SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

April 2024, Vol. 67, Iss. 4, 142401:1–142401:12
https://doi.org/10.1007/s11432-022-3878-5

Post-layout simulation driven analog circuit sizing

Xiaohan GAO[1,2], Haoyi ZHANG[1], Siyuan YE[1], Mingjie LIU[4], David Z. PAN[4], Linxiao SHEN[1], Runsheng WANG[1,3], Yibo LIN[1,3*] & Ru HUANG[1,3]

[1]*School of Integrated Circuits, Peking University, Beijing* 100871*, China;*
[2]*School of Computer Science, Peking University, Beijing* 100871*, China;*
[3]*Institute of Electronic Design Automation, Peking University, Wuxi* 214000*, China;*
[4]*Department of Electrical and Computer Engineering, University of Texas at Austin, Austin* 78712*, USA*

Received 29 October 2022/Revised 8 February 2023/Accepted 7 October 2023/Published online 20 March 2024

Abstract Post-layout simulation provides accurate guidance for analog circuit design, but post-layout performance is hard to be directly optimized at early design stages. Prior work on analog circuit sizing often utilizes pre-layout simulation results as the optimization objective. In this work, we propose a post-layout-simulation-driven (post-simulation-driven for short) analog circuit sizing framework that directly optimizes the post-layout simulation performance. The framework integrates automated layout generation into the optimization loop of transistor sizing and leverages a coupled Bayesian optimization algorithm to search for the best post-simulation performance. Experimental results demonstrate that our framework can achieve over 20% better post-layout performance in competitive time than manual design and the method that only considers pre-layout optimization.

Keywords analog EDA, transistor sizing, Bayesian optimization, post-layout simulation

1 Introduction

Analog circuit sizing is a critical task in analog design automation. The sizing configuration of transistors not only dominates the performance in the schematic design but also has a high impact on the subsequent layout design. Previous analog sizing methods based on pre-layout schematic simulation become incompatible with the increasingly sophisticated performance requirements, which drives the emerging research on analog circuit sizing aware of post-layout effects. With the increasingly sophisticated performance requirements, analog sizing methods only considering the pre-layout simulation are no longer enough. Thus, analog circuit sizing is evolving toward the awareness of post-layout performance.

Conventional analog design flow can be split into schematic phase and layout phase. In the pre-layout phase, an analog designer develops a sizing configuration to meet a performance specification according to the results of the pre-layout simulation. In the conventional design flow, layout design requires end-to-end manual participation. Therefore, it is difficult to consider post-layout performance at the circuit sizing stage, as re-sizing the schematic design could lead to re-drawing the layout manually, which could be extremely time-consuming especially when such a procedure needs to iterate for design closure. However, the emergence of fully automated layout generation tools, such as ALIGN [1], BAG [2, 3], and MAGICAL [4], bring new opportunities for the revolution of the existing fully manual design methodology. As these tools are designed under the philosophy of "no-human-in-the-loop", they can be integrated into the optimization loops for efficient layout generation. The insuperable barrier between the pre-layout design phase and the layout design phase can be broken.

With the technology node shrinking to a smaller scale and the performance requirements becoming more specialized, the discrepancy between pre-layout schematic simulation and post-layout simulation cannot be ignored. Recent studies on analog circuit sizing unveil the insufficiency of simply relying on the pre-layout simulation alone for analog circuit sizing. Liu et al. [5] proposed a post-layout parasitic-aware circuit sizing method using graph neural network as a surrogate model. Ranjan et al. [6] took the

* Corresponding author (email: yibolin@pku.edu.cn)

post-layout parasitics effects into account with a symbolic performance model. BagNet [7] introduces a sizing framework that uses a deep neural network to select a sizing configuration, generate a layout, and extract parasitics for the selected sizing configuration. AutoCkt [8] utilizes deep reinforcement learning for circuit synthesis taking post-layout parasitics into account. These studies integrate the aforementioned layout generation tools as the internal layout generators. However, none of those studies consider the post-simulation performance as the direct optimization objective. Moreover, most studies leverage neural networks as surrogate models, which usually require large amounts of data to achieve high accuracy for finding high-quality solutions [9, 10]. To accelerate the analog design closure, we should go one step further to optimize the post-simulation performance directly.

Despite the diverse modeling of analog circuit sizing problem, most of the sizing studies treat the sizing solution synthesis as a black-box optimization problem. Early attempts include using simulated annealing on symbolic AC models, geometric programming, and evolutionary algorithms [11]. Recent work [12] also explores reinforcement learning combined with graph neural networks for transferable transistor sizing. Li et al. [13] proposed an actor-critic reinforcement learning approach to optimize post-simulation performance. However, they do not combine pre-layout simulation with post-layout simulation to consider the sizing process. Besides, Bayesian optimization is also adopted in analog circuit sizing for optimizing the pre-layout performance [14, 15], as it is suitable for problems with objectives expensive to compute. Recent work considers post-layout performance with Bayesian fusion technique [16, 17], which calculates the initial performance model and requires an additional training process.

To enable direct optimization for the post-layout performance, we propose a post-simulation-driven analog circuit sizing framework. We create a Bayesian optimization paradigm of two coupled Bayesian optimization models. The specially designed framework can perform well in terms of efficiency as well as post-layout performance.

We summarize our major contributions as follows:

• This work directly optimizes post-layout performance at the analog circuit sizing stage.

• We propose a new paradigm of coupled Bayesian optimization and apply this optimization technique to the sizing problem, to leverage both the efficiency of the pre-layout simulation and the precision of the post-layout simulation.

• Experimental results on real-world analog circuit designs demonstrate the advantages of our framework to enhance the post-layout performance in terms of multiple metrics.

The rest of this paper is organized as follows. Section 2 recalls the pre-layout simulation and post-layout simulation (post-simulation for short), and introduces the basic idea of Bayesian optimization. In the last part of Section 2, we formulate the post-simulation-driven analog circuit sizing problem. Section 3 takes a journey on the algorithm details of our proposed framework. Section 4 discusses how we set up our experiment and demonstrates the experimental results. Section 5 summarizes the whole paper.

2 Preliminaries

In this section, we introduce the background of our framework, including pre-layout simulation, post-layout simulation, and Bayesian optimization. Besides, we formulate the post-simulation-driven circuit sizing problem.

