RSizing: Robust Bayesian Optimization for Analog Circuit Sizing Under Process Variations

Jindong Tu[1], Yapeng Li[1], Peng Xu[2], Tuo Li[3], Guoqing Li[3*], Zushuai Xie[4*], Bei Yu[2], Tinghuan Chen[1*]

[1]The Chinese University of Hong Kong, Shenzhen [2]The Chinese University of Hong Kong

[3]Shandong Yunhai Guochuang Cloud Computing Equipment Industry Innovation Co., Lt

[4]Nanjing University of Posts and Telecommunications

***Abstract*—The increasing complexity of CMOS technology and circuit designs has intensified the need for robust analog design automation tools that can handle process variations effectively. This paper presents RSizing, a novel approach for analog circuit sizing that optimizes performance while ensuring robustness against process variations. Our method employs a three-phase strategy: First, it identifies promising design regions through nominal condition optimization to prune the design space efficiently. Second, it performs variation-aware optimization using heteroscedastic Gaussian processes (HGP) to model circuit performance under process variations, capturing the non-uniform nature of process-induced fluctuations across the design space. The HGP models are combined with an efficient acquisition function based on Thompson sampling to guide the exploration of robust designs using limited Monte Carlo simulations. Finally, it refines the solutions through additional targeted Monte Carlo simulations and model calibration. Experimental results on three benchmark circuits demonstrate that RSizing achieves superior performance compared to existing methods, consistently meeting yield requirements while optimizing multiple performance metrics with significantly reduced computational cost.**

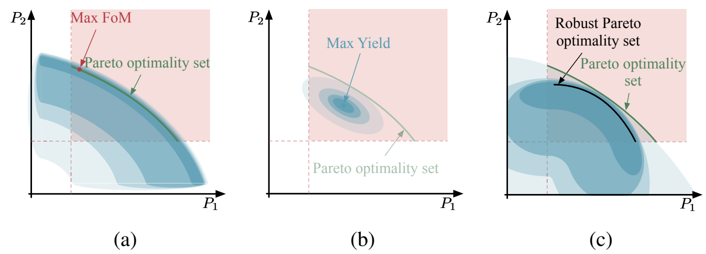


Fig. 1 The difference among typical sizing, yield optimization, and our task: (a) Typical Sizing; (b) Yield Optimization; (c) Our Task. The blue is the probability density of performance variation under process variation. The darker the color, the greater the density. The pink is the region in which all designs satisfy performance specifications.

I. Introduction

As scaling advances, the design of integrated circuits (ICs) becomes increasingly complex and challenging [1]. This evolution necessitates more sophisticated technologies and methodologies to effectively manage higher densities, enhanced functionality, and yield within smaller form factors. While digital circuit design has been largely automated, the design of analog circuits still relies predominantly on manual processes. Given the rising demands for high performance and yield coupled with the need for rapid market entry, manual analog circuit design is becoming increasingly impractical. Therefore, there is an urgent need for automation tools in analog circuit design.

Significant efforts have been dedicated to improving analog circuit sizing in analog circuit design automation tools. The primary goal of sizing is to maximize performance by optimizing device design parameters. This task can be framed as a multi-objective optimization problem aimed at identifying an optimal solution that corresponds to the maximum Figure of Merit (FoM) [2] or the Pareto optimality set [3], as shown in Fig. 1(a). FoM is the weighted sum of all objectives. However, enhancing one performance metric often requires sacrificing another. The Pareto optimality set represents a set of solutions that offer the best trade-offs among all objectives, where no solution is dominated by any other within the feasible solution space. The optimal design parameter vector lies on the Pareto optimality set in Fig. 1(a).

Typical analog circuit sizing methods can be categorized into model-based [4] and simulation-based approaches [2], [3], [5], [6]. Model-based methods require the development of an analytical model that relates device design parameters to performance. However, accurately deriving such performance models can be challenging. As a result, many researchers have focused on simulation-based methods, where performances are represented by black-box functions—known as surrogate models—such as Gaussian processes, which are built and refined through on-the-fly simulations [3]. Using these black-box functions, techniques like evolutionary algorithms [7], [8], Bayesian optimization [2], [3], and reinforcement learning [5], [6] can be employed to identify the Pareto optimality set and optimal solutions corresponding to the maximum FoM. However, traditional analog circuit sizing approaches may lead to low yield, as design parameters often do not account for process variations, as illustrated in Fig. 1(a).

Some studies focus on yield optimization to enhance overall yield. Unlike typical analog circuit sizing, the yield optimization task aims to maximize yield by fine-tuning device design parameters [9]–[11], as illustrated in Fig. 1(b). Existing methods can be categorized into corner-based [12], response-surface-based [10], and Monte Carlo (MC)-based [11] approaches. MC-based methods are the most widely used due to their high accuracy and versatility [11]. But they still require numerous simulations to meet optimization demands. To address this, surrogate models combined with importance sampling have been proposed to reduce the number of required simulations [11], [13]. However, optimizing for yield alone can compromise circuit performance.

Several studies have focused on optimizing circuit performance under process, voltage, and temperature (PVT) corners to enhance design robustness [14], [15]. Shi and Kong formulate the robust analog sizing problem as a constraint satisfaction challenge to identify a circuit sizing vector that meets performance specifications across all PVT corners [14], [15]. However, similar to typical yield optimization approaches, their methods do not guarantee optimized circuit performance. Cao et al. reformulate robust analog sizing as a parameter-to-specification optimization problem, aiming to minimize the difference between target performance and actual circuit performance across PVT corners [16], [17]. Achieving an optimal trade-off between

*Corresponding authors

circuit performance and yield remains challenging. A high design target can lead to low yield, even in the absence of feasible solutions. In [18], circuit performance is maximized under all PVT corners using a customized evolutionary algorithm. However, this approach suffers from low optimization efficiency. Furthermore, existing robust analog sizing methods rely on PVT corner simulations, which do not account for all process variations. Consequently, the sizing solutions may not consistently meet yield requirements, particularly for analog circuits that are more sensitive to process variations.

Recently, robust multi-objective optimization under input noise has been proposed in the machine learning area [19]. The main idea is that the surrogate models predict uncertainty at unexplored design points, enabling efficient identification of robust solutions that maintain good performance under input variations. However, this approach requires prior knowledge of the uncertainty distribution and assumes the uncertainty can be modeled as input noise, which is not applicable to process variation aware analog circuit sizing.

Unlike previous works, we propose RSizing to find a high-quality Pareto optimality set to satisfy yield constraints with unknown distribution process variations. To the best of our knowledge, this is the first work to conduct analog circuit sizing with a fixed yield constraint. The key contributions are summarized as follows:

- We propose an efficient three-phase algorithm that guarantees the required yield while optimizing circuit performance.
- We perform variation aware optimization using surrogate models to model MC simulation data and predict performance variations across the design space. These models guide the exploration towards high performance while ensuring yield requirements.
- Experimental results on three benchmark circuits demonstrate that RSizing achieves superior performance compared to existing methods, consistently meeting yield requirements while optimizing multiple performance metrics with significantly reduced computational cost.

