tSS-BO: Scalable Bayesian Optimization for Analog Circuit Sizing via Truncated Subspace Sampling

Tianchen Gu[1], Jiaqi Wang[2], Zhaori Bi[1,*], Changhao Yan[1], Fan Yang[1], Yajie Qin[3], Tao Cui[2] and Xuan Zeng[1,*]
[1] State Key Laboratory of Integrated Chips and Systems, School of Microelectronics, Fudan University, Shanghai, China
[2] State Key Laboratory of Scientific and Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences & University of Chinese Academy of Sciences, Beijing, China
[3] State Key Laboratory of Integrated Chips and Systems, School of Information Science and Technology, Fudan University, Shanghai, China

***Abstract*—We propose a novel scalable Bayesian optimization method with truncated subspace sampling (tSS-BO) to tackle high-dimensional optimization challenges for large-scale analog circuit sizing. To address the high-dimensional challenges, we propose subspace sampling subject to a truncated Gaussian distribution. This approach limits the effective sampling dimensionality down to a constant upper bound, independent of the original dimensionality, leading to a significant reduction in complexity associated with the curse of dimensionality. The distribution covariance is iteratively updated using a truncated flow, where approximate gradients and center steps are integrated with decaying prior subspace features. We introduce gradient sketching and local Gaussian process (GP) models to approximate gradients without additional simulations to mitigate systematic errors. To enhance efficiency and ensure compatibility with constraints, we utilize local GP models for the selection of promising candidates, avoiding the cost of acquisition function optimization. The proposed tSS-BO method exhibits clear advantages over state-of-the-art methods in experimental comparisons. In synthetic benchmark functions, the tSS-BO method achieves up to $4.93\times$ evaluation speedups and a remarkable over $30\times$ algorithm complexity reduction compared to the Bayesian baseline. In real-world analog circuits, our method achieves up to $2\times$ speedups in simulation number and runtime.**

***Index Terms*—Analog Circuit Sizing, High-dimensional Optimization, Bayesian Optimization, Subspace Technique, Gradient Sketching**

I. Introduction

With the development of integrated chips, the scale of circuits exponentially increases. While automatic tools for digital circuit synthesis are well-established, the automated design tools of analog circuits have not yet been applied on a large scale. Therefore, the analog circuit design is the bottleneck of integrated circuits and systems design [1]. In addition to the expensive simulations, nonlinear circuit performances and complicated performance trade-offs problems for traditional analog circuit design, today's large-scale analog circuits with a large number of design parameters also presents the high-dimensional optimization problems.

The conventional approaches for analog circuit optimizations include accurate model-based methods [2], simulation-based stochastic methods [3] and up-to-date online model-based methods [4]–[9]. These methods are mainly designed for nonlinear optimization problems in analog circuit sizing with few parameters, which can be hardly applied on the large-scale analog circuits with a large number of parameters.

Due to the curse of dimensionality, the design space expands much faster than the simulation budgets. State-of-the-art high-dimensional Bayesian optimization (BO) method TuRBO [10] employs local BO with the trust region technique to address this challenge by enhancing the local exploitation, which has found applications in analog circuit sizing [11], [12]. However, TuRBO's exploratory capacity is confined by trust region, leading to sub-optimal solutions.

Another popular method yielding robust performance in high-dimensional problems is the evolutionary strategy with covariance matrix adaptation (CMA-ES) [13]. By approximating the inverse Hessian matrix with a covariance matrix, CMA-ES adopts a Gaussian distribution to generate a population around the current center. The center and covariance matrix evolve based on these random samples. However, CMA-ES's stochastic sampling approach can hinder convergence efficiency and often demands a large number of simulations.

In addition to the mentioned limitations of the above high-dimensional approaches, another drawback hinders their efficiency in higher dimensions. Both TuRBO and CMA-ES select candidates in the full original dimensions, which becomes inefficient when dimensionality increasing. A possible solution to this problem can be the subspace technique pioneered by Yuan [14], who introduced a 2-dimensional subspace based on gradients and last descent directions. Despite its promise, this subspace technique has not been broadly used in circuit problems. This is primarily due to the cost asscociated with gradient calculation and the loss of high-dimensional information because of the fixed subspace dimensionality.