2.1 Pre-layout simulation vs. post-layout simulation

Previous studies on analog circuit sizing often utilize pre-layout simulation as a performance modeling method [14]. Figure 1 illustrates an example of pre-layout simulation results and post-layout simulation results on loop gain. We plot the results on the gain results varying with two selected sizing parameters. The gain results of pre-layout simulation form the surface of blue dotted points and the gain results of post-layout simulation form the surface of red cross points. As we can see, the pre-layout simulation results are generally more optimistic than the post-layout simulation. The surface of pre-layout simulation results varies much more smoothly with the two parameters. From a global perspective, the pre-layout simulation results turn out to have a similar trend to the post-layout simulation. However, when diving into a local region, the pre-layout simulation cannot model the accurate post-layout performance. Note that this figure is a direct view of how one metric varies with two parameters, while a realistic sizing scenario involves multiple metrics and a series of sizing parameters. For the high-dimensional optimization

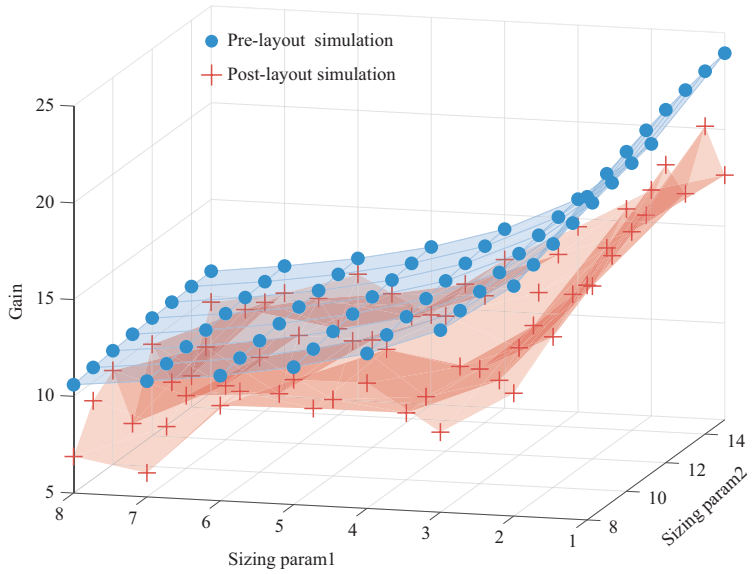

Figure 1 (Color online) Pre-layout and post-layout simulation results on Gain varying with two sizing parameters.

problem, the surfaces of pre-layout simulation and post-layout simulation in the high-dimensional space will be way more different.

The results of the pre-layout simulation can neither reflect the interconnect parasitics nor all the other complex layout-dependent effects on the circuit performance, which introduces a bias that cannot be ignored. As directly optimizing the post-layout performance is desired, switching to post-layout simulation could be a potential solution. However, post-layout simulation suffers from layout generation overhead and heavy computation costs. It is intuitive to discover a hybrid high-performance sizing framework combining pre-layout simulation and post-layout simulation.

2.2 Bayesian optimization

Bayesian optimization is a widely used strategy for global black-box optimization problems, especially for those expensive-to-evaluate black-box functions [18–21]. The essence of Bayesian optimization is rooted in the surrogate model and acquisition function. The surrogate model is supposed to approximate the objective function and quantify the uncertainty on the posterior distribution. The acquisition function proposes sampling points and evaluates the usefulness of the sampled point for maximizing the objective function.

The most commonly used surrogate model is the Gaussian process (GP) [22]. A $\mathrm{GP}(\mu_0, k)$ is a non-parametric model described by prior mean function $\mu_0 : \mathcal{X} \mapsto \mathbb{R}$ and covariance function $k : \mathcal{X} \times \chi \mapsto \mathbb{R}$. The unknown function values $\boldsymbol{f}$ are modeled as a joint Gaussian distribution. Given $\boldsymbol{f}$, the noisy observations $\boldsymbol{y}$ will be normally distributed:

$$\begin{aligned} \boldsymbol{f} \mid \boldsymbol{X} &\sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{K}), \\ \boldsymbol{y} \mid \boldsymbol{f}, \sigma^2 &\sim \mathcal{N}(\boldsymbol{f}, \sigma^2 \boldsymbol{I}), \end{aligned} \tag{1}$$

where the ith element of the mean vector $\boldsymbol{m}$ is $m_i := \mu_0(\boldsymbol{x}_i)$ and the (i, j) element of the covariance matrix $\boldsymbol{K}$ is $K_{i,j} := k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The distribution on $\boldsymbol{f}$ gives the prior distribution $p(\boldsymbol{f})$ for the GP. For a newly observed data point $\boldsymbol{x}$, the conditional probability of random variable $f(\boldsymbol{x})$ on previous observations $\mathcal{D}$ is normally distributed:

$$\begin{aligned} \mu(\boldsymbol{x}) &= \mu_0(\boldsymbol{x}) + \boldsymbol{k}(\boldsymbol{x})^{\mathrm{T}} (\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} (\boldsymbol{y} - \boldsymbol{m}), \\ \sigma^2(\boldsymbol{x}) &= \kappa(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^{\mathrm{T}} (\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{k}(\boldsymbol{x}), \end{aligned} \tag{2}$$

where $\boldsymbol{k}(\boldsymbol{x})$ is the covariance between $\boldsymbol{x}$ and $\boldsymbol{x}_{1:n} \in \mathcal{D}$. The posterior functions will guide the selection for the next data point to be observed.

A key challenge for the acquisition function is to trade off exploitation and exploration. Exploitation means sampling the next data point in which the prediction of the surrogate model is high, while exploration means sampling the next data point in which the surrogate model is highly uncertain. That

is, exploration encourages exploring the objective function space and exploitation encourages finding a high objective. Common mechanisms for acquisition function include expected improvement (EI) [23], probability of improvement [24], Thompson sampling [25], and entropy search portfolio [26].

The EI is a widely applicable mechanism for acquisition function. The EI is defined as

$$\mathrm{EI}_{y^t}(\boldsymbol{x}) := \int_{-\infty}^{\infty} \max(y^t - y, 0) p_M(y|x) \mathrm{d}y, \tag{3}$$

where y^t is a designated threshold, and M is some model for black-box function $f : \mathcal{X} \to \mathbb{R}^N$. EI is the expectation under model M that f exceeds the threshold y^t. The criteria of the EI mechanism is to optimize EI to approximate $f(x)$.

