NOFIS: Normalizing Flow for Rare Circuit Failure Analysis

Zhengqi Gao
MIT
Cambridge, MA, USA
zhengqi@mit.edu

Dinghuai Zhang
Université de Montréal
Montréal, QC, Canada
dinghuai.zhang@mila.quebec

Luca Daniel
MIT
Cambridge, MA, USA
dluca@mit.edu

Duane S. Boning
MIT
Cambridge, MA, USA
boning@mtl.mit.edu

ABSTRACT

Accurate estimation of rare failure occurrence probability is crucial for ensuring the proper and reliable functioning of integrated circuits (ICs). Conventional Monte Carlo methods are inefficient, demanding an exorbitant number of samples to achieve reliable estimates. Inspired by the exact sampling capabilities of normalizing flows, we revisit this problem and propose normalizing flow assisted importance sampling, termed NOFIS. NOFIS first learns a sequence of proposal distributions associated with predefined nested subset events by minimizing KL divergence losses. Next, it estimates the rare event probability by utilizing importance sampling in conjunction with the last proposal. The efficacy of our NOFIS method is substantiated through comprehensive qualitative visualizations, affirming the optimality of the learned proposal distribution, as well as 10 quantitative experiments, which highlight NOFIS's superior accuracy over baseline approaches.

CCS CONCEPTS

• **Hardware → Electronic design automation**.

KEYWORDS

Rare circuit failure, importance sampling, normalizing flows

ACM Reference Format:
Zhengqi Gao, Dinghuai Zhang, Luca Daniel, and Duane S. Boning. 2024. NOFIS: Normalizing Flow for Rare Circuit Failure Analysis. In *61st ACM/IEEE Design Automation Conference (DAC '24), June 23–27, 2024, San Francisco, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649329.3658459

1 INTRODUCTION

A rare event [4] is characterized by an occurrence probability close to zero (e.g., less than 10^{-4}). The estimation of such rare event probabilities is of significant interest across various domains, particularly in integrated circuits (ICs) [10, 11, 14–17, 20, 21]. For instance, it has been illustrated that for an SRAM array to function properly, each individual SRAM cell needs to have a failure rate (i.e., probability) less than 10^{-6} [2, 8, 10, 12].

The Monte Carlo (MC) approach is widely recognized as inefficient for the rare circuite failure probability estimation problem [7, 20]. For instance, when aiming to estimate a small probability such as 10^{-6}, the MC method may require more than 10^8 samples to achieve a relatively low estimation variance. However, gathering such a large number of samples can be unaffordable, as typically the data acquisition needs to invoke expensive circuit simulations. In other words, beyond the pursuit of estimation accuracy, the number of data samples used is a critical metric as well.

To confront this challenge—ensuring precise probability estimation for rare circuit failure within a data sample budget, various methods were established [1, 10–12, 14–17, 20, 21, 23], with the earliest dating back to at least [10]. To name a few works, subset simulation [1] involves constructing a series of *nested subset events* [1] with progressively decreasing occurrence probabilities, with the last subset representing the original rare event of interest. It has been applied to estimate SRAM and DFF rare failure probabilities under semiconductor process variations [19]. Importance sampling (IS)[4, 10, 14] aims to obtain a *proposal distribution* and estimate the rare circuit failure probability through a weighted ratio. APA[23] and APE [21] are specifically designed, utilizing the correlated characteristics of cells in an SRAM array, to yield more accurate estimates compared to loop flattening [23]. Meta-model with tensor approximation [15] and scaled-sigma sampling [20] have also been proposed for addressing this problem.

We posit that the recently popularized technique of normalizing flows (NFs) [5, 6, 13] provides an unprecedented and highly efficient tool for rare circuit failure probability estimation. The elegance of applying it to this task is that NFs impose a sequence of transformations to shift a base distribution to a desired target distribution, and we realize that this procedure could be adapted to reflect the learning of a sequence of proposal distributions associated with several nested subset events [1]. By setting the original rare event as the last subset event, the ultimate shifted distribution in the NFs will be a good proposal distribution for the original rare event. Thus, this final proposal distribution can be combined with IS to generate an accurate estimate of the original rare event probability. To verify the proposed NOFIS method, we conducted extensive 2-D visualizations to justify its strong capability in recovering the theoretically optimal proposal distribution. Moreover, compared to six baseline methods across 10 test cases (covering Opamp, Charge

DAC '24, June 23–27, 2024, San Francisco, CA, USA

ACM ISBN 979-8-4007-0601-1/24/06
https://doi.org/10.1145/3649329.3658459

Pump, and photonic Y-branch), NOFIS consistently demonstrates superior estimation accuracy with fewer data samples.

2 PROPOSED METHOD

Mathematically, the rare circuit failure probability estimation problem is defined by a tuple $\mathcal{F} = (p, \Omega)$, where $p(\cdot) \in \mathcal{P}^D$ represents a D-dimensional data generating distribution, and $\Omega \subseteq \mathbb{R}^D$ represents the integral region associated with the rare event. Without loss of any generality and for conciseness, we parameterize $\Omega = \{\mathbf{x} \in \mathbb{R}^D | g(\mathbf{x}) \leq 0\}$ by a characteristic function $g(\cdot) : \mathbb{R}^D \to \mathbb{R}$. As an example, semiconductor process variation is usually modeled as a standard multivariate Gaussian distribution $p(\mathbf{x}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. Evaluating $g(\mathbf{x})$ needs to invoke a simulation to obtain the circuit performance, and $g(\mathbf{x}) \leq 0$ means the circuit fails the specification requirement (e.g., $g(\mathbf{x}) = \text{Gain}(\mathbf{x}) - 60\text{dB}$ for an Opamp). Our goal is to estimate the rare event probability represented by the integral:

$$P_r = P[\Omega] = \int_{\Omega} p(\mathbf{x})\, d\mathbf{x} = \int 1[\mathbf{x} \in \Omega]\, p(\mathbf{x})\, d\mathbf{x} \tag{1}$$

where $1[\cdot]$ represents the indicator function. The challenge lies in that P_r is exceptionally small (e.g., less than 10^{-4}) due to either Ω having an extremely small volume, or its majority being concentrated in the tail of the distribution p. In our context, the distribution p is easy to evaluate and sample from, often following a standard Gaussian distribution [20]. On the other hand, Ω is complicated and unknown in advance, while evaluating the function value $g(\cdot)$ is time-expensive. Thus, our goal is to accurately estimate P_r with as few function calls to $g(\cdot)$ as possible.

The IS approach introduces a proposal distribution $q(\cdot) \in \mathcal{P}^D$ and estimates P_r by drawing N_{IS} i.i.d. samples from q:

$$P_r^{\text{IS}} = \frac{1}{N_{\text{IS}}} \sum_{n=1}^{N_{\text{IS}}} 1[\mathbf{x}^n \in \Omega] \frac{p(\mathbf{x}^n)}{q(\mathbf{x}^n)}, \quad \mathbf{x}^n \sim q(\cdot). \tag{2}$$

It is evident that as long as the support of q includes that of p, the IS estimator remains unbiased (i.e., $\mathbb{E}_q[P_r^{\text{IS}}] = P_r$). Additionally, simple derivations demonstrate that the proposal distribution:

$$q^\star(\mathbf{x}) \propto p(\mathbf{x}) 1[\mathbf{x} \in \Omega] = \frac{1}{P[\Omega]} \cdot p(\mathbf{x}) 1[\mathbf{x} \in \Omega] \tag{3}$$

is theoretically optimal, as it can result in a zero-variance unbiased estimator [3, 4]. It is important to note that since Ω is defined by the characteristic function $g(\cdot)$ which requires expensive circuit simulations, $q^\star(\mathbf{x})$ is unknown in practice, and furthermore, direct sampling from $q^\star(\cdot)$ might not be feasible. As a result, it is common to implement the IS method by limiting the range of consideration for $q(\cdot)$ to a parametrized distribution family $\boldsymbol{Q}$ that allows for exact sampling, such as a finite mixture of Gaussian distributions [3, 14].

NFs are ideal to compose the distribution family $\boldsymbol{Q}$, due to their great expressive power and the capability to do exact density evaluation and sampling. For later simplicity, we introduce the notation $\Omega_a = \{\mathbf{x} \in \mathbb{R}^D | g(\mathbf{x}) \leq a\}$ for any $a \in \mathbb{R}$. Motivated by [1], we start from M nested subset events $\Omega_{a_1} \supsetneq \Omega_{a_2} \supsetneq \cdots \supsetneq \Omega_{a_M}$ with decreasing occurrence probabilities, which are induced by a strictly decreasing sequence $\{a_m\}_{m=1}^M$ satisfying $a_M = 0$, ensuring that $\Omega_{a_M} = \Omega$. We emphasize that the value of M and the sequence $\{a_m\}_{m=1}^M$ are both hyper-parameters of our algorithm, and we defer the empirical rules for setting them to the end of this section.

As shown in Figure 1, we exploit an NF model defined by a base distribution $q_0(\cdot)$, and MK invertible and trainable transformations $\{\mathbf{f}_i(\cdot) = \mathbf{f}(\cdot; \boldsymbol{\theta}_i) : \mathbb{R}^D \to \mathbb{R}^D\}_{i=1}^{MK}$, where $\boldsymbol{\theta}_i$ represents the i-th learnable parameters. The NF model starts from a random variable $\mathbf{z}_0 \sim q_0(\cdot)$ on the left end, and repeatedly applies each function $\mathbf{f}_i$ according to $\mathbf{z}_{i+1} = \mathbf{f}_{i+1}(\mathbf{z}_i)$. For simplicity, we denote the distribution associated with the intermediate random variable $\mathbf{z}_i$ by $q_i \in \mathcal{P}^D$. According to the change of variable theorem and the inverse function theorem, we have:

$$q_{j+1}(\mathbf{z}_{j+1}) = q_j(\mathbf{z}_j) \left| \det \left(\frac{d\mathbf{z}_j}{d\mathbf{z}_{j+1}} \right) \right| = q_j(\mathbf{z}_j) \left| \det \mathbf{J}_{\mathbf{f}_{j+1}} \right|^{-1} \tag{4}$$

where $\det(\cdot)$ denotes the determinant of a square matrix, and $\mathbf{J}_{\mathbf{f}}$ represents the Jacobian matrix of function $\mathbf{f}$. Take the logarithm of both sides in Eq. (4) and sum it by varying index j, yielding:

$$\log q_i(\mathbf{z}_i) = \log q_0(\mathbf{z}_0) - \sum_{j=1}^{i} \log |\det \mathbf{J}_{\mathbf{f}_j}|. \tag{5}$$

We use $\{\mathbf{z}_{mK}\}_{m=1}^M$ as anchor points and aim to transform their associated distributions $\{q_{mK}\}_{m=1}^M$ into effective proposal distributions for estimating the probabilities of the M nested subset events $\{P[\Omega_{a_m}]\}_{m=1}^M$. Our key motivation is that we have the freedom to make the distinction between Ω_{a_m} and $\Omega_{a_{m+1}}$ to be small. Consequently, the shift from q_{mK} to $q_{(m+1)K}$ is also expected to be marginal and to be easily learned by the NFs through K function transformations $\{\mathbf{f}_{mK+i}\}_{i=1}^K$. In the following, we describe an M-step training process, where the m-th step aims to train q_{mK}.