To address the limitations above, we propose a novel scalable Bayesian optimization with truncated subspace sampling (tSS-BO) for high-dimensional optimization in large-scale analog circuit sizing. By employing the truncated subspace sampling technique with a truncated Gaussian distribution, we adaptively reduce the effective sampling dimensionality way down to a constant upper bound and enhance the 2-dimensional subspace technique by incorporating approximate gradients and update steps with decaying prior features. The BO candidates are selected from the subspace samples guided by local Gaussian

*Corresponding authors: {zhaori_bi, xzeng}@fudan.edu.cn

979-8-3503-4859-0/DATE24/© 2024 EDAA

process (GP) models and BO method to improve constraint feasibility and efficiency. Our local GP models maintain constant training complexity using a fixed number of training samples and the BO selection is computationally friendly without acquisition function optimization. Notably, to reduce the gradient approximate errors and to save simulations, gradients for subspace update are approximated through gradient sketching and through GP models without extra simulations.

The rest of this paper is organized as follows. Section II provides an overview of background knowledge. Section III introduces the proposed tSS-BO method. Section IV provides experiment results on synthetic benchmark functions and two real-world circuits and Section V concludes the paper.

II. Background

A. Problem Formulation

The analog circuit optimization problem can be formulated as a constrained optimization problem as follows,

$$\begin{aligned} &\underset{\boldsymbol{x}\in S}{\text{minimize}}\ f(\boldsymbol{x}), \\ &\text{s.t. } c^{(i)}(\boldsymbol{x}) \le 0, i = 1...N_c, \end{aligned} \tag{1}$$

where $S = [0,1]^d$ is the normalized design space and a circuit design $\boldsymbol{x}$ is presented with a d-dimensional vector. $f(\cdot)$ denotes the objective performance and $c^{(i)}(\cdot) \le 0$ represent the performance constraints. For constrained optimization problems, a normal approach is to convert the constrained optimization into an unconstrained Lagrangian optimization as follows,

$$\underset{\boldsymbol{x}\in S}{\text{minimize}}\ L(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{N_c} \omega_i \cdot \max(c^{(i)}(\boldsymbol{x}), 0), \tag{2}$$

where ω_i is the weight dealing with the unit difference.

B. Bayesian Optimization

BO is a global optimization method utilizing the probabilistic models and acquisition functions [15]. The model used here is the GP model specified by a Matern-5/2 kernel as follows,

$$k_{Mat52}(\boldsymbol{x}, \boldsymbol{x}') = \theta^2(1 + \sqrt{5}\|\boldsymbol{r}\| + \frac{5}{3}\boldsymbol{r}^2)\exp(-\|\frac{\sqrt{5}\boldsymbol{r}}{\boldsymbol{l}}\|), \tag{3}$$

where $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{x}'$, θ and $\boldsymbol{l}$ are hyper-parameters. The GP model gives the posterior distribution $\mathcal{N}(\mu, \sigma^2)$ at a point $\boldsymbol{x}^*$ with dataset $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, and a kernel $k(\cdot,\cdot)$ as follows,

$$\begin{aligned} \mu &= \boldsymbol{k}_* \mathbf{K}^{-1} \boldsymbol{y}, \\ \sigma^2 &= k_{**} - \boldsymbol{k}_* \mathbf{K}^{-1} \boldsymbol{k}_*^T, \end{aligned} \tag{4}$$

where $\mathbf{K}$ is an $N \times N$ covariance matrix with $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\boldsymbol{k}_*$ is an $1 \times N$ vector with $k_{*i} = k(\boldsymbol{x}^*, \boldsymbol{x}_i)$, and k_{**} is the intrinsic kernel. μ and σ^2 predict the expectation and variation.

BO methods select a candidate with the best utility inferred by acquisition functions. The weighted Expected Improvement (wEI) acquisition function estimates the expected improvement of a point considering probability of feasibility [5] as follows,

$$\begin{aligned} \alpha_{wEI}(\boldsymbol{x}) &= \alpha_{EI}(\boldsymbol{x}) \cdot \alpha_{PF}(\boldsymbol{x}), \\ \alpha_{EI}(\boldsymbol{x}) &= (\gamma - \mu_x)\Phi(\frac{\gamma - \mu_x}{\sigma_x}) + \sigma_x \phi(\frac{\gamma - \mu_x}{\sigma_x}), \\ \alpha_{PF}(\boldsymbol{x}) &= \prod_{i=1}^{N_c} \Phi(-\frac{\mu_c^{(i)}}{\sigma_c^{(i)}}), \end{aligned} \tag{5}$$

where μ_x and σ_x are the expectation and deviation of the objective performance, $\mu_c^{(i)}$ and $\sigma_c^{(i)}$ are the expectation and deviation of the i-th constraint, γ is the best objective performance of the feasible simulation points, $\Phi(\cdot)$ and $\phi(\cdot)$ are the CDF and PDF of a standard normal distribution.