2.3 Problem formulation

The goal of our framework is to optimize the post-layout performance in the circuit sizing stage. The measurement of analog circuit performance often involves multiple performance metrics. Thus, the modeling of the analog circuit sizing can be seen as a multi-objective constrained optimization. To better formulate the sizing problem, we adopt the figure-of-merit (FOM) representation. We describe FOM formulation for analog sizing as follows:

$$\begin{aligned} \max_{P} \quad & \mathcal{FOM}(P) = \sum_i \alpha_i f_i(P) \\ \text{s.t.} \quad & \mathrm{thres}^j_{\mathrm{low}} \leqslant f_j(P) \leqslant \mathrm{thres}^j_{\mathrm{high}}, \end{aligned} \tag{4}$$

where P represents the sizing configuration, $f_i(\cdot)$ and $f_j(\cdot)$ denote the metrics of post-layout performance, α_i is the corresponding coefficient for the ith metric term in the FOM, and thres^j are thresholds to constrain the jth metric. Each sizing configuration P represents a set of assignments for all sizing parameters, including the number of fingers and finger width for each transistor. We define the sizing parameter space as the high-dimensional space of all feasible assignments for sizing parameters, and the sizing configuration is sampled from the sizing parameter space. The proposed framework aims to maximize $\mathcal{FOM}$ without constraint violation over sizing parameters and finally provides a good sizing solution that each metric performs well in post-layout simulation.

3 Algorithm

In this section, we introduce our analog circuit sizing framework driven by post-layout simulation. The framework builds up a compact closure considering the post-layout simulation at early sizing stage. We take advantage of automatic layout generation to integrate the post-layout simulation into the closure. For efficiency considerations, we utilize pre-layout simulation to compensate for the time-consuming post-simulation loop. We design a coupled Bayesian optimization that makes use of pre-layout simulation results to explore the parameter space and apply post-layout simulation results to guide the best FOM exploiting. As analog circuit sizing involves discrete parameters such as the number of fingers for a transistor, we consider the Bayesian estimator for mixed-variable optimization. We will detail the proposed framework in the following subsections.

3.1 Overview of our framework

Figure 2 shows the workflow of our framework. Different from previous analog sizing work, we propose a novel coupled Bayesian optimization paradigm. Our framework combines two models named Bayes-α and Bayes-β. Bayes-α consists of a tree-structured Parzen estimator (TPE) and a GP. Bayes-α applies Bayesian optimization on the black-box function of pre-layout simulation. Bayes-β attempts to optimize the black-box function of a more complex procedure with an automatic layout generator and post-layout simulation in it. The intrinsics of the two black-box functions are the pre-layout performance and post-layout performance correspondingly.

Our framework combines the two GP-based Bayesian optimization parts of Bayes-α and Bayes-β in a way we call coupled Bayesian optimization. We propose the concept of coupled Bayesian optimization.

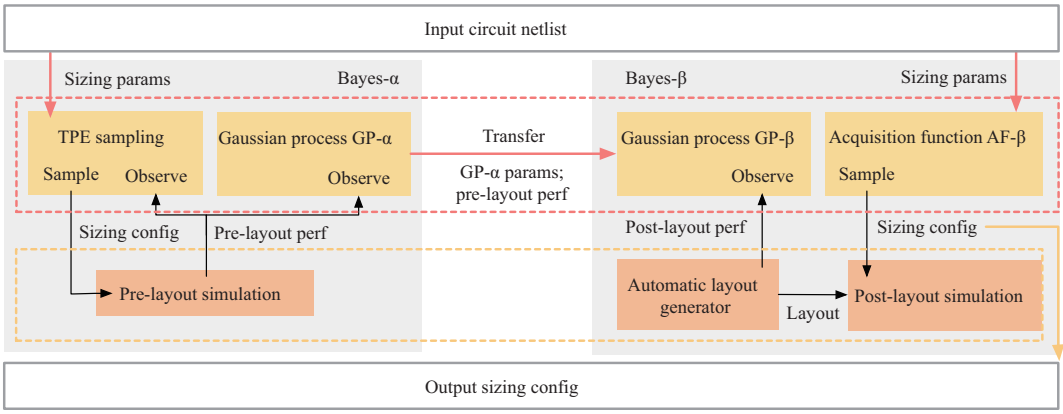

Figure 2 (Color online) Overall flow of our proposed framework.

Definition 1. A group of Bayesian optimization models is said to be coupled if

(1) they share the same form of acquisition functions;

(2) their surrogate models can be seen as generating from the same rendering process with different values of hyperparameters.

To be more specific, in our framework, Bayes-α and Bayes-β are coupled, as we can transfer the belief that surrogate model GP^{α} gains from the pre-layout performance black-box function to the domain of surrogate model GP^{β}. That is, we can make use of the observed FOM values in pre-layout simulation to benefit the sampling in Bayesian optimization for post-layout performance. We will discuss more about how to realize this powerful property in Subsection 3.2.

As we mentioned in Subsection 2.2, a well-designed acquisition function is supposed to trade off exploration and exploitation in a balanced way. For two coupled Bayesian optimization models, there is a different way to trade off exploration and exploitation. Our framework makes Bayes-α explore the sizing parameter space and Bayes-β focus on exploiting the higher FOM values. We will tackle sampling from the mixed parameter space for Bayes-α in Subsection 3.3. Furthermore, we show how we do efficient exploration with a TPE for Bayes-α in Subsection 3.4.

As the coupled Bayesian optimization is an iterative process, our framework conducts iterations with the two GPs GP^{α} and GP^{β}. Frequent calls of post-layout simulation can be extremely time-consuming, and therefore we set a variable interval to control how many iterations we run one single post-layout simulation loop. Before sampling from Bayes-β, we update Bayes-β with Bayes-α.

3.2 Transfer surrogate model

As the computation cost of automatic layout generation and post-layout simulation is far greater than pre-layout simulation, our framework explores how to make less observation in Bayes-β but manages to approximate the best FOM of post-layout simulation. To benefit the Bayes-β from the richer sampling points but not accurate simulation results in Bayes-α, we explore the transfer of the surrogate model in Bayes-α to the one in Bayes-β.

Bayes-α and Bayes-β have their own GP-based surrogate model, and the GP GP^{α} of Bayes-α gains the belief about the black-box function f^{α}. To clarify, we use the notation f^{α} for the black-box function of pre-layout simulation, and f^{β} for the black-box function of post-layout simulation. A naive strategy is to regard the GP^{α} and GP^{β} as two independent processes, but the strategy will not work due to the lack of observed data points for GP^{β}. We propose a strategy to gain belief about black-box f^{β} from both GP^{β} and GP^{α}.