2.1 Step 1: Training q_K Associated with Ω_{a_1}

Let us for now ignore all components after $\mathbf{z}_K$ in Figure 1 and focus on training $\{\mathbf{f}_i\}_{i=1}^K$ to produce q_K as an effective proposal distribution for estimating the probability $P[\Omega_{a_1}]$. As the data generating distribution p in our concerned problem is easy to evaluate and sample from, we could take it as the NF's base distribution, i.e., $q_0 = p$. To begin with, we modulate the data generating distribution p to produce a distribution $p_1^\tau \in \mathcal{P}^D$:

$$p_1^\tau(\mathbf{x}) = \frac{1}{Z} e^{\min(\tau(a_1 - g(\mathbf{x})), 0)}\, p(\mathbf{x}) \tag{6}$$

where $\tau > 0$ is a temperature hyper-parameter, and Z is a normalization constant ensuring valid distribution. Recall that the condition $g(\mathbf{x}) > a_1$ is equivalent to $\mathbf{x} \notin \Omega_{a_1}$, we can understand that p_1^τ essentially compresses the height of $p(\mathbf{x})$ when $\mathbf{x}$ lies outside the set Ω_{a_1}, and the extent of this compression is determined by the margin between $g(\mathbf{x})$ and a_1. Next, we use p_1^τ as a target to learn a proposal distribution that allows for easy sampling. Noticing that any distribution defined in the NF model (such as the one we consider here, q_K) is easy to sample from, we minimize the following KL divergence loss to drive q_K to be close to p_1^τ:

$$\begin{aligned} D[q_K || p_1^\tau] &= \int q_K(\mathbf{z}_K) \log \frac{q_K(\mathbf{z}_K)}{p_1^\tau(\mathbf{z}_K)}\, d\mathbf{z}_K \approx \frac{1}{N} \sum_{n=1}^{N} \log \frac{q_K(\mathbf{z}_K^n)}{p_1^\tau(\mathbf{z}_K^n)} \\ &\propto -\frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{K} \log |\det \mathbf{J}_{\mathbf{f}_j}^n| - \frac{1}{N} \sum_{n=1}^{N} \log p_1^\tau(\mathbf{f}_{K:1}(\mathbf{z}_0^n)) \end{aligned} \tag{7}$$

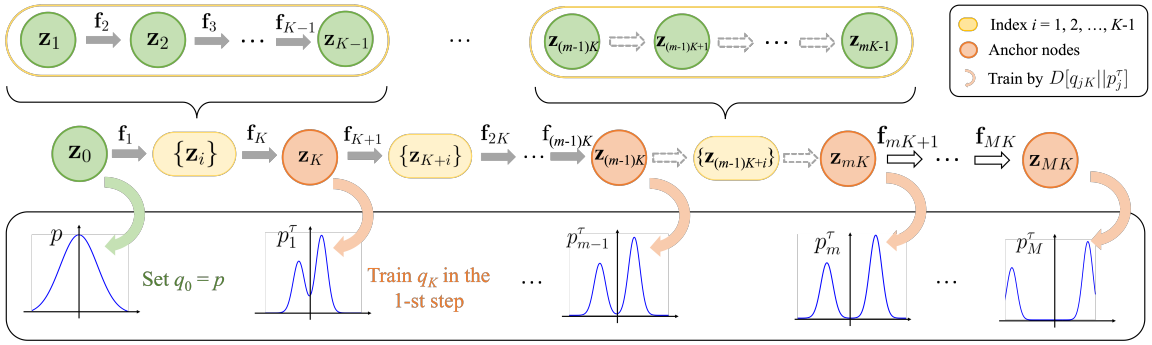


Figure 1: An illustration of our proposed NOFIS approach. Nodes $\{\mathbf{z}_{jK}\}_{j=1}^M$ along the normalizing flow highlighted in orange serve as anchor points. The distributions $\{q_{jK}\}_{j=1}^M$ associated with these nodes will be learned to align with the constructed target distributions $\{p_j^\tau\}_{j=1}^M$, achieved by adjusting the functions $\{\mathbf{f}_i\}_{i=1}^{MK}$. When learning q_{mK}, the gray-filled arrows represent frozen functions, the gray dashed-line arrows are learnable, while the gray solid-line arrows are yet to be trained.

where $\mathbf{z}_0^n$ is a sample drawn from p. To derive the last line, we use Eqs. (4-5), $q_0 = p$ and the short notation $\mathbf{z}_K^n = \mathbf{f}_{K:1}(\mathbf{z}_0^n) = \mathbf{f}_K \circ \mathbf{f}_{K-1} \circ \cdots \circ \mathbf{f}_1(\mathbf{z}_0^n)$, and omit those terms don't depend on the learnable functions $\{\mathbf{f}_i\}_{i=1}^K$. Note that the normalization constant Z in p_1^τ is not needed in the computation, as it will appear as a constant $\log Z$ in Eq. (7) which won't affect training.