C. Subspace Technique

The subspace definition from Yuan [14] is as follows,

$$S_k^s = span\{\boldsymbol{g}_k, \boldsymbol{s}_k\}, \tag{6}$$

where $\boldsymbol{g}_k$ is the gradient at the k-th iteration and $\boldsymbol{s}_k = \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$ is the last descent direction. The two directions are inspired by Conjugate Gradient methods since the conjugate gradient step is contained in this subspace. The subspace is established locally around a point $\boldsymbol{x}_k$ and the next descent step is got by optimizing the sub-problem in the subspace as follows,

$$\boldsymbol{x}_{k+1} := \underset{\boldsymbol{x}\in\boldsymbol{x}_k + S_k^s}{\arg\min}\ f(\boldsymbol{x}). \tag{7}$$

This 2-D subspace has been further extended into a more-dimensional form inspired by quasi-Newton methods [16],

$$S_k^s = span\{\boldsymbol{g}_k, \boldsymbol{g}_{k-1}, ..., \boldsymbol{s}_k, \boldsymbol{s}_{k-1}, ...\}. \tag{8}$$

III. Proposed Method

In this section, we present our approach: scalable Bayesian optimization with truncated subspace sampling (tSS-BO). The proposed method comprises three key steps: truncated subspace sampling, BO selection, and subspace update steps. At each iteration, we draw random subspace samples subject to a multivariate Gaussian distribution centered around $\boldsymbol{x}_k$ with covariance $\sigma_k^2 \boldsymbol{C}_k$, truncated by the design space. The covariance matrix is constructed with gradients and center steps, and the effective sampling dimensions hinge on its eigenvectors and eigenvalues. To enhance the simulation efficiency and capability of constraints, we employ the BO method to select the promising candidates from subspace samples for simulation. The GP models are trained near the current center using a fixed number of samples to reduce the computation redundency. To update the covariance matrix, we use the last center step $\boldsymbol{s}_{k+1}$ and two approximate gradients. We approximate the current gradient $\overline{\boldsymbol{g}}_{k+1}$ on GP models and the delayed gradient $\tilde{\boldsymbol{g}}_k$ by gradient sketching without additional simulations. Detailed implementation is listed in Alg.1 and illustrated in Fig.1.

A. Truncated subspace sampling

Ideally, we establish the subspace through a covariance matrix $\boldsymbol{C}_k$ with gradients of the Lagrangian $\boldsymbol{g}_i$ and center steps $\boldsymbol{s}_i$ for $i = k...1$. To avoid magnitude difference, the gradients and center steps used here are all normalized. In general, the subspace can be written as follows,

$$\boldsymbol{C}_k = \frac{1}{2k}\sum_{i=1}^{k} \beta_i(\boldsymbol{g}_i \cdot \boldsymbol{g}_i^T + \boldsymbol{s}_i \cdot \boldsymbol{s}_i^T) = \boldsymbol{W}_k \boldsymbol{\Lambda}_k \boldsymbol{W}_k^T \tag{9}$$

where $\boldsymbol{W}_k$ and $\boldsymbol{\Lambda}_k$ give the orthogonal column eigenvectors and corresponding diagonal eigenvalues, and β_i is the weights. Instead of using the same weights or a fixed subspace dimensionality, we adaptively change the effective subspace