Our strategy is inspired by the re-parameterization trick [27,28]. Note that for an arbitrary parameter configuration $\boldsymbol{x}$, the value $f^{\alpha}(\boldsymbol{x})$ and $f^{\beta}(\boldsymbol{x})$ are strongly correlated in some way unknown. We assume that they are jointly distributed as

$$\begin{bmatrix} f^{\alpha}(\boldsymbol{x}) \\ f^{\beta}(\boldsymbol{x}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu^{\alpha} \\ \mu^{\beta} \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}^{\alpha} & \boldsymbol{K}^{\alpha,\beta} \\ \boldsymbol{K}^{\beta,\alpha} & \boldsymbol{K}^{\beta} \end{bmatrix} \right), \tag{5}$$

where μ^α, μ^β are mean vectors, and $\boldsymbol{K}^\alpha$, $\boldsymbol{K}^\beta$ are GP kernels, defined in (2). $\boldsymbol{K}^{\alpha,\beta}$ and $\boldsymbol{K}^{\beta,\alpha}$ are covariance matrices between f^α and f^β.

Consider GP^α and GP^β as two models generated by the same sampler from two different sets of hyperparameters. We adapt the re-parameterization trick to formulate the posterior distribution [27]. For a new observation of parameter configuration $(\boldsymbol{x}, f^\alpha(\boldsymbol{x}), f^\beta(\boldsymbol{x}))$, we first update $\boldsymbol{K}$ with general Bayesian optimization strategy, and then update $\boldsymbol{K}^\beta$ for GP^β:

$$K^\beta = K^\beta - K^{\beta,\alpha}(K^\alpha)^{-1}K^{\alpha,\beta}. \tag{6}$$

3.3 Tackle mixed-variable optimization

Traditional Bayesian optimization methods usually deal with black-box functions with continuous variables. Analog circuit sizing faces optimizing continuous variables and discrete variables simultaneously. The Bayesian optimization does not naturally support mixed-variable optimization. To enhance the optimization process, we design our acquisition function to sample from both continuous and discrete variables, based on the TPE approach.

The TPE approach derives from the EI optimization scheme [18]. As we mentioned before, the Gaussian-process-based method models $p(y|\boldsymbol{x})$, and furthermore TPE models both $p(y|\boldsymbol{x})$ and the distribution of objective function values $p(y)$. We inherit the replacing strategy from the TPE that models $p(y|\boldsymbol{x})$ by replacing the prior distributions on sizing parameter space with non-parametric densities. The most used parameters like the number of fingers and finger width can be described with uniform variables and quantized uniform variables. Under the replacing strategy, for example, we can describe a parameter of finger width as a truncated Gaussian mixture. Therefore, we can produce a variety of densities over the sizing parameter space $\mathcal{X}$ and we define two densities for sizing parameter space:

$$\begin{aligned} \ell(\boldsymbol{x}) =& \frac{\partial x * \boldsymbol{P}^\ell}{\partial \mu_{\mathcal{X}}}, \quad x \in \{x^i | f(x^i) < y^*\}, \\ g(\boldsymbol{x}) =& \frac{\partial x * \boldsymbol{P}^g}{\partial \mu_{\mathcal{X}}}, \quad x \in \{x^i | f(x^i) \geqslant y^*\}, \end{aligned} \tag{7}$$

where y^* is the best FOM value observed yet, $\ell(\boldsymbol{x})$ represents the density induced from the observations $\{x^i\} \in \mathcal{X}$ whose prediction is less than y^*, $g(\boldsymbol{x})$ represents the density induced from other observed x^i. $x * \boldsymbol{P}^\ell$ is the probability measure on measurable space $(\mathcal{X}, \mathcal{A}^\ell)$, $\mathcal{A}^\ell := \{x^i | f(x^i) \leqslant y^*\}$ and $\mu_{\mathcal{X}}$ is a reference measure. Eq. (7) defines how we get the two densities $\ell(\boldsymbol{x})$ and $g(\boldsymbol{x})$ by Radon-Nikodym derivative. We adopt the formulation from [29] to numerically calculate the $\ell(\boldsymbol{x})$ and $g(\boldsymbol{x})$. Here we gives how we calculate $\ell(\boldsymbol{x})$ as example

$$\ell(\boldsymbol{x}) = \frac{1}{n^\ell h(n^\ell)} \sum_{x^i} K\left(\frac{\boldsymbol{x} - x^i}{h(n^\ell)}\right), \quad x^i \in \mathcal{A}^\ell, \tag{8}$$

where n^ℓ stands for the size of set $\mathcal{A}^\ell$, $K(\cdot)$ is a real-valued Borel function, and $h(\cdot)$ is a sequence of positive real numbers, and they satisfy the conditions described in [29]. As we can calculate (8) for both continuous and discrete variables, the density definition is applicable to all parameter spaces met in the analog circuit sizing problem. We will continue the subsequent deduction for acquisition function design in the next subsection.

3.4 Control the acquisition function

As we expect Bayes-α to explore more parameter space, a controlling strategy for TPE is desired. We adapt the TPE [18] to control whether to sample aggressively. Sampling aggressively implies more exploitation and sampling less aggressively indicates more exploration. Following (7) and (8), Then we can define the $p(\boldsymbol{x}|y)$ on sizing parameter space using the pre-defined $\ell(\boldsymbol{x})$ and $g(\boldsymbol{x})$

$$p(y|\boldsymbol{x}) = \begin{cases} \ell(\boldsymbol{x}), & y < y^*, \\ g(\boldsymbol{x}), & y \geqslant y^*. \end{cases} \tag{9}$$

Under the assumption of (9), we can design acquisition functions for our framework to optimize the EI introduced in (3). By a simple application of the Bayesian theorem, we derive

$$\mathrm{EI}_{y^*}(\boldsymbol{x}) = \int_{-\infty}^{y^*} (y^* - y) \frac{p(\boldsymbol{x}|y)p(y)}{p(\boldsymbol{x})} \mathrm{d}y, \tag{10}$$

here we take y^* for the threshold y^t. As the marginalization property indicates, we have

$$p(\boldsymbol{x}) = \int_{\mathbb{R}} p(\boldsymbol{x}|y)p(y)\mathrm{d}y, \tag{11}$$

from which we deduce that

$$\begin{aligned} \int_{-\infty}^{y^*} (y^* - y)p(\boldsymbol{x}|y)p(y)\mathrm{d}y &= \ell(\boldsymbol{x}) \int_{-\infty}^{y^*} (y^* - y)p(y)\mathrm{d}y \\ &= p(y < y^*)y^*\ell(\boldsymbol{x}) - \ell(\boldsymbol{x}) \int_{-\infty}^{y^*} p(y)\mathrm{d}y. \end{aligned} \tag{12}$$