Several important clarifications must be made. Firstly, the NFs utilize specific network architectures to parameterize $\mathbf{f}_i(\cdot)$ as $\mathbf{f}(\cdot;\boldsymbol{\theta}_i)$. It is crucial to meticulously design the form of $\mathbf{f}(\cdot;\boldsymbol{\theta}_i)$ [5, 6], to ensure that the evaluation of the determinant of its Jacobian matrix, as required by Eq. (7), is straightforward. Secondly, we have the option to employ the learned q_K for estimating $P[\Omega_{a_1}]$ by incorporating it with the IS approach. However, we won't pursue it as our sole objective is the final rare event probability $P[\Omega_{a_M}] = P[\Omega]$. Namely, learning q_K is for ease of learning subsequent distributions such as q_{2K}, q_{3K}, and ultimately q_{MK}. Thirdly, since when $a_1 \to +\infty$, $P[\Omega_{a_1}] \to 1.0$. We can freely choose a_1 such that $P[\Omega_{a_1}]$ is not too small (e.g., greater than 0.1) to ensure an adequate number of samples $\mathbf{z}_K^n$ are located within Ω_{a_1}.

Fourthly, based on Eq. (3), we know that the theoretically optimal proposal distribution for estimating $P[\Omega_{a_1}]$ equals $p(\mathbf{x})\mathbb{1}[\mathbf{x} \in \Omega_{a_1}]/P[\Omega_{a_1}]$. For convenience, we denote this best proposal as p_1^∞, since it is the limit of p_1^τ when $\tau \to \infty$. It seems appealing to use p_1^∞ as the target in Eq. (7) instead of p_1^τ. However, we observe that it brings severe training issues in practice. This can also be explained in theory — if there exists a sample $\mathbf{z}_K^n = \mathbf{f}_{1:K}(\mathbf{z}_0^n)$ located outside Ω_{a_1}, then $p_1^\infty(\mathbf{f}_{1:K}(\mathbf{z}_0^n))$ strictly equals zero, rendering the training loss undefined. On the other hand, if all sampled $\mathbf{z}_K^n$'s locate inside Ω_{a_1}, then we actually drive q_K to the data generating distribution p because $p_1^\infty(\mathbf{f}_{1:K}(\mathbf{z}_0^n)) \propto p(\mathbf{f}_{1:K}(\mathbf{z}_0^n))$ holds true for all n and the normalization constant doesn't matter when training with Eq. (7).

2.2 Step $2 \sim M$: Training q_{mK} by Freezing $q_{(m-1)K}$

Once the successful learning of q_K is achieved through the training of $\{\mathbf{f}_i\}_{i=1}^K$ using the approach discussed in the previous subsection, we could train $\{\mathbf{f}_{K+i}\}_{i=1}^K$ to learn a subsequent q_{2K} working as a proposal distribution for Ω_{a_2} similarly by minimizing $D[q_{2K}||p_2^\tau]$. To facilitate our discussion, we will describe a general m-th step, where m is any integer between 2 and M. At the beginning of the m-th step, all functions $\{\mathbf{f}_i\}_{i=1}^{(m-1)K}$ are trained such that q_{jK} is an effective proposal distribution associated with Ω_{a_j}, for any $j = 1, 2, \cdots, m-1$. Our goal in this step is to train $\{\mathbf{f}_{(m-1)K+i}\}_{i=1}^K$ to enforce q_{mK} working as an effective proposal distribution for Ω_{a_m}. Similar to Eq. (6) and (7), we use the following training loss:

$$D[q_{mK}||p_m^\tau] \propto -\frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{mK}\log|\det \mathbf{J}_{\mathbf{f}_j}^n| - \frac{1}{N}\sum_{n=1}^{N}\log p_m^\tau(\mathbf{f}_{mK:1}(\mathbf{z}_0^n)) \tag{8}$$

where $\mathbf{z}_0^n \sim p(\cdot)$ and $p_m^\tau \in \mathcal{P}^D$ is a constructed target distribution:

$$p_m^\tau(\mathbf{x}) = \frac{1}{Z}e^{\min(\tau(a_m - g(\mathbf{x})),0)}\,p(\mathbf{x}). \tag{9}$$

When minimizing Eq. (8), the functions $\{\mathbf{f}_i\}_{i=1}^{(m-1)K}$ will be held constant (as indicated by the gray-filled arrows in Figure 1). Our focus will solely be on training the functions $\{\mathbf{f}_{(m-1)K+i}\}_{i=1}^K$, which are represented by the gray dashed-line arrows in Figure 1. Recall that q_{mK} is related to $q_{(m-1)K}$ through the learnable transformations $\{\mathbf{f}_{(m-1)K+i}\}_{i=1}^K$ and that the distribution $q_{(m-1)K}$ has already been well calibrated matching to $\Omega_{a_{m-1}}$. Consequently, there is no compelling reason to further train the previous $\mathbf{f}_i$'s (where $i \le (m-1)K$) in the m-th step, as $\{\mathbf{f}_{(m-1)K+i}\}_{i=1}^K$ alone possess ample expressive power to capture the distribution shift from p_{m-1}^τ to p_m^τ effectively.

Summary. Algorithm 1 outlines the major steps of the proposed NOFIS approach for rare circuit failure estimation. It is worth mentioning that the NOFIS method necessitates a total of $(MEN + N_{\text{IS}})$ function calls to $g(\cdot)$. We empirically find that NOFIS is suitable to estimate $P_r \le 10^{-4}$, otherwise, the advantages of NOFIS over MC may be limited given the same function call budget. We will provide a quantitative explanation in the numerical result section.

Firstly, to estimate probabilities $P_r \approx 10^{-x}$ (where x is a positive integer), we empirically find that choosing M equals x is adequate. This observation aligns with previous experiences [1, 19]. As a rule of thumb, $\{a_m\}_{m=1}^M$ should approximately make the elements in $\{P[\Omega_{a_m}]\}_{m=1}^M$ scaled by 0.1 in order.