II. Preliminaries

A. Problem Formulation

In this paper, we conduct analog circuit sizing with a given yield constraint. Considering multiple performances to be optimized and avoiding enhancing one performance metric with sacrificing another, like previous works [3], [18], we formulate the analog circuit sizing as a multi-objective optimization problem.

Let $f^{(1)}(\boldsymbol{x}), f^{(2)}(\boldsymbol{x}), \cdots, f^{(M)}(\boldsymbol{x})$ denote the nominal performance metrics, where $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$ is the design parameters, $\mathcal{X}$ is the parameter space, and d is the design parameter dimension. Denote the performance vector as $\boldsymbol{f}(\boldsymbol{x}) = [f^{(1)}(\boldsymbol{x}), f^{(2)}(\boldsymbol{x}), \cdots, f^{(M)}(\boldsymbol{x})]$. These M objectives would possibly conflict with each other. Finding one solution that maximizes all of these objectives simultaneously is difficult. Practically, to strike a balance between these objectives, we want to identify the Pareto optimality set.

Definition 1 (Pareto optimality)**.** *In an M-dimension maximization problem, an objective vector $\boldsymbol{f}(\boldsymbol{x})$ is said to dominate $\boldsymbol{f}(\boldsymbol{x}')$ if*

$$\begin{aligned} &\forall i \in [1, M], f^{(i)}(\boldsymbol{x}) \geq f^{(i)}(\boldsymbol{x}') \text{ and} \\ &\exists j \in [1, M], f^{(j)}(\boldsymbol{x}) > f^{(j)}(\boldsymbol{x}'). \end{aligned} \tag{1}$$

A point $\boldsymbol{x}$ is Pareto optimality if there is no other $\boldsymbol{x}'$ in parameter space satisfying that $f(\boldsymbol{x}')$ dominates $f(\boldsymbol{x})$. In the whole parameter space, the set of points that are not dominated by others is called the Pareto optimality set. For the Pareto optimality designs, there does not exist an alternative choice that can improve every objective without sacrificing others. In multi-objective optimization problems, the realistic and accurate goal is to identify the Pareto optimality set containing all the Pareto optimal optimality points.

The quality of the Pareto optimality set is evaluated using the hypervolume (HV) [3]. Given a set of non-dominated solutions $\mathcal{P}$ and a reference point $\boldsymbol{r} \in \mathbb{R}^M$, HV is defined as the M-dimensional Lebesgue measure of the region dominated by $\mathcal{P}$ and bounded by $\boldsymbol{r}$, denoted as $\mathrm{HV}(\mathcal{P}|\boldsymbol{r})$.

To account for process variations, for each performance function $f(\boldsymbol{x})$, we consider the impact of process variations as a noise term, and use $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon$ to represent the performance. The corresponding vector of M noisy performances is denoted as $\boldsymbol{y}(\boldsymbol{x}) = [y^{(1)}(\boldsymbol{x}), ..., y^{(M)}(\boldsymbol{x})]$. Formally, we give our problem formulation as follows.

Problem 1 (Analog Circuit Sizing Under Process Variations)**.** *Given an analog circuit and required yield, find a high-quality Pareto optimality set and its design parameters to satisfy yield constraints with process variations.*

As illustrated in Fig. 1(c), our goal is to find a robust Pareto optimality set within the pink region, where the points satisfy both performance specifications and yield requirements α. In other words, the probability of their performances satisfying design specifications is not less than α, i.e., $p(\boldsymbol{y}(\boldsymbol{x}) \geq \boldsymbol{z}) \geq \alpha$.

B. Bayesian Optimization

Bayesian optimization (BO) is an effective technique for optimizing black-box functions, particularly when function evaluations are computationally expensive [20]. The BO framework operates through two key components: a probabilistic surrogate model and an acquisition function. The surrogate model provides well-calibrated uncertainty estimates of the objective function and is iteratively refined with new observation data. The acquisition function guides the optimization process by balancing the exploration of uncertain regions and the exploitation of promising areas, with each new evaluation point selected by maximizing this function.

However, BO often faces efficiency challenges in high-dimensional spaces due to excessive exploration. Trust region-based BO [21], [22] addresses these limitations by constraining the search within well-defined local trust regions. Each trust region (TR) is characterized by a hyperrectangle with a center point and specific edge length, along with its dedicated local surrogate model. During optimization, the TR center shifts to the best point discovered within the region while its size adapts dynamically. The TR expands after consecutively improving the best observed value, or shrinks after consecutively failing to improve. When a TR becomes too small, it restarts at a new random location to maintain global exploration capability. This adaptive TR strategy enables efficient focus on promising regions while maintaining the ability to explore the global space.

Gaussian process (GP) is the most commonly used surrogate model for Bayesian optimization. It provides a probabilistic approach to estimate an unknown function $f : \mathbb{R}^d \to \mathbb{R}$ from a dataset $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ denotes the input vector of dimension d and $y_i \in \mathbb{R}$ denotes the scalar output.

$$y_i = f(\boldsymbol{x}_i) + \varepsilon_i \text{ with } \varepsilon_i \sim \mathcal{N}\left(0, g^2(\boldsymbol{x}_i)\right), \tag{2}$$

where the noise ε_i is typically assumed to follow a Gaussian distribution with zero mean and variance $\mathcal{N}\left(0, g^2(\boldsymbol{x}_i)\right)$. This variance can be constant or input-dependent.

Gaussian process is fully specified by a mean function $m(\boldsymbol{x})$ and a covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ [23]. For a query point $\boldsymbol{x}_*$, the posterior

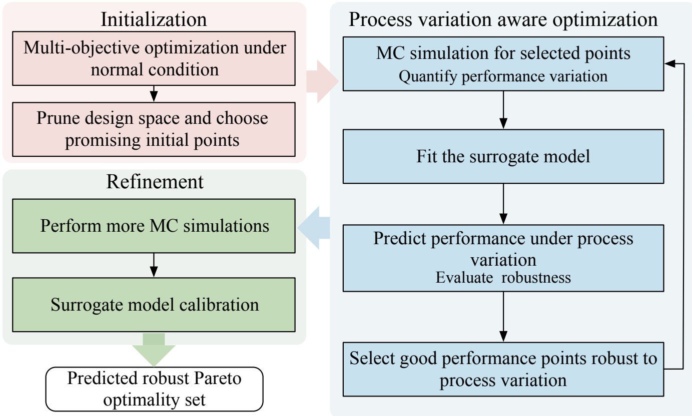


Fig. 2 Overall flow.