Therefore, we get the proportional formulation

$$\mathrm{EI}_{y^*}(\boldsymbol{x}) = \frac{p(y < y^*)y^*\ell(\boldsymbol{x}) - \ell(\boldsymbol{x}) \int_{-\infty}^{y^*} p(y)\mathrm{d}y}{p(y < y^*)\ell(\boldsymbol{x}) + p(y \geqslant y^*)g(\boldsymbol{x})}, \tag{13}$$

$$\mathrm{EI}_{y^*}(\boldsymbol{x}) \propto \left(p(y < y^*) + \frac{g(\boldsymbol{x})}{\ell(\boldsymbol{x})p(y \geqslant y^*)} \right)^{-1}. \tag{14}$$

From (14), we know that our proposed acquisition function will be more likely to sample the next sizing parameter configuration $\boldsymbol{x}$ from parameter space $\mathcal{X}$ with higher $\frac{g(\boldsymbol{x})}{\ell(\boldsymbol{x})}$.

As we discussed before, we expect the Bayesian optimization Bayes-α to explore more about the parameter space. We assign a less aggressive threshold, that is, a small y^* for Bayes-α, to encourage exploring more parameter space.

4 Experimental results

In this section, we demonstrate the efficiency of our proposed framework. We conduct experiments on two real-world analog circuits, including an inverter-based operational transconductance amplifier (OTA) and a low-dropout regulator (LDO). The two circuits are designed under TSMC 40 nm technology. We conduct our experiments on a CentOS workstation with an Intel Xeon Gold 5218R CPU and 128 GB memory. We adopt the MAGICAL [4, 30, 31] as our automatic layout generator. All the pre-layout simulation results and post-layout simulation results are generated from Cadence Virtuoso and Mentor Graphics Calibre.

We implement a baseline considering pre-layout simulation (denoted as Plain) and another baseline considering both pre-layout simulation and post-layout simulation (denoted as BMF). The former adopts the basic Bayesian optimization method from [5]. The latter applies the Bayesian model fusion technique [16] to train a post-layout performance Bayesian model and then uses the Bayesian model to find a sizing configuration. We set the number of post-layout training samples the same as the post-layout simulation iterations. The number of iterations in the follow-up Bayesian optimization also takes the same value. We list the specific values in Subsections 4.1 and 4.2.

We estimate the runtime by program execution time other than CPU time. We demonstrate the execution pattern of our optimization flow with Figure 3. Bayes-α and Bayes-β run asynchronously in most of the time. Bayes-β messages Bayes-α a sampled sizing configuration at the beginning of a Bayes-β iteration. Bayes-α messages Bayes-β the parameters of GP^{α} and the pre-layout simulation result of the sampled sizing configuration. As for BMF [16], we first run pre-layout simulations and select a part of sizing configurations with top performance for the training, which is a sequential execution pattern.

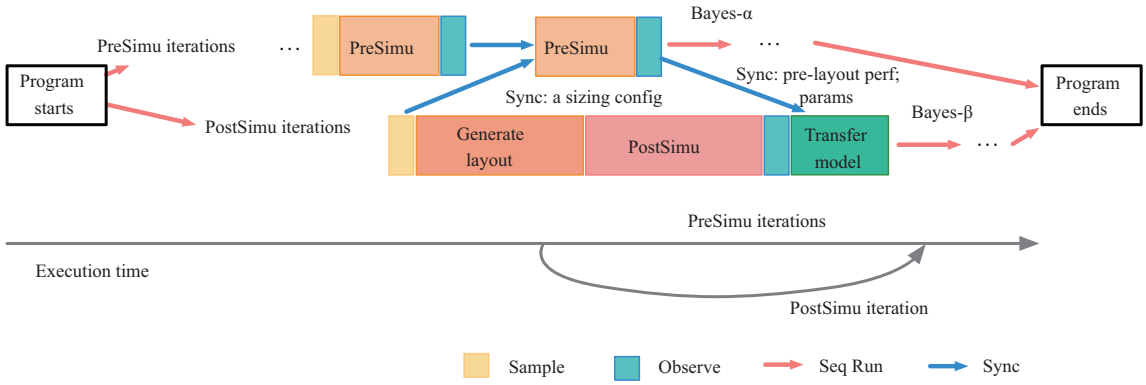

Figure 3 (Color online) Asynchronous execution pattern of our optimization flow with pre-layout simulation and post-layout simulation overlapping on the execution time.

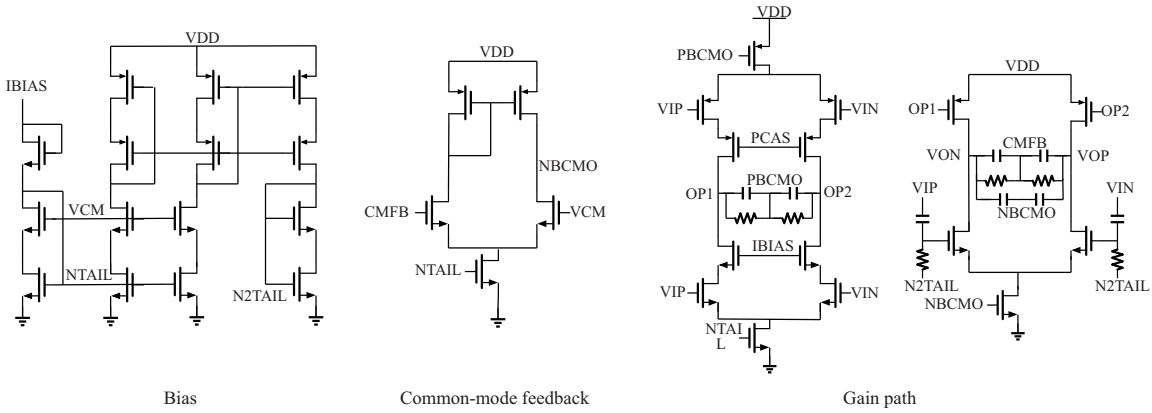

Figure 4 Schematic of the inverter-based OTA.

