameters. In terms of circuit design, we concentrate on two particular design variables: the width and length of an individual transistor. We evaluate the time-to-threshold (TT) performance within a specified range of $27°C$ and determine the yield by simulating the transient response until the sum output reaches a predefined threshold voltage. The experimental results for yield optimization are presented in Table III.

In this tough assessment, only ASAIS and OPT manage to achieve the best potential design. BYA also shows a good performance in terms of worst performance and standard deviation. We believe that, based on practical projects, the

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

		#Simulation=5000						#Simulation=10000					
Circuit	Metric	WEIBO	MESBO	ASAIS	KDEBO	BYA	Proposed	WEIBO	MESBO	ASAIS	KDEBO	BYA	Proposed
OTA (Yield)	Best	99.94%	**99.99%**	99.96%	**99.99%**	99.98%	**99.99%**	**99.99%**	**99.99%**	99.98%	**99.99%**	**99.99%**	**99.99%**
	Worst	99.08%	99.07%	99.84%	90.94%	99.60%	**99.93%**	99.05%	99.93%	99.86%	99.91%	99.60%	**99.95%**
	Mean	99.52%	99.82%	99.90%	98.96%	99.93%	**99.97%**	99.53%	**99.97%**	99.93%	99.96%	99.94%	**99.97%**
	Std	0.33%	0.37%	0.04%	2.68%	0.12%	**0.02%**	0.39%	0.02%	0.05%	0.09%	0.12%	**0.01%**
6T-SRAM (Fail Rate)	Best	**2.50e-8**	**2.50e-8**	**2.50e-8**	**2.50e-8**	**2.50e-8**	**2.50e-8**	**2.50e-8**	**2.50e-8**	**2.50e-8**	**2.50e-8**	**2.50e-8**	**2.50e-8**
	Worst	3.86e-4	2.70e-6	**5.00e-8**	9.24e-2	4.82e-1	**5.00e-8**	5.75e-6	**5.00e-8**	**5.00e-8**	5.19e-4	1.50e-7	**5.00e-8**
	Mean	3.96e-5	4.23e-5	3.75e-8	9.53e-3	1.54e-1	**3.00e-8**	1.24e-6	3.06e-8	3.33e-8	1.91e-4	5.00e-8	**2.75e-8**
	Std	1.15e-4	8.10e-7	1.25e-8	2.76e-2	1.98e-1	**1.00e-8**	1.73e-6	1.04e-8	1.18e-8	2.32e-4	3.54e-8	**7.50e-9**
Adder (Fail Rate)	Best	7.50e-7	5.00e-8	**2.50e-8**	5.00e-8	4.00e-8	**2.50e-8**	3.33e-8	5.00e-8	**2.50e-8**	5.00e-8	4.00e-8	**2.50e-8**
	Worst	2.08e-5	1.24e-3	6.67e-8	2.03e-2	**5.00e-8**	**5.00e-8**	2.00e-6	1.50e-7	**5.00e-8**	1.30e-6	**5.00e-8**	**5.00e-8**
	Mean	6.80e-6	2.50e-4	4.35e-8	2.84e-3	4.50e-8	**3.88e-8**	1.55e-6	6.00e-8	4.44e-8	3.45e-7	4.50e-8	**3.55e-8**
	Std	8.61e-6	4.91e-4	1.67e-8	6.04e-3	**5.00e-9**	9.43e-9	7.58e-7	3.00e-8	1.04e-8	4.54e-7	**5.00e-9**	7.68e-9

mean performance is more of a concern. In that sense, OPT is the best among all methods in delivering the best yield design with 10 runs. More surprisingly, OPT manages to do this with always the minimum number of simulations. In Case 1, OPT achieves a performance improvement of 1.17x-168x with a speedup of 1.64x-7.46x; In Case 2, OPT achieves a performance improvement of 1.28x-3.94x with a speedup of 1.84x-6.11x.

E. Resource-based Comparison

Averaging the results over all three experiments, OPT achieves an average speedup of 5.0x-8.7x with higher yield designs compared to other baselines. However, this conclusion might not be convincing enough because each model has its own stopping criteria, and the reported results can be biased as some methods might perform very well at some stages. We follow the classic way to compare optimization methods based on the same given resources. All methods are altered to remove their stopping criteria and kept running while we compare the optimized yield against the time/computational resources. We conduct the Case 1 experiment for all the previous circuits. Results at total simulations of 5,000 and 10,000 are recorded. The statistical results over 10 random runs are shown in Table IV.