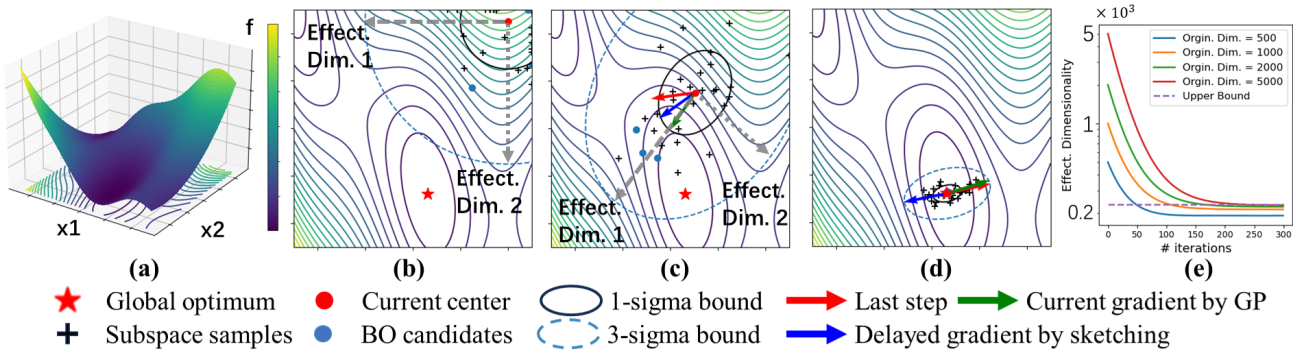

Fig. 1: Illustration of the proposed tSS-BO method. **(a)** The landscape of objective function. **(b)** The covariance matrix is initialized as an identity matrix. The BO candidates are selected from random subspace samples. **(c)** The covariance is updated with new features as (10). **(d)** The method converges to the global optimum. **(e)** Illustration of the effective dimensionality reduction in problems with different dimensionalities with β=0.95.

Algorithm 1: tSS-BO: scalable Bayesian optimization with truncated subspace sampling

Input: objective f; constraints $c^{(1\ldots N_c)}$; batch size N_s; initial dataset D_0; max simulations N_{max}; local GP training number N_{train}; decaying factor β.
Output: optimal design $\boldsymbol{x}^*$.

Initialize $\boldsymbol{C}_1 = \boldsymbol{I}, \sigma_1 = 0.3, k \leftarrow 1, D \leftarrow D_0$, the first center $\boldsymbol{x}_1$ and prior gradient $\tilde{\boldsymbol{g}}_0$ as (22) and (23);
while $\#simulations < N_{max}$ **do**
  Draw truncated subspace samples $\boldsymbol{X}_k^{(s)}$ as (12);
  Train local GP models near $\boldsymbol{x}_k$ with N_{train} points with the smallest normalized distances as (14);
  Select BO candidates $\boldsymbol{X}_k^{(BO)}$ from $\boldsymbol{X}_k^{(s)}$ with largest N_s wEI values calculated as (5);
  Invoke parallel simulations and update dataset $D \leftarrow D + \{(\boldsymbol{x}_i, f_i, c_i^{(1\ldots N_c)}) | \boldsymbol{x}_i \in \boldsymbol{X}_k^{(BO)} + \{\boldsymbol{x}_k\}\}$;
  Update $\boldsymbol{x}_{k+1}$ and σ_{k+1} as (15) and (17);
  Approximate delayed gradient $\tilde{\boldsymbol{g}}_k$ using gradient sketching as (21);
  Approximate current gradient $\overline{\boldsymbol{g}}_{k+1}$ on GP models;
  Update the last step $\boldsymbol{s}_{k+1} = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$;
  Update covariance matrix $\boldsymbol{C}_{k+1}$ as (10), $k \leftarrow k+1$;
end
Return optimal design $\boldsymbol{x}^*$ from the simulation history.

dimensions through eigenvectors and eigenvalues. In practice, we compute the covariance matrix in a truncated flow,

$$\boldsymbol{C}_{k+1} = \beta \boldsymbol{C}_k + \frac{1-\beta}{3}\chi(d)(\tilde{\boldsymbol{g}}_k \cdot \tilde{\boldsymbol{g}}_k^T + \overline{\boldsymbol{g}}_{k+1} \cdot \overline{\boldsymbol{g}}_{k+1}^T + \boldsymbol{s}_{k+1} \cdot \boldsymbol{s}_{k+1}^T), \tag{10}$$

where $\tilde{\boldsymbol{g}}_k$ denotes the delayed approximate gradient at the previous center $\boldsymbol{x}_k$ as detailed in Section III-C, $\overline{\boldsymbol{g}}_{k+1}$ denotes the expectation of gradient at the current center $\boldsymbol{x}_k$ on GP models, and $\boldsymbol{s}_{k+1} = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$ denotes the last center step. $\beta < 1$ is the decaying factor. We can choose the value $\beta = \exp(\ln(0.01)/M)$ to disregard subspace features preceding the last M iterations. The intial covariance matrix is set as the identity matrix. $\chi(d)$ is the expectation of Euclidean norm of multivariate normal distribution as follows,