4.1 Inverter-based OTA

The schematic of the OTA is shown in Figure 4. This circuit is reproduced from [4], which is a taped-out case.

We perform symmetry detection for this OTA circuit [32] and reduce the design parameters to 20 for the sizing process. To meet a reasonable design requirement, we construct the specific form of FOM for this circuit:

$$\begin{aligned} \max_{P} \quad & \alpha_1 \mathrm{Gain} + \alpha_2 \lg \mathrm{UGB} \\ \text{s.t.} \quad & \mathrm{thres}_{\mathrm{low}} \leqslant \mathrm{PM} \leqslant \mathrm{thres}_{\mathrm{high}}, \end{aligned} \tag{15}$$

where Gain denotes the close loop gain, UGB stands for the unity gain bandwidth, and PM represents the phase margin. As Gain involves logarithmic calculation, we add a logarithmic form for the other term UGB to normalize the optimization objective. In this experimental setup, we consider Gain and UGB to be equally important, and thus we set their coefficient α_1 and α_2 to 1.0, respectively. To represent the constraint on phase margin, we introduce a penalty term to the optimization objective. To guarantee the circuit functionality, the phase margin is supposed to range from 45 to 80, and therefore we set $\mathrm{thres}_{\mathrm{low}} = 45$ and $\mathrm{thres}_{\mathrm{high}} = 80$.

The penalty term is defined as

$$\mathrm{penalty} = \max(\mathrm{thres}_{\mathrm{high}}, \mathrm{PM}) - \min(\mathrm{thres}_{\mathrm{low}}, \mathrm{PM}). \tag{16}$$

Plain is the basic Bayesian optimization method on pre-layout simulation results [5]. BMF-16 and BMF-32 is the Bayesian model fusion which employs 16 and 32 post-layout training samples, respectively. The iterations on the predicted Bayesian model also take 16 and 32, respectively. Manual is an open-source taped-out case from MAGICAL [4]. In Table 1, Ours-10 and Ours-5 are our methods of coupled Bayesian optimization. The x of Ours-x represents the interval that we run post-layout simulation once

Table 1 FOM results and post-simulation performance of the inverted-based OTA

Performance	Plain [5]	BMF-16 [16]	BMF-32 [16]	Manual [4]	Ours-10	Ours-5
Gain (dB)	17.09	17.19	20.34	14.15	19.89	**20.52**
UGB (MHz)	7.11	7.35	8.04	5.11	8.54	**9.03**
PM (°)	45<, <80	45<, <80	45<, <80	45<, <80	45<, <80	45<, <80
FOM	32.86	33.00	36.24	29.59	35.85	**36.53**
#PreSimu	160	160	160	–	160	160
#PostSimu	–	16	32	–	16	32
Runtime	87 min 2 s	140 min 5 s	198 min 10 s	–	95 min 13 s	104 min 27 s

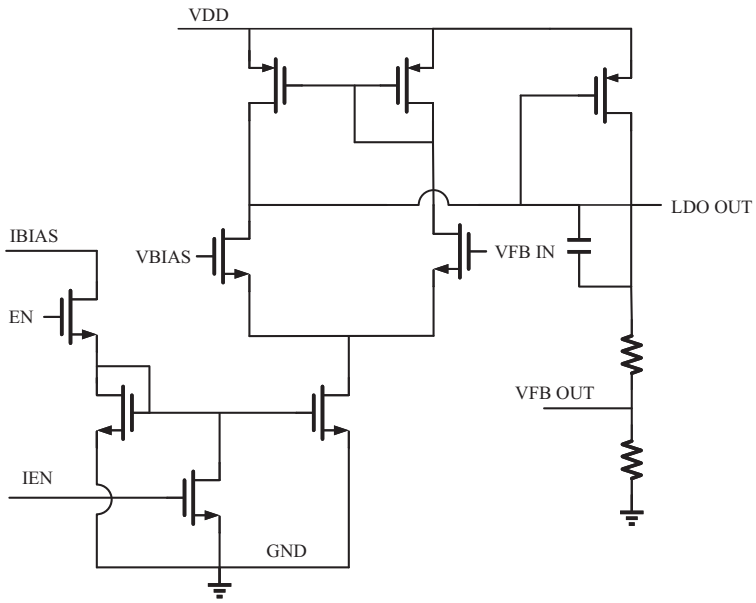

Figure 5 Schematic of the LDO.

every x runs of the pre-layout simulation. We restrict the maximum number of iterations to 160 and initialize the optimization process with random starting points. By iteration, we mean one single loop of sampling a sizing configuration, measuring by simulation, and updating our model. As we know from Figure 3, the runtime of post-layout iteration could overlap with the runtime of several pre-layout iterations. We estimate the time of pre-layout simulation (25 ± 5 s), the time of layout generation (70 $\pm$50 s), the time of post-layout simulation (80 ± 15 s), the time of sample&observe ($\leqslant 5$ s), and the time of transferring model ($\leqslant 5$ s) for one iteration. We set the timeout as 240 s for the layout generator.

Table 1 summarizes the post-layout performance of the inverter-based OTA circuit. We present the performance with gain, unity gain bandwidth, and phase margin metrics. #PreSimu and #PostSimu represent the number of runs of pre-layout simulation and post-layout simulation. Both Ours-10 and Ours-5 outperform Plain and manually sized design. Ours-10 outperforms BMF-16 and Ours-5 outperforms BMF-32. Our framework satisfies the constraint on phase margin. Ours-5 achieves a 45% improvement in gain, a 76% improvement in unity gain bandwidth, compared with manual sizing configuration, and a 20% improvement in gain, a 27% improvement in unity gain bandwidth, compared with Plain with Bayesian optimization on pre-layout simulation.