distribution of the new observation y_* can be derived as:

$$p(y_*|\boldsymbol{x}_*, \boldsymbol{g}, \boldsymbol{g}_*, \mathcal{D}) = \mathcal{N}(\mu_{y_*}, \sigma^2_{y_*}), \tag{3}$$

$$\mu_{y_*} = \mu_{f_*} = \boldsymbol{k}_*^\top (\boldsymbol{K} + \boldsymbol{S})^{-1} \boldsymbol{y}, \tag{4}$$

$$\sigma^2_{y_*} = \sigma^2_{f_*} + g^2_* = k_{**} - \boldsymbol{k}_*^\top (\boldsymbol{K} + \boldsymbol{S})^{-1} \boldsymbol{k}_* + g^2_*, \tag{5}$$

where $\boldsymbol{y} = [y_1, \ldots, y_n]^\top$ is the observed outputs and $\boldsymbol{g} = [g_1, \ldots, g_n]^\top$ is the noise standard deviations, $\boldsymbol{S} = \mathrm{diag}(g_1^2, \ldots, g_n^2)$ is the noise variance matrix, and $g_*^2 = g^2(\boldsymbol{x}_*)$ is the noise variance at the query point. $\boldsymbol{K}$ is an $N \times N$ covariance matrix with $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\boldsymbol{k}_*$ is the covariance vector calculated by $k(\boldsymbol{x}_*, \boldsymbol{x}_i)$ between the query point $\boldsymbol{x}_*$ and training input $\boldsymbol{x}_i$, $k_{**} = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$ is the self-covariance term at the query point.

C. Motivation

When optimizing circuit performance under process variation, a robust design point is characterized by both strong nominal performance and low sensitivity to variations. This insight implies that designs with superior nominal performance tend to maintain reasonable robustness against process variations. Consequently, high-quality solutions identified under typical process conditions can serve as a basis for design space exploration, circumventing costly MC simulations in the early stages. To enable efficient exploration of the design space while accounting for variations, we adopt a practical modeling approach using heteroscedastic Gaussian processes with limited MC simulations. This framework allows us to systematically identify promising design regions that exhibit both strong nominal performance and robustness to variations. Once promising regions are identified, more comprehensive MC simulations can be conducted to obtain reliable yield guarantees.

III. Algorithm

A. Overall flow

Our RSizing contains three stages: initialization, process variation aware optimization and refinement, as shown in Fig. 2. In the initialization stage, we first leverage evaluation history from multi-objective optimization under nominal process conditions to identify and prune inferior design regions. Through clustering analysis, we partition the remaining promising regions into initial trust regions, establishing a foundation for subsequent focused exploration. In the process variation aware optimization stage, we use heteroscedastic Gaussian process models to model limited MC simulation data. These models simultaneously capture both performance mean and variation patterns, enabling the prediction of circuit behavior under process variations with yield guarantees. Based on the HGP predictions, we employ an acquisition function based on Thompson sampling to select promising design points, balancing exploration of uncertain regions with the exploitation of promising areas. In the refinement stage, we calibrate our models using additional MC simulations in promising regions identified from previous stages. The calibration process addresses potential non-Gaussian characteristics in performance distributions through adaptive quantile estimation, ensuring more accurate yield assessment and performance guarantees. This strategic approach enables us to explore complex design spaces while minimizing expensive MC simulations efficiently, ultimately identifying robust solutions that simultaneously satisfy both performance metrics and yield requirements.

B. Initialization

The key insight of this stage is that designs with superior nominal performance $\boldsymbol{f}(\boldsymbol{x})$ often maintain relatively good performance under process variations $\boldsymbol{y}(\boldsymbol{x})$, though the exact Pareto dominance relationships may change. This motivates our space-pruning strategy that progressively narrows down the search space while maintaining diversity in promising solutions.

We start with multi-objective optimization under typical process conditions, which generates an evaluation history $\{(\boldsymbol{x}_i, \boldsymbol{f}_i)\}_{i=1}^n$, corresponding to the purple points in Fig. 3(a). Next, we extract the Pareto optimality set and their associated design points for MC simulation. The worst-case performance obtained from these MC simulations establishes a performance lower bound, represented by the yellow boundary in Fig. 3(a). Based on this performance lower bound, we partition the evaluated designs $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ into two sets: $\mathcal{X}_{good}$ and $\mathcal{X}_{bad}$. By analyzing the distribution of points in the design space, we identify and prune regions where $\mathcal{X}_{bad}$ points are concentrated while $\mathcal{X}_{good}$ points are sparse. Among the remaining design points, we remove those with similar $\boldsymbol{x}$ and $\boldsymbol{f}$ values to maintain diversity. The retained points, shown as purple points in Fig. 3(b), serve as initial points for the subsequent optimization phase. We also perform MC simulations for these initial points, as shown in Fig. 3(c). Finally, these points are clustered using k-means to establish trust regions, which represent promising and diverse areas for the subsequent optimization phase, as illustrated in Fig. 3(d).

C. Process variation aware optimization

Surrogate model. Previous works have employed Gaussian process to model the nominal circuit performance $f(\boldsymbol{x})$ [24], [25]. However, to guide process variation-aware optimization, we need to build a model that can fit the MC simulation data and predict performance distribution under process variation for new design points. Traditional Gaussian processes assume homoscedastic noise [26], which becomes limiting when modeling noisy performance function y since the impact of process variations often varies significantly across different design points. Heteroscedastic Gaussian processes [26] allow input-dependent noise variance, effectively addressing this limitation. For each trust region, We construct M heteroscedastic Gaussian processes to model M noisy performance functions $[y^{(1)}(\boldsymbol{x}), ..., y^{(M)}(\boldsymbol{x})]$, each $y(\boldsymbol{x})$ is defined by $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, g^2(\boldsymbol{x}))$, where $g^2(\boldsymbol{x})$ denotes the input dependent noise variance caused by process variations.

In the HGP framework, a second Gaussian Process is employed to model the logarithm of noise variance, $z(\boldsymbol{x}) = \log(g^2(\boldsymbol{x}))$, using a distinct covariance function $k_z(\boldsymbol{x}, \boldsymbol{x}')$. Modeling the logarithm

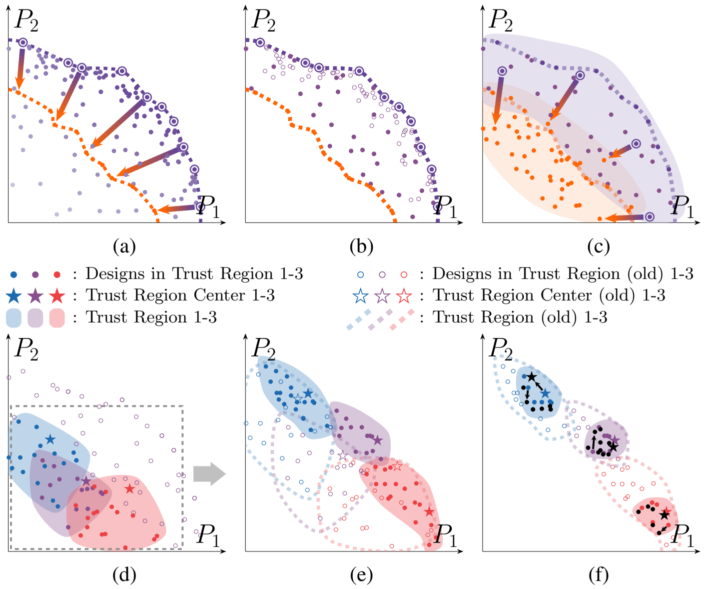


Fig. 3 Illustration of our algorithm: (a) MC simulation for Pareto optimality set (purple line) to establish performance lower bound (orange line); (b) Inferior regions pruning and similar designs removal; (c) MC simulation for Initial Points; (d) Trust regions formation by clustering; (e) Process variation aware optimization; (f) Refinement.