Secondly, regarding the temperature hyper-parameter τ, let us consider two points $\mathbf{x} \in \Omega_{a_m}$ and $\mathbf{x}' \notin \Omega_{a_m}$. Then our constructed p_m^τ should satisfy the constraint: $p_m^\tau(\mathbf{x}) \ge p_m^\tau(\mathbf{x}')$ for it to be meaningful as a target. Substituting the expression of p_m^τ as shown

in Eq. (9) into this inequality results in a lower bound on τ. Moreover, as we discussed in the fourth remark in Section 2.1, τ cannot be excessively large either. For more numerical results, please refer to the ablation studies in Section 3.2.

Finally, if our sole objective is to estimate $P[\Omega_{a_1}]$ which is around 0.1, we don't need learning at all. Instead, we could perform MC sampling from p. However, when being applied to estimate $P_r = P[\Omega_{a_M}]$, MC would likely yield a trivial estimate of $P_r = 0$ because all generated samples lie outside Ω_{a_M}. Essentially, our NOFIS approach attempts to simplify estimating $P_r = P[\Omega_{a_M}]$ by memorizing $\Omega_{a_{m-1}}$ and its associated p^{τ}_{m-1} through $q_{(m-1)K}$ in the NFs. This enables the subsequent learning of Ω_{a_m} to become manageable, because Ω_{a_m} is chosen to only have minor change from $\Omega_{a_{m-1}}$, and sampling from $q_{(m-1)K}$ is tractable due to NFs.

Algorithm 1 NOFIS

1: Provide a data generating distribution $p \in \mathcal{P}^D$ and an integral region $\Omega = \{\mathbf{x} \in \mathbb{R}^D | g(\mathbf{x}) \leq 0\}$.
2: Define a NF characterized by a base distribution $q_0 = p$, and a series of invertible transformations $\{\mathbf{f}_i(\cdot) = \mathbf{f}(\cdot; \boldsymbol{\theta}_i) : \mathbb{R}^D \to \mathbb{R}^D\}_{i=1}^{MK}$.
3: Choose hyper-parameters: (i) a strictly decreasing sequence $\{a_m\}_{m=1}^{M}$ satisfying $a_M = 0$, and (ii) the temperature hyper-parameter $\tau > 0$.
4: **for** $m = 1$ to M **do**
5: If $m \geq 2$, freeze $\{\boldsymbol{\theta}_i\}_{i=1}^{(m-1)K}$.
6: **for** $e = 1$ to E **do**
7: Draw N samples $\{\mathbf{z}_0^n\}_{n=1}^{N}$ from the base q_0.
8: Calculate the loss $D[q_{mK}||p^{\tau}_m]$ using Eq. (8).
9: Perform backward propagation and update the model parameters $\{\boldsymbol{\theta}_{(m-1)K+i}\}_{i=1}^{K}$.
10: **end for**
11: **end for**
12: Return P_r^{IS} using the learned q_{MK} as the proposal distribution based on Eq. (2).

3 NUMERICAL RESULTS

As justified in Section 2, we set $p = \mathcal{N}(\mathbf{0}, \mathbf{I})$ for all of our numerical experiments and utilize RealNVP [6] as the backbone NF model. In the subsequent Section 3.1, we present visualizations of several 2D test cases, assuming an unlimited number of function calls to $g(\cdot)$. Its primary objective is to qualitatively justify that our NOFIS approach can learn a q_{MK} fully recovering the optimal proposal distribution, in an ideal scenario where there is *no limit on function calls*. Conversely, *the limited function call* scenario represents the practical situation when deploying the algorithm. We quantitatively evaluate NOFIS's performance in Section 3.2 under this restricted scenario. Our algorithm is implemented in Pytorch and runs on a Linux cluster with V100 GPUs. Our source code is available at: https://github.com/zhengqigao/NOFIS-DAC24/.

3.1 Qualitative Analysis

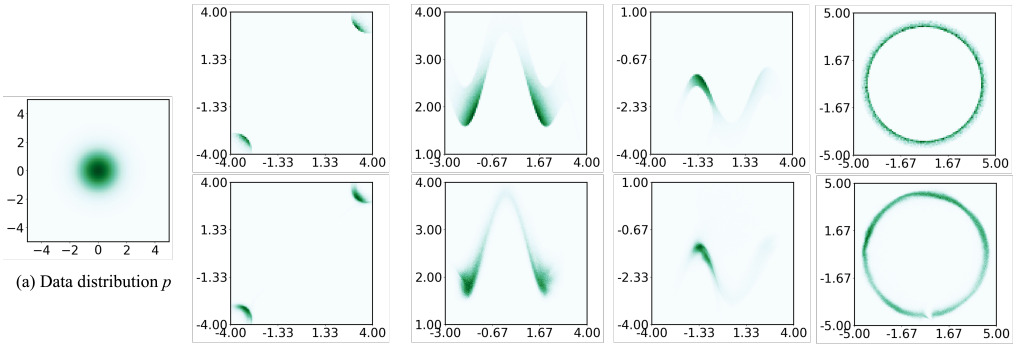
We deliberately design several functions $g(\mathbf{x})$ in 2D to make the integral region Ω to possess different shapes and locate at the tail of p. Figure 2 shows the learned q_{MK} in these cases. Taking Figure 2 (b) as an example, we consider the integral region $\Omega = \{(x_1, x_2) \,|\, g(x_1, x_2) \leq 0\}$, where $g(x_1, x_2) = \min[(x_1 + 3.8)^2 + (x_2 + 3.8)^2, (x_1 - 3.8)^2 + (x_2 - 3.8)^2] - 1$. The best proposal distribution $q^{\star}$ defined in Eq. (3) is shown in the top row of Figure 2 (b). It is evident that $q^{\star}$ lies at the tail of the original data generating distribution p. Directly using an NF model to learn this $q^{\star}$ is not feasible due to numerical issues in training.