4.2 LDO

The schematic of the LDO is shown in Figure 5. This circuit is designed by an experienced designer. We use 12 design parameters for sizing this circuit. The specific form of FOM for this LDO circuit is defined as

$$\begin{aligned} \max_{P} \quad & \alpha_1 \mathrm{Gain} + \alpha_2 V_{\mathrm{OU}} \\ \text{s.t.} \quad & \mathrm{thres}_{\mathrm{low}} \leqslant \mathrm{PM} \leqslant \mathrm{thres}_{\mathrm{high}}, \end{aligned} \tag{17}$$

Table 2 FOM results and post-simulation performance of the LDO

Performance	Plain [5]	BMF-30 [16]	BMF-60 [16]	Manual	Ours-4	Ours-2
Gain (dB)	70.16	72.21	72.71	73.22	**73.45**	72.15
OU (V)	1.40	0.73	0.36	1.43	0.47	**0.28**
PM (°)	60<, <90	60<, <90	60<, <90	60<, <90	60<, <90	60<, <90
FOM	−6.98	−0.08	3.63	−6.98	2.65	**4.42**
#PreSimu	120	120	120	–	120	120
#PostSimu	–	30	60	–	30	60
Runtime	54 min 15 s	99 min 2 s	143 min 52 s	–	96 min 30 s	105 min 50 s

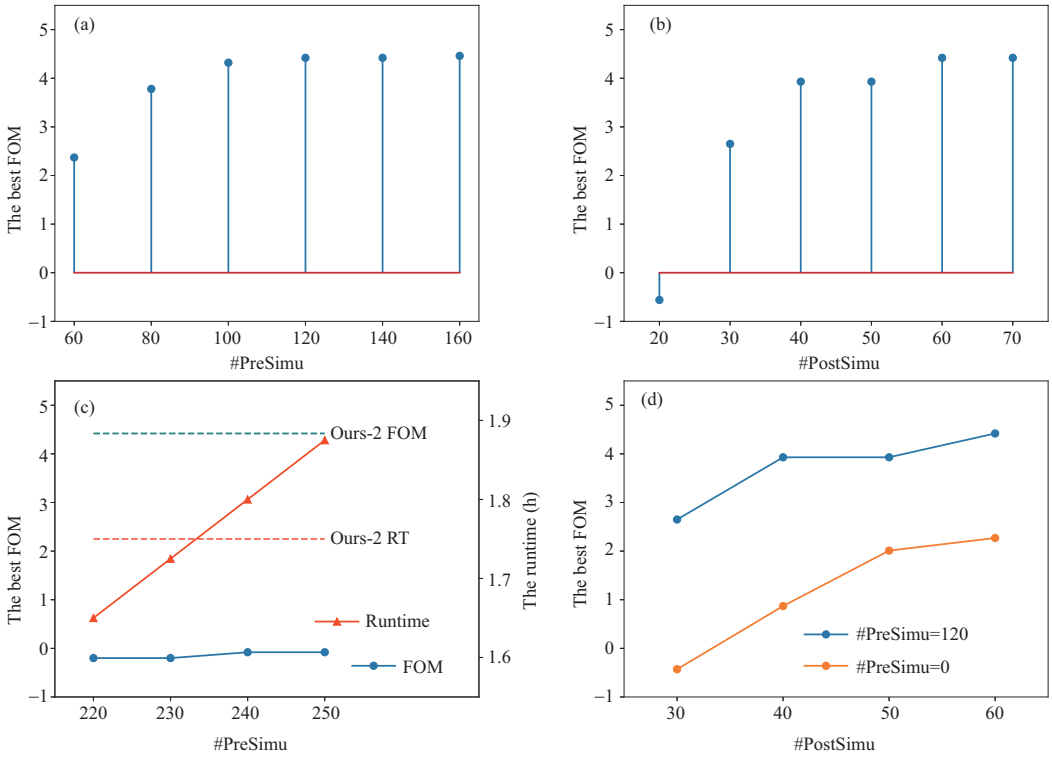

Figure 6 (Color online) Best FOM results of different #PreSimu and #PostSimu settings. (a) Different #PreSimu when #PostSimu = 60; (b) different #PostSimu when #PreSimu = 120; (c) different #PreSimu when #PostSimu = 0; (d) comparing #PreSimu = 120 and #PreSimu = 0.

where Gain denotes the close-loop gain, PM denotes the phase margin, V_{OU} stands for overshot up voltage. As the design is supposed to decrease the overshot up voltage, we assign a negative coefficient for the term. To be more specific, we take $\alpha_1 = 0.1$ and $\alpha_2 = -10.0$ so that the two terms are of the same order of magnitude. The penalty for phase margin constraint follows the formulation shown in (16), in which $\mathrm{thres}_{\mathrm{low}}$ takes 60 and $\mathrm{thres}_{\mathrm{high}}$ takes 90.

Table 2 shows the results of post-layout performance for the LDO circuit. Here we demonstrate Ours-4 and Ours-2 for this case, which run one post-layout simulation per 4 and 2 iterations, respectively. We choose smaller intervals 4 and 2 for this case because there is a large gap between the distributions of the pre-layout simulation and the post-layout simulation of the LDO. Similar to the first case, we compare our framework with the manual sizing configuration Manual and the baseline method Plain, BMF-30, BMF-60. Here BMF-30 and BMF-60 take 30 and 60 post-layout training samples, respectively. We estimate the time of pre-layout simulation (20 ± 5 s), the time of layout generation (20 ± 15 s), the time of post-layout simulation (65 ± 15 s), the time of sample&observe ($\leqslant 5$ s), and the time of transferring model ($\leqslant 5$ s) for one iteration. As shown in the Table 2, both Ours-4 and Ours-2 achieve better performance in multiple metrics. Ours-4 achieves the best value of gain. Ours-2 obtains the best value of overshot up voltage. Taken together, our framework outperforms the manual sizing configuration and basic Bayesian optimization in FOM values within a reasonable runtime.

Figure 6 shows the performance of our method with respect to different #PreSimu and #PostSimu set-

tings. Figure 6(a) shows the best FOM results with different pre-layout iterations when fixing #PostSimu = 60, and Figure 6(b) shows the results with different post-layout iterations when fixing #PreSimu = 120. Both pre-layout and post-layout iterations contribute to the post-layout performance. Figure 6(c) shows the performance of Plain with different pre-layout iterations when their runtime is close to Ours-2. We can see that barely with pre-layout simulation, the post-layout performance saturates quickly even given more simulations. Figure 6(d) compares the performance over post-layout iterations between #PreSimu = 0 and #PreSimu = 120. Note that the blue line shares the same data with Figure 6(b). We can see that combining post-layout simulation with pre-layout simulation can significantly improve the efficiency of searching for better sizing configuration. With these experiments, we conclude that incorporating pre-layout simulation and post-layout simulation can improve both the quality of performance and efficiency. Meanwhile, our proposed algorithm can synergistically balance amounts of pre-layout simulations and post-layout simulations in an asynchronous pattern.