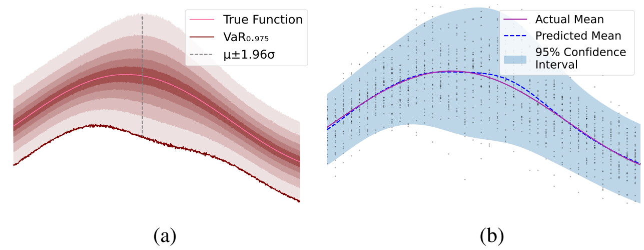


Fig. 4 (a) Function with input-dependent noise and its $\text{VaR}_{0.975}$ bound; (b) HGP model prediction with the MC simulation data.

of variance instead of modeling variance brings two benefits: it facilitates a smoother representation and ensures predictive positive variance through the exponential transformation, i.e., $g^2(\boldsymbol{x}) = \exp(z(\boldsymbol{x}))$. Consequently, the HGP framework incorporates two coupled GPs: one for inferring the unknown function (the f-process) and another for capturing the input-dependent noise (the z-process).

Since z appears as a latent variable, the predictive posterior distribution of y_* at a query point $\boldsymbol{x}_*$ can be derived by integrating out $\boldsymbol{z}$ and z_* using the law of total probability:

$$p(y_*|\boldsymbol{x}_*, \mathcal{D}) = \int\int p(y_*|\boldsymbol{x}_*, \boldsymbol{z}, z_*, \mathcal{D}) \cdot p(\boldsymbol{z}, z_*|\boldsymbol{x}_*, \mathcal{D}) d\boldsymbol{z} dz_*, \tag{6}$$

where $\boldsymbol{z} = [z_1, \dots, z_n]^\top$ are the log noise variances at n training input points, and z_* is the log noise variance at query point $\boldsymbol{x}_*$.

The integral as shown in Equation (6) becomes analytically intractable concerning the full posterior of the noise levels $p(\boldsymbol{z}, z_*|\boldsymbol{x}_*, \mathcal{D})$, thus we need to resort to approximation methods. Kersting *et al.* propose to replace the full posterior distribution of the varying noise by a point estimate at the most likely value [27]. Since noise z is also modeled by a GP, its posterior follows a normal distribution. Thus, the most likely value of noise $(\tilde{\boldsymbol{z}}, \tilde{z}_*)$ is given by the posterior mean:

$$(\tilde{\boldsymbol{z}}, \tilde{z}_*) = \underset{(\boldsymbol{z}, z_*)}{\operatorname{argmax}}\, p(\boldsymbol{z}, z_*|\boldsymbol{x}_*, \mathcal{D}) = (\mu_{\boldsymbol{z}}, \mu_{z_*}), \tag{7}$$

where $\mu_{\boldsymbol{z}}$ and μ_{z_*} are the posterior means of the z-process at training inputs and query point $\boldsymbol{x}_*$, respectively.

By replacing the full posterior $p(\boldsymbol{z}, z_*|\boldsymbol{x}_*, \mathcal{D})$ with its point estimate at most likely value $(\tilde{\boldsymbol{z}}, \tilde{z}_*)$, the integral in Equation (6) can be approximated by:

$$p(y|\boldsymbol{x}, \mathcal{D}) \approx p(y|\boldsymbol{x}, \tilde{\boldsymbol{z}}, \tilde{z}_*, \mathcal{D}) = p(y|\boldsymbol{x}, \mu_{\boldsymbol{z}}, \mu_{z_*}, \mathcal{D}). \tag{8}$$

Notice that $p(y_*|\boldsymbol{x}_*, \mu_{\boldsymbol{z}}, \mu_{z_*}, \mathcal{D})$ shares the same form as Equation (3), thus the predictive posterior distribution of y_* can be written as $\mathcal{N}(\mu_{y_*}, \sigma^2_{y_*})$. Here, μ_{y_*} can be directly obtained from Equation (4), and since $g^2(\boldsymbol{x})$ can be represented by $\exp(z(\boldsymbol{x}))$, $\sigma^2_{y_*}$ can be obtained by substituting $\exp(\mu_{z_*})$ for g^2_* in Equation (5).

The performance distributions affected by process variations may not strictly follow Gaussian distributions and vary across different performance metrics and design points. Accurate characterization of these distributions requires extensive MC simulations which is computationally expensive. Instead, we use limited MC samples at each design point and approximate them with Gaussian models characterized by mean and variance. This approach can effectively capture the general trend of performance distributions. In the process variation aware optimization stage, for each observation point $\boldsymbol{x}*$, we utilize k MC samples $y_1, \dots, y_k$ of each individual performance metric to fit an HGP model. We model the mean using f-process with $\hat{\mu}(\boldsymbol{x}) = \frac{1}{k}\sum i = 1^k y_i$ and fit the log-variance using z-process with $\hat{z}(\boldsymbol{x}) = \log(\frac{1}{k-1}\sum_{i=1}^k (y_i - \hat{\mu}(\boldsymbol{x}))^2)$ respectively. Each HGP is trained only with samples within its corresponding trust region, which reduces the running time for model fitting.

Robustness prediction by model. To maintain the required yield while optimizing performances, we adopt value at risk (VaR), which provides a probabilistic guarantee on performance values. For a noisy performance function $y(\boldsymbol{x})$, VaR at point $\boldsymbol{x}$ with yield requirement $\alpha \in [0, 1]$ is defined as:

$$\text{VaR}_\alpha\big[y(\boldsymbol{x})\big] = \max\{z \in \mathbb{R} : p\big(y(\boldsymbol{x}) \geq z\big) \geq \alpha\}. \tag{9}$$

Intuitively, VaR defines the best achievable performance at design point $\boldsymbol{x}$ while maintaining the yield requirement α.