We set $K = 8$ and $M = 5$ in our NOFIS approach, so q_8, q_{16}, q_{24}, q_{32}, and q_{40} will be taken as anchors matching to p_1^{τ}, p_2^{τ}, p_3^{τ}, p_4^{τ}, and p_5^{τ}. To further justify our approach, we visualize intermediate distributions $\{q_8, q_{16}, q_{24}, q_{32}\}$ in Figure 3 (a)-(d), while q_{40} is already displayed in the bottom row of Figure 2 (b). The region Ω_{a_m} induced by a_m encompasses two circles centered at $(\pm 3.8, \pm 3.8)$ with a radius of $\sqrt{a_m + 1}$. According to Eq. (3), the heatmap of the optimal proposal distribution for estimating $P[\Omega_{a_m}]$ corresponds to "modulating/coloring" Ω_{a_m} based on the magnitude of p, resulting two thin leaf shape as exemplified in the top row of Figure 2 (b). Furthermore, as a_m decreases alongside m, the radius also decreases, leading to a gradual outward shift of the two thin leaves from the origin. This phenomenon could indeed be observed in Figure 3 (a)-(d). Moreover, $\{a_1, a_2, a_3, a_4, a_5\}$ are set to $\{26, 15, 8, 3, 0\}$ in this case, and the radii of the learnt leaf shapes in Figure 2 (a)-(d) are surely consistent with the expression $\sqrt{a_m + 1}$. Last but not least, training loss curves are plotted in Figure 3 (e).

3.2 Quantitative Experiments

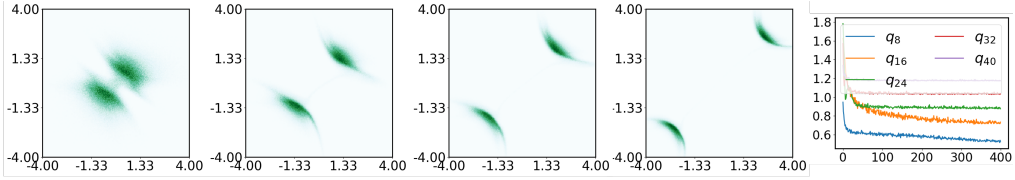
To substantiate the effectiveness of the proposed NOFIS approach, we assess its performance across 10 test cases outlined in Table 1. These cases include five synthetic scenarios (#1-#5), three instances featuring real circuit examples (#6: Opamp [22], #8: Charge Pump [9], #9: Photonic Y-branch under process variation, one involving a physical oscillator (#7) [18] subjected to position variation, and the remaining one illustrating the performance degradation of ResNet18 under parameter variation (#10). Due to page limit, here we briefly describe the settings for the three circuit examples. For the Opamp and Charge Pump, the width of the MOS transistors in the circuits follow a standard Gaussian distribution due to process variation, and the integral region Ω represents Gain smaller than 72 dB in Opamp and current mismatch at the output larger than 370 uA in Charge Pump, respectively. For the Y-branch example, $\mathbf{x}$ represents the random boundary deformation, and Ω represents the power transmission smaller than 32%. Generally, in our experiments, we set E to $15 \sim 20$, N to $100 \sim 400$, N_{IS} to $20 \sim 5000$, M to $4 \sim 6$, and τ to $10 \sim 30$. Specific values vary depending on the test cases; for example, in the #2 Cube test case, E, M, and N need to be larger as the target P_r is extremely small.

Two evaluation metrics are considered: (i) the number of function calls and (ii) the prediction error measured in the logarithm. The golden P_r is obtained by a large number of MC samples for all cases, except for the Cube test case (#2) where an analytical solution exists. Furthermore, we implement the following six baseline methods for comparison purposes: (i) MC: The conventional Monte Carlo method [4]. (ii) SIR: Simple regression. A deep neural network is first trained to learn the mapping $g(\mathbf{x})$ using N samples. Afterwards, N_{eval} samples (e.g., $N_{\text{eval}} = 10^9$) are generated from the distribution p, and their function values are evaluated using the neural network. The rare event probability estimation involves



(a) Data distribution p (b) (c) (d) (e)

Figure 2: (a) The heatmap represents the data generating distribution $p = \mathcal{N}(0, \mathrm{I})$. (b)-(e) The top row displays the theoretically optimal proposal distribution $q^\star$ defined in Eq. (3), while the bottom row illustrates the learned proposal distribution q_{MK} generated by the NFs using Algorithm 1. They exhibit a strong alignment in every case. When we overlay (b)-(e) onto (a), we notice that the region Ω's highlighted by green in (b)-(e) occur at the tail of p in (a).



(a) (b) (c) (d) (e)

Figure 3: (a)-(d) The intermediate distributions $\{q_8, q_{16}, q_{24}, q_{32}\}$ of the NF model are plotted. They have been successfully trained, and the highlighted regions are centered at $(\pm 3.8, \pm 3.8)$ with radii that match our expected expression $\sqrt{a_m + 1}$. (e) The training loss in each step is plotted against the epoch. For better visualization, the Y-axis is presented on a logarithmic scale.

Table 1: Results on synthetic (#1-#5) and real-world (#6-#10) experiments averaged from 20 repeated runs are reported in the format 'number of calls / logarithm error'. Here 'K' represents one thousand, and '—' indicates algorithm failure.