For the OTA circuit, OPT is always better in all cases. With only 5,000 simulations, OPT can achieve an average yield of 99.97%, which outperforms all other methods even with 10,000 simulations. In contrast, only MESBO can achieve similar performance to OPT with 10,000 simulations. Compared to all baselines, OPT achieves a performance improvement of 0.06%-0.45% with 5,000 simulations and 0.00%-0.44% with 10,000 simulations. For the classic 6T-SRAM circuit, all methods manage to achieve the best performance, which is consistent with the literature. Nonetheless, most of them need 10,000 simulations to achieve close results to OPT with 5,000 simulations. Compared to all baselines, OPT achieves a performance improvement of 1.25x-5,133,333x with 5,000 simulations and 1.11x-6,945x with 10,000 simulations, highlighting the efficiency of OPT in optimization with a limited number of simulations. For the adder circuit, most methods can achieve reasonably good results with 5,000 simulations except for MESBO. OPT shows clear stability and the best performance for almost all cases except for the standard deviation of BYA. Compared to all baselines, OPT achieves a performance improvement of 1.12x-73,000x with 5,000 simulations and 1.25x-44x with 10,000 simulations, again highlighting the efficiency of OPT in optimization.

TABLE V: Sequential Ensemle vs. Deep Ensemble

	OTA (Yield)		6T-SRAM (Fail Rate)		Adder (Fail Rate)	
	Deep Ens.	Seq.Ens.	Deep Ens.	Seq.Ens.	Deep Ens.	Seq.Ens.
Best	99.87%	**99.99%**	5.00e-8	**2.50e-8**	5.00e-8	**2.50e-8**
Worst	99.77%	**99.93%**	5.05e-4	**5.00e-8**	1.50e-7	**5.00e-8**
Mean	99.82%	**99.97%**	1.01e-4	**3.00e-8**	6.43e-8	**4.05e-8**
Std	0.03%	**0.02%**	2.00e-4	**1.00e-8**	3.50e-8	**9.78e-9**

F. Ablation Study for Sequential Ensemble

Finally, we assess the effectiveness of the proposed sequential ensemble method by replacing it with the classic deep ensemble method and conducting yield optimization in Case 1 experiments of the three circuits. The results are presented in Table V, where the sequential ensemble shows a clear improvement over the deep ensemble with an improvement of 1.59x-30,000x.

V. Conclusion

We propose OPT, a novel yield optimization framework equipped with CNF and BO without the need for a surrogate for the first time. The superiority is supported by extensive experiments on real-world circuit benchmarks, ablation studies, and special designed joint yield estimation validation. The performance of OPT can be further improved by using other advanced sampling methods, e.g., generative diffusion model.

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 Transactions on Computer-Aided Design of Integrated Circuits and Systems, 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 Transactions on Computer-Aided Design of Integrated Circuits and Systems, 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 2008 IEEE/ACM International Conference on Computer-Aided Design. 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 2018 55th ACM/ESDA/IEEE Design Automation Conference (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 Proceedings of the 2019 International Symposium on Physical Design, 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 2020 IEEE/ACM International Conference On Computer Aided Design (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 Proceedings of the 56th Annual Design Automation Conference 2019, 2019, pp. 1–6.

[8] S. Yin, G. Dai, and W. W. Xing, "High-dimensional yield estimation using shrinkage deep features and maximization of integral entropy reduction," in 2023 28th Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2023.

[9] 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, 2022, pp. 1195–1200.

[10] M. Wang, F. Yang, C. Yan, X. Zeng, and X. Hu, "Efficient bayesian yield optimization approach for analog and sram circuits," in 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC). IEEE, 2017, pp. 1–6.

[11] S. Zhang, F. Yang, D. Zhou, and X. Zeng, "Bayesian methods for the yield optimization of analog and sram circuits," in 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC).

[12] C. Jiang, X. Fan, Y. Xing, C. Duan, and J. Zhang, "Min norm failure vector guided yield optimization method for nanometer sram design," in 2019 International Conference on Computer, Information and Telecommunication Systems (CITS), 2019, pp. 1–4.

[13] 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 Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 41, no. 3, pp. 695–708, 2022.

[14] 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 2021 58th ACM/IEEE Design Automation Conference (DAC), 2021, pp. 895–900.

[15] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," IEEE Transactions on Information Theory, vol. 58, no. 5, pp. 3250–3265, 2012.

[16] C. Winkler, D. E. Worrall, E. Hoogeboom, and M. Welling, "Learning likelihoods with conditional normalizing flows," ArXiv, vol. abs/1912.00042, 2019.

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

[18] 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

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

[20] 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, 2016, pp. 153–160.

[21] J. Yao, Z. Ye, and Y. Wang, "An efficient sram yield analysis and optimization method with adaptive online surrogate modeling," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 7, pp. 1245–1253, 2015.