Multivariate Value-at-Risk (MVaR) [28] extends the VaR concept to the multi-objective scenario. Due to multiple objectives, the single maximum is replaced by a Pareto optimality set. The MVaR of $\boldsymbol{y}(\boldsymbol{x})$ for a given point $\boldsymbol{x}$ and yield requirement $\alpha \in [0, 1]$ is:

$$\text{MVaR}_\alpha\big[\boldsymbol{y}(\boldsymbol{x})\big] = \mathcal{P}\big(\{\boldsymbol{z} \in \mathbb{R}^M : p\big(\boldsymbol{y}(\boldsymbol{x}) \geq \boldsymbol{z}\big) \geq \alpha\}\big). \tag{10}$$

Similar to VaR, MVaR can be interpreted as a performance vector $\boldsymbol{z}$ that satisfies $p(\boldsymbol{y}(\boldsymbol{x}) \geq \boldsymbol{z}) \geq \alpha$. Since the yield requirement α is typically high, MVaR essentially represents the worst-case performance vector under process variations. In this high-yield scenario ($\alpha \approx 1$), the joint probability becomes dominated by individual performance constraints, allowing MVaR to be effectively approximated using the VaR of individual performance metrics:

$$\text{MVaR}_\alpha\big[\boldsymbol{y}(\boldsymbol{x})\big] \approx [\text{VaR}_\alpha[y^{(1)}(\boldsymbol{x})], \dots, \text{VaR}_\alpha[y^{(M)}(\boldsymbol{x})]]. \tag{11}$$

HGP provides an efficient way to calculate VaR. As defined in Equation (9), VaR_α can be expressed as the $(1-\alpha)$-quantile of $y(\boldsymbol{x})$. Given that the predictive posterior follows a Gaussian distribution $\mathcal{N}(\mu_y, \sigma^2_y)$, VaR_α can be conveniently computed as $\mu_y - \Phi^{-1}(\alpha)\sigma_y$, where Φ^{-1} is the inverse cumulative distribution function of the standard normal distribution. For instance, $\mathrm{VaR}_{0.975} = \mu_y - 1.96\sigma_y$. By substituting this VaR expression into Equation (11), we can efficiently estimate MVaR at any new design point using HGP predictions. This analytical form of MVaR prediction is particularly valuable for our optimization. Here, MVaR serves as a robust optimization objective that inherently guarantees the specified yield requirement - by continuously optimizing MVaR, we can systematically search for design points that not only achieve superior performance but also maintain robustness against process variations while satisfying the yield constraints.

Fig. 4 illustrates the VaR approximation and HGP modeling process. In Fig. 4(a), we demonstrate a function with input-dependent noise, where the red shaded region represents different confidence levels, and the dashed line indicates the $\mu \pm 1.96\sigma$ interval (95% confidence interval). The curve at the bottom shows $\mathrm{VaR}_{0.975}$, which ensures that the performance exceeds this bound with a probability of 97.5%, thus satisfying a 97.5% yield requirement. Fig. 4(b) shows how our heteroscedastic Gaussian process (HGP) model effectively captures both the mean performance and its variations. The black dots represent the MC simulation data, while the purple and blue dashed lines show the actual and predicted means, respectively. The blue-shaded region indicates the 95% confidence interval of the prediction. The close alignment between the actual and predicted means, along with the appropriate coverage of simulation data points by the confidence interval, validates our HGP model's capability to capture both the performance trends and their input-dependent variations.

Acquisition Function. Our objective is to optimize the robust Pareto optimality set formed by MVaR values, where each solution inherently satisfies the yield requirement. By maximizing the hypervolume (HV) of this set, we can improve the quality of solutions while maintaining their robustness against process variations. To efficiently select new design points that are most likely to increase the HV, we propose an acquisition function based on Thompson sampling (TS) [29]. TS balances exploration and exploitation by randomly sampling from the posterior distribution of GP models and generating new query points through optimization on these samples. This approach naturally supports batch selection. In practice, we determine the batch size q based on the MC simulation number that can be executed in parallel.

The selection process begins by computing MVaR values (set $\mathcal{M}$) for all evaluated design points $[\boldsymbol{x}_1, ..., \boldsymbol{x}_n]$ using HGP, followed by constructing the current Pareto optimality set $\mathcal{P}$ based on $\mathcal{M}$. We then generate candidate points $\mathcal{X}_{\text{cand}}$ from all trust regions, from which q points will be selected as the next batch. Once a point is selected, it is removed from $\mathcal{X}_{\text{cand}}$. We use a sequential greedy strategy where q points are selected one by one. When selecting the i-th point, we draw samples from the joint posterior distribution over previously selected points $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_{i-1}\}$ and remaining candidate points in $\mathcal{X}_{\text{cand}}$. For each candidate point $\hat{\boldsymbol{x}}$, we combine its predicted MVaR value with those of previously selected points to form a new set $\mathcal{M}_{\text{new}} = \{\tilde{M}(\boldsymbol{x}_1), ..., \tilde{M}(\boldsymbol{x}_{i-1}), \tilde{M}(\hat{\boldsymbol{x}})\}$ and evaluate the HV improvement:

$$\mathrm{HVI}(\mathcal{P}, \mathcal{M}_{\text{new}}, \mathbf{r}) = \mathrm{HV}(\mathcal{P} \cup \mathcal{M}_{\text{new}}, \mathbf{r}) - \mathrm{HV}(\mathcal{P}, \mathbf{r}). \tag{12}$$

The candidate point yielding the maximum HV improvement (HVI) is then selected and added to the batch. By calculating the joint HVI with previously selected points, we can achieve comprehensive exploration. If a previously selected point has already improved specific objectives in a region, nearby candidates with similar improvements would yield limited HVI, thus encouraging the acquisition function to explore different regions and objective trade-offs. This sequential strategy ensures diverse batch selection while enabling trust regions to collaboratively maximize the global hypervolume. For a trust region, if at least one proposed candidate improves the global hypervolume, the iteration is considered successful; otherwise, it is deemed a failure. The trust regions then execute their corresponding update strategies. As illustrated in Fig. 3(e), during the process variation-aware optimization phase, through multiple iterations, our algorithm progressively discovers superior solutions while trust region centers converge towards the optimal regions in the design space.

D. Refinement

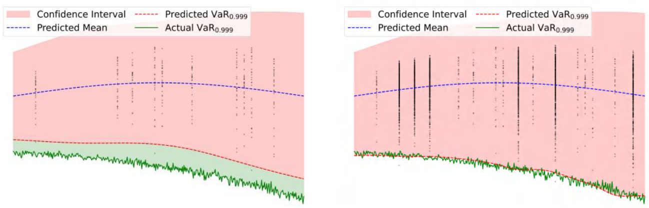
In the previous phase, we predicted and optimized MVaR using HGP, expecting designs in Pareto optimality set to meet the yield requirement α. However, this requirement is often not satisfied due to three main factors. First, the limited number of MC simulations leads to imprecise mean and variance estimates; second, the performance variations at individual points may not strictly follow Gaussian distribution, making the VaR calculation based on Gaussian assumptions less accurate; finally, the MVaR approximation using individual VaRs in Equation (11) introduces bias when $\alpha < 1$.