	Dim.	Golden P_r	MC	SIR	SUC	SUS	SSS	Adapt-IS	NOFIS (ours)
(#1) Leaf	2	4.74E-6	50.0K / 9.11	50.0K / 9.30	47.5K / 4.79	42.0K / 0.23	40.0K / 0.70	35.0K / 0.25	32.0K / 0.11
(#2) Cube	6	2.15E-9	500K / 11.33	500K / 10.62	279.9K / 7.28	206.0K / 0.096	400.0K / 1.53	227.0K / 6.23	197.5K / 0.078
(#3) Rosen	10	4.69E-4	7.0K / 1.87	7.0K / 0.96	8.3K / 0.85	7.0K / 0.40	8.0K / 0.46	8.4K / 15.07	7.0K / 0.32
(#4) Levy	20	3.70E-6	50.0K / 11.80	50.0K / 14.56	50.0K / 4.31	49.0K / 0.53	—	56.0K / 9.20	48.2K / 0.44
(#5) Powell	40	3.15E-5	10.0K / 11.0	10.0K / 3.66	9.6K / 3.52	9.0K / 5.80	8.0K / 0.84	7.9K / 15.56	7.0K / 0.38
(#6) Opamp	5	1.30E-5	100K / 5.4	50K / 3.63	49K / 3.58	45K / 0.08	60K / 0.85	48K / 2.89	45K / 0.07
(#7) Oscillator	6	1.81E-6	100K / 13.58	50K / 0.24	40.1K / 4.33	45K / 0.13	40K / 1.17	43K / 2.62	31K / 0.12
(#8) Charge Pump	16	5.75E-6	100K / 8.27	100K / 8.73	50.5K / 3.66	45K / 0.15	40K / 1.31	43K / 12.77	35K / 0.12
(#9) Y-branch	26	4.27E-5	50K / 2.52	50K / 4.18	23.9K / 2.84	35.0K / 0.18	40K / 0.30	43K / 15.28	32.5K / 0.11
(#10) ResNet18	62	6.00E-5	20K / 4.16	20K / 8.13	22.9K / 3.62	20K / 0.55	20K / 3.12	—	18K / 0.61

calculating the ratio of how many of these N_{eval} samples fall within Ω. (iii) SUS: Subset simulation [1, 19]. (iv) SUC: Subset classification. In short, the MCMC sampling in SUS is replaced with modern deep neural networks. (v) SSS: Scaled-sigma sampling [20]. (vi) Adapt-IS: Adaptive importance sampling [4, 14].

Table 1 reports the rare event estimation results on all test cases. NOFIS attains the lowest error while requiring the fewest function calls across all examples, except for the last ResNet18 case where it performs slightly worse than SUS. Taking the case (#1) Leaf as an example, our NF model is trained using $M = 4$ steps, $E = 20$ epochs, and a batch size of $N = 400$, resulting in a total of $MEN = 32000$ function calls. Additionally, generating the IS estimator requires extra $N_{IS} = 20$ function calls in the end. The left part of Figure 4 showcases the learned proposal distribution q_{MK}, and the right part further reveals that when increasing N_{IS}, the estimation could become even more accurate. It is worth noting that the Leaf test case here is precisely the one depicted in Figure 2 (b). Comparing the left part of Figure 4 to the lower part of Figure 2 (b), we conclude limiting the number of function calls leads to a degradation in the learned proposal distribution, but NOFIS still successfully captures the two-leaf shape and generates highly accurate probability estimates.

Lastly, we examine the effects of various implementation choices on the performance of NOFIS using the circuit examples — Opamp, Charge Pump (CP), and Y-branch. The results previously reported in Table 1 are labeled as the "nominal" configuration. The left segment of Figure 5 displays the prediction error when a single incremental

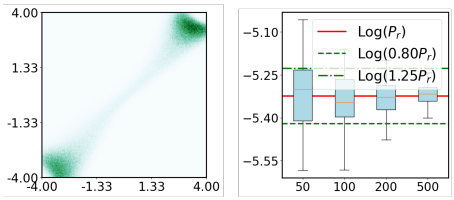


Figure 4: Although limiting the number of function calls leads to degradation in the learned distribution, NOFIS still generates accurate estimates. Left: The learned q_{MK} for Case (#1) in a single run with 32K function calls. Right: Utilize this acquired q_{MK} to generate an IS estimator with varying N_{IS}. The X-axis and Y-axis denote N_{IS} and logarithm probability.

change is applied to the nominal setup. For the 'LongThre' parameter, we set $M = 9$, and for 'SmallTemp', we use $\tau = 1$, whereas the nominal settings have $M \in [4, 6]$ and $\tau \in [10, 30]$. It is noteworthy that altering the freezing approach, using extended threshold sequences, or employing smaller temperatures doesn't consistently lead to improvements in NOFIS performance. Moreover, the right part of Figure 5 uncovers two significant observations: (i) NOFIS demonstrates great robustness within the temperature range of $\tau \in [10, 200]$, and (ii) a carefully tuned temperature τ could potentially yield **even better outcomes** for the proposed NOFIS method. For example, the optimal results (depicted by the lowest markers) on the red Opamp, blue CP, and green Y-branch curves in the right section of Figure 5 achieve prediction errors of 0.026, 0.054, and 0.023, respectively. These estimation errors are considerably smaller than their counterparts (i.e., 0.07, 0.12, and 0.11) reported in Table 1, while utilizing the same number of function calls.

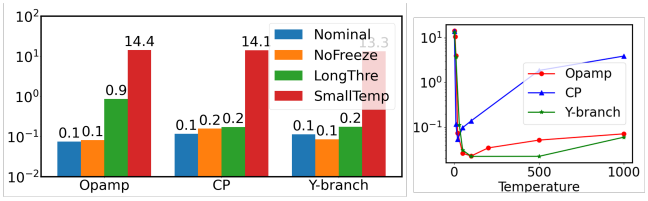


Figure 5: Left: Ablation studies are carried out on non-freezing, long threshold sequences, and small temperature τ. Right: NOFIS's error is plotted versus τ. The Y-axis in both figures represents the logarithm prediction error.