5 Conclusion

In this paper, we propose a post-layout-simulation-driven analog circuit sizing framework. By integrating the automatic layout generator and post-layout simulation into the circuit sizing loop, we develop a method that directly optimizes the post-layout performance. Our framework leverages a coupled Bayesian optimization technique, to represent a group of strongly correlated Bayesian optimization models satisfying special properties. The coupled Bayesian optimization helps our framework trade-off exploration and exploitation. Our framework takes advantage of the efficiency of pre-layout simulation to explore more sizing parameter space and the post-layout simulation to exploit parameter configuration which is likely to induce a high FOM value. Compared with the Bayesian optimization on pre-layout simulation alone and the manual sizing results, our framework is more promising for generating sizing configuration with better post-layout performance. In future work, we can further enhance the efficiency of our proposed framework.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 62141404, 62034007) and 111 Project (Grant No. B18001). We would like to thank Kenuo XU for drawing Figure 1.

References

1 Kunal K, Madhusudan M, Sharma A K, et al. Align: open-source analog layout automation from the ground up. In: Proceedings of the Design Automation Conference, 2019. 1–4
2 Crossley J, Puggelli A, Le H P, et al. BAG: a designer-oriented integrated framework for the development of AMS circuit generators. In: Proceedings of the International Conference on Computer-Aided Design, 2013. 74–81
3 Chang E, Han J, Bae W, et al. BAG2: a process-portable framework for generator-based AMS circuit design. In: Proceedings of the IEEE Custom Integrated Circuits Conference (CICC), 2018. 1–8
4 Chen H, Liu M J, Tang X Y, et al. Magical 1.0: an open-source fully-automated AMS layout synthesis framework verified with a 40-nm 1 GS/s $\delta\sigma$ ADC. In: Proceedings of the IEEE Custom Integrated Circuits Conference (CICC), 2021. 1–2
5 Liu M J, Turner W J, Kokai G F, et al. Parasitic-aware analog circuit sizing with graph neural networks and Bayesian optimization. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, 2021. 1372–1377
6 Ranjan M, Verhaegen W, Agarwal A, et al. Fast, layout-inclusive analog circuit synthesis using pre-compiled parasitic-aware symbolic performance models. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, 2004. 604–609
7 Hakhamaneshi K, Werblun N, Abbeel P, et al. BagNet: berkeley analog generator with layout optimizer boosted with deep neural networks. In: Proceedings of the International Conference on Computer-Aided Design, 2019. 1–8
8 Settaluri K, Haj-Ali A, Huang Q J, et al. AutoCkt: deep reinforcement learning of analog circuit designs. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, 2020. 490–495
9 Bengio Y, Goodfellow I, Courville A. Deep Learning. Cambridge: MIT press, 2017
10 Chai Z M, Zhao Y X, Lin Y B, et al. CircuitNet: an open-source dataset for machine learning applications in electronic design automation (EDA). Sci China Inf Sci, 2022, 65: 227401
11 Boyd S P, Lee T H, et al. Optimal design of a CMOS OP-AMP via geometric programming. IEEE TCAD, 2001, 20: 1–21
12 Wang H R, Wang K, Yang J C, et al. GCN-RL circuit designer: transferable transistor sizing with graph neural networks and reinforcement learning. In: Proceedings of the Design Automation Conference, 2020. 1–6
13 Li Y G, Lin Y S, Madhusudan M, et al. A circuit attention network-based actor-critic learning approach to robust analog transistor sizing. In: Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD (MLCAD), 2021. 1–6
14 Zhang S H, Lyu W L, Yang F, et al. An efficient multi-fidelity Bayesian optimization approach for analog circuit synthesis. In: Proceedings of the Design Automation Conference, 2019. 1–6
15 Zhang S H, Yang F, Zhou D, et al. An efficient asynchronous batch Bayesian optimization approach for analog circuit synthesis. In: Proceedings of the Design Automation Conference, 2020. 1–6
16 Wang F, Zhang W Y, Sun S P, et al. Bayesian model fusion: large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data. In: Proceedings of the Design Automation Conference, 2013. 1–6
17 Liu J Z, Hassanpourghadi M, Zhang Q C, et al. Transfer learning with Bayesian optimization-aided sampling for efficient ams circuit modeling. In: Proceedings of the International Conference on Computer-Aided Design, 2020. 1–9

18 Bergstra J, Bardenet R, Bengio Y, et al. Algorithms for hyper-parameter optimization. In: Proceedings of the Advances in Neural Information Processing Systems, 2011, 24

19 Snoek J, Larochelle H, Adams R P. Practical Bayesian optimization of machine learning algorithms. In: Proceedings of the Advances in Neural Information Processing Systems, 2012

20 Shahriari B, Swersky K, Wang Z Y, et al. Taking the human out of the loop: a review of Bayesian optimization. Proc IEEE, 2015, 104: 148–175

21 Liu Y R, Hu Y Q, Qian H, et al. ZOOpt: a toolbox for derivative-free optimization. Sci China Inf Sci, 2022, 65: 207101

22 Rasmussen C E. Gaussian processes in machine learning. In: Proceedings of the Summer School on Machine Learning, 2003. 63–71

23 Mockus J, Tiesis V, Zilinskas A. The application of Bayesian methods for seeking the extremum. Toward Global Opt, 1978, 2: 117–129

24 Kushner H J. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. J Basic Eng, 1964, 86: 97–106

25 Thompson W R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika, 1933, 25: 285–294

26 Hennig P, Schuler C J. Entropy search for information-efficient global optimization. J Mach Learn Res, 2012, 13: 1809–1837

27 Kingma D P, Welling M. Auto-encoding variational bayes. 2013. ArXiv:1312.6114

28 Rezende D J, Mohamed S, Wierstra D. Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the International Conference on Machine Learning, 2014. 1278–1286

29 Srivastava R C. Estimation of probability density function based on random number of observations with applications. Int Stat Rev Revue Inte de Stat, 1973, 41: 77

30 Zhu K R, Chen H, Liu M J, et al. Effective analog/mixed-signal circuit placement considering system signal flow. In: Proceedings of the International Conference on Computer-Aided Design, 2020

31 Chen H, Zhu K R, Liu M J, et al. Toward silicon-proven detailed routing for analog and mixed-signal circuits. In: Proceedings of the International Conference on Computer-Aided Design, 2020

32 Liu M J, Li W X, Zhu K R, et al. S3DET: detecting system symmetry constraints for analog circuits with graph similarity. In: Proceedings of the 25th Asia and South Pacific Design Automation Conference, 2020. 193–198