To obtain more reliable solutions, we initiate a refinement phase when the hypervolume improvement converges. Since TR centers gradually move towards optimal solutions, their neighborhoods represent potential high-quality designs. During this phase, we conduct additional Monte Carlo simulations at selected design points near TR centers to obtain high-fidelity estimates of critical quantiles governing yield requirements. Originally, the HGP model estimated quantiles via the parametric form $\mu - \Phi^{-1}(\alpha)\sigma$ which relies on Gaussian assumptions. To address potential non-Gaussian characteristics, we replace the static coefficient $\Phi^{-1}(\alpha)$ with a correction factor k that adaptively adjusts the quantile estimation:

$$k = \frac{\mu - q_\alpha}{\sigma}, \tag{13}$$

where q_α denotes the empirical quantile obtained from Monte Carlo simulations. We use another Gaussian process to model k. By incorporating additional MC data from the refinement phase, the framework iteratively improves both the GP model for k and HGP model. This joint update enables the framework to better capture non-Gaussian characteristics and correct potential estimation biases in the design space. Based on our observations, design points within local trust region tend to have similar statistical characteristics in their performance distributions (e.g., skewness and shape). This local similarity enables us to leverage spatial correlation in the GP model - when we update the mean, variance, and correction factor k through MC simulations at selected design points, these updates naturally propagate to neighboring points.

Fig. 5 demonstrates this calibration process where the performance distributions under different design parameters exhibit left-skewed characteristics. In Fig. 5(a), the green solid line represents the actual VaR, while the red dashed line shows the predicted VaR. Black dots represent MC simulation samples. Initially, due to limited MC samples and Gaussian assumptions, the predicted VaR is optimistic compared to the actual VaR. In Fig. 5(b), through intensive MC simulations at selected design points (shown as dense black dots), the calibration effort significantly improves the prediction accuracy across the input space, as evidenced by the close alignment between predicted and actual VaR.



(a) (b)

Fig. 5 (a) Initial VaR prediction by HGP model; (b) Refined VaR prediction after calibration.

TABLE I Parameter number and Objectives.

Circuit	# of Param.	Objectives
OPAMP	36	Gain Static current (I_{DDQ}) Offset voltage (V_{OS}) CMRR
OSC	14	Dynamic current (I_{DD}) Static current (I_{DDQ})
PLL	77	Average current ($I_{\mathrm{DD,avg}}$) Clock signal peak-to-peak voltage ($V_{\mathrm{pp,clk}}$)

This process iterates until the predicted VaR aligns well with the MC simulation results, leading to more accurate MVaR estimates and yield predictions. As shown in Fig. 3(f), the original design points are calibrated to their refined performance values (shown as black points). We then output the Pareto optimality set formed by these calibrated MVaR values as our final solutions, which achieve both superior performance and guaranteed yield requirements. It is worth noting that our refinement does not address the approximation error introduced by MVaR estimation as shown in Equation (11), which may introduce bias when yield requirements are lower. However, this approximation error is acceptable in practice since the yield requirement is typically very high.

IV. Experiment

A. Experimental Settings

We evaluate our proposed method on benchmark circuits from ITC 2017 [30], which consists of three representative analog circuits: an operational amplifier (OPAMP), an oscillator (OSC), and a phase-locked loop (PLL). The circuit schematics are shown in Fig. 6. We construct the design space by selecting a subset of key design parameters, and the entire exploration flow is implemented using Empyrean PyAether [31]. TABLE I shows the dimensions of design parameters and objectives for each circuit. Circuit simulations are conducted using the Spectre simulator in the TSMC 40nm process, where single simulation time is about 5s, 11s, and 1 minute for OPAMP, OSC, and PLL respectively. All experiments are performed on a Linux machine equipped with an 80 cores CPU (2.70GHz) and an A100 GPU with 80GB memory.

For each circuit, three different yield requirements (97.5%, 99.0%, and 99.9%) are considered to obtain the Pareto optimality set. During initialization, we perform multi-objective optimization under typical process conditions using MORBO [22] with a batch size of 5 over 60 iterations. For the process variation aware optimization stage, the number of MC simulations k is set to 50, which is sufficient to capture the general trend of performance distributions. Based on circuit complexity and computational resource constraints, the batch sizes for OPAMP, OSC, and PLL circuits are set to 6, 4, and 2 respectively. In the refinement stage, the number of MC simulations at each design point scales with yield requirements - 100 for 97.5%, 200 for 99.0%, and 500 for 99.9%, ensuring sufficient accuracy at different yield targets.

Since both prior work and commercial EDA tools do not provide systematic solutions for analog circuit sizing with a fixed yield constraint, we select the general multi-objective genetic algorithm NSGA-II [8], the state-of-the-art robust Bayesian optimization method MARS-NEI [19], and efficient PVT and mismatch robust analog circuit sizing framework PVTSizing [15] as baselines for comparison. For NSGA-II, the population count is set to 40 and the number of generations is 50. For yield targets of 97.5%, 99.0%, and 99.9%, we use the worst results from 40, 100, and 500 MC simulations respectively as optimization objectives. Due to time-consuming PLL MC simulations, NSGA-II was omitted for its excessive simulation requirements.

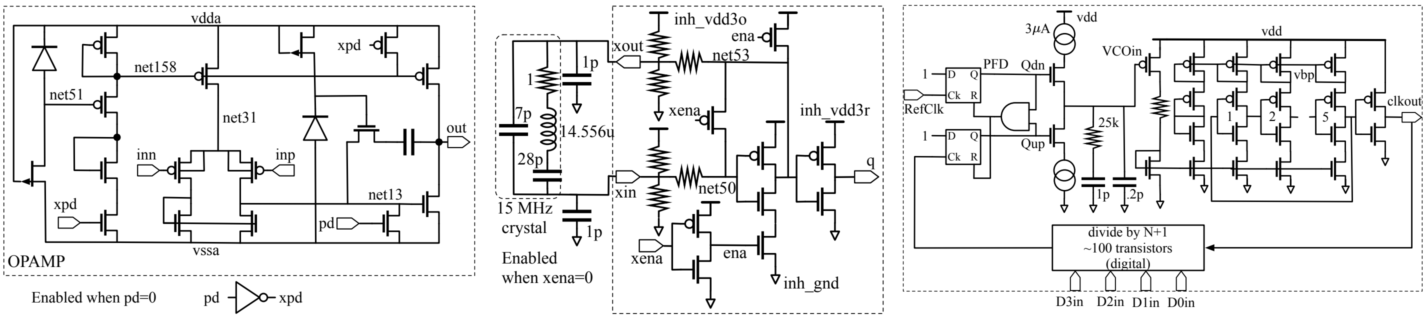
MARS-NEI was designed for robust optimization problems with known input noise, proposing a scalarization method that transforms multi-objective risks into a scalar value and employs noisy expected improvement (NEI) [32] as the acquisition function. To overcome the requirement of prior knowledge on noise distribution, we adapt this method to handle unknown circuit process variations. In the process variation aware optimization stage, it uses the same heteroscedastic Gaussian process as ours to fit 50 MC simulation data and similarly refines solutions near the Pareto optimality set with additional MC simulations after the hypervolume converges. The hypervolume reference point $\boldsymbol{r}$ was set to the worst values found during optimization. For fair comparison, performance metrics were normalized when calculating the hypervolume.