4 CONCLUSIONS AND LIMITATIONS

In this paper, we introduce NOFIS, an efficient method for estimating rare event probabilities through normalizing flows. NOFIS learns a sequence of functions to shift a base distribution towards an effective proposal distribution, using nested subset events as bridges. Our qualitative analysis underscores NOFIS's adeptness in accurately recovering the optimal proposal distribution. Our quantitative exploration across 10 test cases justifies NOFIS's superiority over six baseline methods. The effectiveness of NOFIS hinges on accurately configuring nested subset events. Yet, the prevailing approach, both in this work and previous studies [1, 19], entails human intervention. Developing an automated method for defining nested subset events stands as a crucial avenue for future research.

REFERENCES

[1] Siu-Kui Au and James L Beck. 2001. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic engineering mechanics* 16, 4 (2001), 263–277.

[2] A.J. Bhavnagarwala, Xinghai Tang, and J.D. Meindl. 2001. The impact of intrinsic device fluctuations on CMOS SRAM cell stability. *IEEE Journal of Solid-State Circuits* 36, 4 (2001), 658–665.

[3] Gino Biondini. 2015. An introduction to rare event simulation and importance sampling. In *Handbook of Statistics*. Vol. 33. Elsevier, 29–68.

[4] James Antonio Bucklew and J Bucklew. 2004. *Introduction to rare event simulation*. Vol. 5. Springer.

[5] Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516* (2014).

[6] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using Real NVP. In *International Conference on Learning Representations*.

[7] Lara Dolecek, Masood Qazi, Devavrat Shah, and Anantha Chandrakasan. 2008. Breaking the simulation barrier: SRAM evaluation through norm minimization. In *2008 IEEE/ACM International Conference on Computer-Aided Design*. IEEE, 322–329.

[8] Zhengqi Gao, Jun Tao, Yangfeng Su, Dian Zhou, Xuan Zeng, and Xin Li. 2020. Efficient Rare Failure Analysis Over Multiple Corners via Correlated Bayesian Inference. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 10 (2020), 2029–2041.

[9] Zhengqi Gao, Jun Tao, Fan Yang, Yangfeng Su, Dian Zhou, and Xuan Zeng. 2019. Efficient Performance Trade-off Modeling for Analog Circuit based on Bayesian Neural Network. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–8.

[10] Rouwaida Kanj, Rajiv Joshi, and Sani Nassif. 2006. Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events. In *Proceedings of the 43rd Design Automation Conference*. 69–72.

[11] Kentaro Katayama, Shiho Hagiwara, Hiroshi Tsutsui, Hiroyuki Ochi, and Takashi Sato. 2010. Sequential importance sampling for low-probability and high-dimensional SRAM yield analysis. In *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 703–708.

[12] S. Mukhopadhyay, H. Mahmoodi, and K. Roy. 2004. Statistical design and optimization of SRAM cell for yield enhancement. In *IEEE/ACM International Conference on Computer Aided Design*. 10–13.

[13] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. 2021. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research* 22, 1 (2021), 2617–2680.

[14] Xiao Shi, Fengyuan Liu, Jun Yang, and Lei He. 2018. A fast and robust failure analysis of memory circuits using adaptive importance sampling method. In *Proceedings of the 55th Design Automation Conference*. 1–6.

[15] Xiao Shi, Hao Yan, Qiancun Huang, Jiajia Zhang, Longxing Shi, and Lei He. 2019. Meta-Model Based High-Dimensional Yield Analysis Using Low-Rank Tensor Approximation. In *Proceedings of the 56th Design Automation Conference 2019*.

[16] Xiao Shi, Hao Yan, Chuwen Li, Jianli Chen, Longxing Shi, and Lei He. 2020. A Non-Gaussian Adaptive Importance Sampling Method for High-Dimensional and Multi-Failure-Region Yield Analysis. In *Proceedings of the 39th International Conference on Computer-Aided Design*.

[17] Amith Singhee and Rob A. Rutenbar. 2009. Statistical Blockade: Very Fast Statistical Simulation and Modeling of Rare Circuit Events and Its Application to Memory Design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28, 8 (2009), 1176–1189.

[18] Jingwen Song, Pengfei Wei, Marcos Valdebenito, and Michael Beer. 2021. Active learning line sampling for rare event analysis. *Mechanical Systems and Signal Processing* 147 (2021), 107113.

[19] Shupeng Sun and Xin Li. 2014. Fast statistical analysis of rare circuit failure events via subset simulation in high-dimensional variation space. In *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 324–331.

[20] Shupeng Sun, Xin Li, Hongzhou Liu, Kangsheng Luo, and Ben Gu. 2015. Fast statistical analysis of rare circuit failure events via scaled-sigma sampling for high-dimensional variation space. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34, 7 (2015), 1096–1109.

[21] Jun Tao, Handi Yu, Dian Zhou, Yangfeng Su, Xuan Zeng, and Xin Li. 2017. Correlated rare failure analysis via Asymptotic Probability Evaluation. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6.

[22] Zushu Yan, Pui-In Mak, Man-Kay Law, and Rui Martins. 2012. A 0.016mm2 144uW three-stage amplifier capable of driving 1-to-15nF capacitive load with >0.95MHz GBW. In *2012 IEEE International Solid-State Circuits Conference*. 368–370.

[23] Handi Yu, Jun Tao, Changhai Liao, Yangfeng Su, Dian Zhou, Xuan Zeng, and Xin Li. 2016. Efficient statistical analysis for correlated rare failure events via Asymptotic Probability Approximation. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–8. https://doi.org/10.1145/2966986.2967029