PVTSizing aims to find circuit design solutions that meet performance constraints across all corners. It uses TuRBO [21] to generate high-quality initial points and employs Actor-Critic reinforcement learning framework for multi-task optimization. The approach first optimizes the design across PVT corners before addressing mismatch effects. In the mismatch phase, it evaluates the current design with multiple MC simulations, identifies the worst-performing mismatch corners, and uses the Actor-Critic network to generate and select optimized solutions. This process repeats until a solution passes verification across all mismatch and PVT corners. We focused on 5 process corners during the PVT phase, as our method specifically addresses the impact of process variations. For each circuit, we run PVTSizing three times, with three different constraint targets selected from representative points of the Pareto optimality set under Rsizing's 99.9% yield requirement.

B. Results and Discussion

The actual yield of each Pareto-optimal solution was verified through 1000 MC simulations. We report the mean yield with its standard deviation, along with the minimum and maximum yields observed in TABLE II. The results show our method achieves consistently satisfactory performance across different circuits and yield requirements. While the worst-case solutions may not meet the specified yield requirements, setting a higher target yield during optimization can address this issue, though at the cost of degraded circuit performance, representing an trade-off in the design process.

For each circuit, we selected two performance metrics and visualized their Pareto optimality set under different yield requirements and typical process in Fig. 7. Note that all performance metrics are transformed to maximization problems by taking their negative values. The results clearly demonstrate the trend across different yield requirements: Pareto optimality set corresponding to higher yield requirements exhibit inferior performance and are located in



(a) (b) (c)

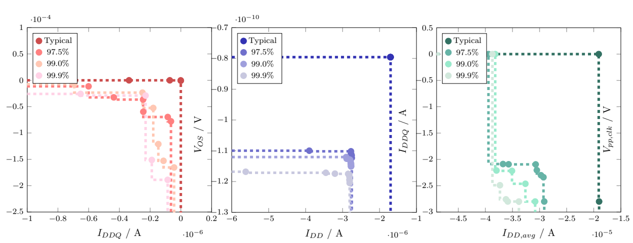
Fig. 6 Benchmark circuits: (a) OPAMP; (b) OSC; (c) PLL.

TABLE II Yield Results (%).

Yield Req.	OPAMP			OSC			PLL		
	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
97.5	96.75±0.34	95.5	97.4	97.18±0.22	96.3	97.7	96.43±0.35	95.1	97.2
99.0	98.72±0.25	97.9	99.3	98.85±0.18	98.2	99.3	98.68±0.29	97.6	99.4
99.9	99.79±0.08	99.5	100.0	99.86±0.05	99.7	100.0	99.74±0.10	99.4	100.0

TABLE III Performance Comparison.

Circuits	Yield Req.	NSGA-II [8]			MARS-NEI [19]			Our RSizing		
		Y (%)	HV	RT (h)	Y (%)	HV	RT (h)	Y (%)	HV	RT (h)
OPAMP	97.5	95.23	3865	2.64	96.83	4512	3.25	96.75	**4968**	**2.18**
	99.0	97.92	2969	3.73	98.77	3486	3.34	98.72	**3692**	**2.27**
	99.9	99.16	2112	8.25	99.82	2465	3.52	99.79	**2673**	**2.41**
OSC	97.5	95.84	2620	6.45	97.21	2785	4.12	97.18	**2897**	**2.86**
	99.0	98.25	2574	8.48	98.83	2726	4.35	98.85	**2812**	**3.02**
	99.9	99.62	2431	20.63	99.88	2591	4.69	99.86	**2660**	**3.23**
PLL	97.5	-	-	>30	96.49	6183	9.87	96.43	**6742**	**7.62**
	99.0	-	-	>30	98.71	5965	11.08	98.68	**6561**	**8.85**
	99.9	-	-	>30	99.75	5668	13.92	99.74	**6143**	**11.34**



(a) (b) (c)

Fig. 7 Pareto optimality set of different yield requirements and typical process: (a) OPAMP; (b) OSC; (c) PLL.

the lower-left region of the plots. This enables flexible design point selection based on specific yield and performance requirements.

TABLE III presents the comparison of mean yield (Y) across the Pareto optimality set, hypervolume (HV) measuring solution quality, and runtime (RT) for different methods. Our RSizing demonstrates competitive performance across these metrics compared to other approaches. The results show that both MARS-NEI and our RSizing achieve yield values closer to the specified requirements compared to NSGA-II, while also achieving better performance as indicated by higher hypervolume values. For the comparison between MARS-NEI and RSizing, RSizing's yield is occasionally slightly lower than MARS-NEI, which is primarily because RSizing explores higher-performance regions where process variations are likely to have more complex and pronounced effects. However, our method's space pruning and efficient robustness evaluation strategies enable superior performance with reduced computational cost across all benchmark circuits, especially for high-dimensional problems like PLL.

TABLE IV Yield/Runtime Comparison with PVTSizing.

Circuit	Target	PVTSizing [15]		Our RSizing	
		Y (%)	RT (h)	Y (%)	RT (h)
OPAMP	T1	95.1	1.53	**99.5** (↑4.4)	2.41
	T2	98.4	1.18	**99.8** (↑1.4)	
	T3	96.2	1.25	**99.8** (↑3.6)	
OSC	T1	99.6	0.73	**100.0** (↑0.4)	3.23
	T2	98.8	0.95	**99.9** (↑1.1)	
	T3	99.0	0.86	**99.8** (↑0.8)	
PLL	T1	96.9	6.34	**99.8** (↑2.9)	11.34
	T2	98.2	5.17	**99.8** (↑1.6)	
	T3	97.3	5.46	**99.7** (↑2.4)	

TABLE IV compares yield performance and runtime between our method and PVTSizing across all circuits at three target points. While PVTSizing shows individual runtime values for each target point, RSizing reports a single runtime value as it efficiently generates multiple yield-compliant Pareto-optimal solutions in one execution. PVTSizing achieves lower yield values, demonstrating its limitations in effectively capturing process variation trends. Although PVTSizing shows runtime advantages for single-point optimization, RSizing's strategy enables comprehensive exploration of performance trade-offs under specified yield constraints. This strategy addresses a common design concern where performance is compromised by selecting conservative specifications to guarantee yield requirements, enabling exploration of the circuit's maximum achievable performance.

V. Conclusion

This paper presents RSizing, a novel approach for robust analog circuit sizing that effectively handles process variations while optimizing circuit performance. Our method combines space pruning with targeted exploration and employs HGP modeling to accurately capture the non-uniform nature of process-induced performance variations. We propose a three-phase optimization strategy that begins with design space pruning through nominal condition optimization, followed by process variation-aware optimization using heteroscedastic Gaussian processes and efficient acquisition functions, and concludes with solution refinement using targeted MC simulations. Experimental results demonstrate that RSizing achieves superior performance across benchmark circuits.

Acknowledgement

This work is supported in part by the National Key Research and Development Program of China (No. 2023YFB4402900) and the National Natural Science Foundation of China (No. 62304197).

References

[1] P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, *Analysis and design of analog integrated circuits*. John Wiley & Sons, 2024.

[2] S. Zhang, F. Yang, C. Yan, D. Zhou, and X. Zeng, "An efficient batch-constrained Bayesian optimization approach for analog circuit synthesis via multiobjective acquisition ensemble," *IEEE TCAD*, vol. 41, no. 1, pp. 1–14, 2021.

[3] S. Yin, R. Wang, J. Zhang, X. Liu, and Y. Wang, "Fast surrogate-assisted constrained multiobjective optimization for analog circuit sizing via self-adaptive incremental learning," *IEEE TCAD*, vol. 42, no. 7, pp. 2080–2093, 2022.

[4] H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich, "The sizing rules method for analog integrated circuit design," in *Proc. ICCAD*. IEEE, 2001, pp. 343–349.

[5] K. Settaluri, Z. Liu, R. Khurana, A. Mirhaj, R. Jain, and B. Nikolic, "Automated design of analog circuits using reinforcement learning," *IEEE TCAD*, vol. 41, no. 9, pp. 2794–2807, 2021.

[6] A. F. Budak, P. Bhansali, B. Liu, N. Sun, D. Z. Pan, and C. V. Kashyap, "DNN-opt: An RL inspired optimization for analog circuit sizing using deep neural networks," in *Proc. DAC*. IEEE, 2021, pp. 1219–1224.

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

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

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

[10] D. D. Weller, M. Hefenbrock, M. Beigl, and M. B. Tahoori, "Fast and efficient high-sigma yield analysis and optimization using kernel density estimation on a Bayesian optimized failure rate model," *IEEE TCAD*, vol. 41, no. 3, pp. 695–708, 2021.

[11] X. Wang, C. Yan, F. Yang, D. Zhou, and X. Zeng, "An efficient yield optimization method for analog circuits via Gaussian process classification and varying-sigma sampling," in *Proc. DAC*, 2022, pp. 625–630.

[12] R. Schwencker, F. Schenkel, M. Pronath, and H. Graeb, "Analog circuit sizing using adaptive worst-case parameter sets," in *Proc. DATE*. IEEE, 2002, pp. 581–585.

[13] Y. Liu, G. Dai, Y. Cheng, W. Kang, and W. W. Xing, "OPT: Optimal proposal transfer for efficient yield optimization for analog and SRAM circuits," in *Proc. ICCAD*. IEEE, 2023, pp. 1–9.

[14] W. Shi, H. Wang, J. Gu, M. Liu, D. Z. Pan, S. Han, and N. Sun, "Robustanalog: Fast variation-aware analog circuit design via multi-task RL," in *Proc. MLCAD*, 2022, pp. 35–41.

[15] Z. Kong, X. Tang, W. Shi, Y. Du, Y. Lin, and Y. Wang, "PVTSizing: A TuRBO-RL-based batch-sampling optimization framework for PVT-robust analog circuit synthesis," in *Proc. DAC*, 2024, pp. 1–6.

[16] J. Gao, W. Cao, and X. Zhang, "RoSE: Robust analog circuit parameter optimization with sampling-efficient reinforcement learning," in *Proc. DAC*. IEEE, 2023, pp. 1–6.

[17] W. Cao, J. Gao, T. Ma, R. Ma, M. Benosman, and X. Zhang, "RoSE-Opt: Robust and efficient analog circuit parameter optimization with knowledge-infused reinforcement learning," *IEEE TCAD*, 2024.

[18] J. Li, H. Zhi, W. Shan, Y. Li, Y. Zeng, and Y. Li, "Multi-task evolutionary to PVT knowledge transfer for analog integrated circuit optimization," in *Proc. ICCAD*. IEEE, 2023, pp. 1–9.

[19] S. Daulton, S. Cakmak, M. Balandat, M. A. Osborne, E. Zhou, and E. Bakshy, "Robust multi-objective Bayesian optimization under input noise," in *Proc. ICML*. PMLR, 2022, pp. 4831–4866.

[20] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[21] D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek, "Scalable global optimization via local Bayesian optimization," *Proc. NIPS*, vol. 32, 2019.

[22] S. Daulton, D. Eriksson, M. Balandat, and E. Bakshy, "Multi-objective bayesian optimization over high-dimensional search spaces," in *Proc. UAI*. PMLR, 2022, pp. 507–517.

[23] C. E. Rasmussen, C. K. Williams *et al.*, *Gaussian processes for machine learning*. Springer, 2006.

[24] M. Barros, J. Guilherme, and N. Horta, "Analog circuits optimization based on evolutionary computation techniques," *Integration*, vol. 43, no. 1, pp. 136–155, 2010.

[25] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Multi-objective Bayesian optimization for analog/RF circuit synthesis," in *Proc. DAC*, 2018, pp. 1–6.

[26] P. Goldberg, C. Williams, and C. Bishop, "Regression with input-dependent noise: A Gaussian process treatment," *Proc. NIPS*, vol. 10, 1997.

[27] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, "Most likely heteroscedastic Gaussian process regression," in *Proc. ICML*, 2007, pp. 393–400.

[28] A. Prékopa, "Multivariate value at risk and related topics," *Annals of Operations Research*, vol. 193, no. 1, pp. 49–69, 2012.

[29] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," *Proc. NIPS*, vol. 24, 2011.

[30] S. Sunter and P. Sarson, "A/MS benchmark circuits for comparing fault simulation, DFT, and test generation methods," in *Proc. ITC*, 2017, pp. 1–7.

[31] Empyrean, "Empyrean PyAether: Unified data ecosystem for full-custom circuit design," Technical Documentation, 2024. [Online]. Available: https://www.empyrean-tech.com

[32] B. Letham, B. Karrer, G. Ottoni, and E. Bakshy, "Constrained Bayesian optimization with noisy experiments," *Bayesian Analysis*, vol. 14, no. 2, pp. 495–519, 2019.