$$\begin{aligned}\chi(d) &= E\|\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})\| = \sqrt{2}\Gamma(\frac{d+1}{2})/\Gamma(\frac{d}{2}) \\ &\approx \sqrt{d}(1 - \frac{1}{4d} + \frac{1}{21d^2}),\end{aligned} \tag{11}$$

where $\Gamma(\cdot)$ is the gamma functions. We draw N_s subspace samples $\boldsymbol{X}_k^{(s)}$ from the Gaussian distribution specified by the center $\boldsymbol{x}_k$, covariance $\sigma_k^2 \boldsymbol{C}_k$ as follows,

$$\begin{aligned}\boldsymbol{Z}_k &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \\ \boldsymbol{Q}_k &= \sigma_k \boldsymbol{W}_k \boldsymbol{\Lambda}_k^{1/2} \boldsymbol{W}_k^T \boldsymbol{Z}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma_k^2 \boldsymbol{C}_k), \\ \boldsymbol{X}_k^{(s)} &= \boldsymbol{x}_k + \boldsymbol{Q}_k \sim \mathcal{N}(\boldsymbol{x}_k, \sigma_k^2 \boldsymbol{C}_k).\end{aligned} \tag{12}$$

As illustrated in Fig.1(b), we firstly initialize the covariance matrix as an identity matrix. The dimensions are all equivalently significant and the effective dimensionality is same as that of original problem. Then the covariance is updated with new features from the last center step and two approximate gradients as (10) as shown in Fig.1(c). The varying significance of two effective dimensions, associated with covariance matrix eigenvalues, is evident. Sampling points are mainly oriented with the prior eigenvector of the covariance matrix.

The presence of small eigenvalues reduces the affect of corresponding eigenvectors, resulting in the reduction of effective dimensionality. However, the small eigenvalues are still numerically positive, which means that the subspace samples also exhibit a distribution along minor eigenvectors, retaining the high-dimensional information. The effective dimensionality has a constant upper bound regardless of the original dimensionality after amounts of iterations. A simple estimate of effective dimensionality is $d_k^{(eff)} = \|\boldsymbol{\Lambda}_k^{1/2}\|_1^2 / \|\boldsymbol{\Lambda}_k^{1/2}\|^2$ and the corresponding upper bound is derived as follows and shown in Fig.1(e),

$$\lim_{k \to +\infty} \sup\{d_k^{(eff)}\} = 3 \times \frac{1+\sqrt{\beta}}{1-\sqrt{\beta}}. \tag{13}$$

B. Selection and update

The subspace samples from (12) subjects to the truncated Gaussian distribution, we use BO method to select the most promising candidates to improve the simulation efficiency and to handle the constraints. To reduce $O(N^3)$ training complexity of GP model with respect to the training samples number N,

we train local GP models trained using N_{train} samples with the smallest normalized distance from $\boldsymbol{x}_k$ calculated by (14).

$$l_k(\boldsymbol{x}) = \|\boldsymbol{C}_k^{-1/2}\boldsymbol{r}_k\| = \|\boldsymbol{W}_k\boldsymbol{\Lambda}_k^{-1/2}\boldsymbol{W}_k^T\boldsymbol{r}_k\|, \tag{14}$$

where $\boldsymbol{r}_k = \boldsymbol{x} - \boldsymbol{x}_k$ is the difference vector.

We evaluate the wEI value of $\boldsymbol{X}_k^{(s)}$ as (5) and choose N_s samples with the largest wEI values as BO candidates $\boldsymbol{X}_k^{(BO)}$. The selected BO candidates, along with the current center $\boldsymbol{x}_k$, are then simulated. We update the center point using the best half part of the simulation points,

$$\boldsymbol{x}_{k+1} = \sum_{i=1}^{\lceil N_s/2 \rceil} \mu_i \boldsymbol{x}_i, \tag{15}$$

where μ_i is the normalized weight with $\sum_{i=1}^{\lceil N_s/2 \rceil} \mu_i = 1$, $\boldsymbol{x}_i \in \boldsymbol{X}_k^{(BO)}$ is sorted by the Lagrangian of simulation data L_i from small to large. A trivial choice can be $\mu_i = 1/\lceil N_s/2 \rceil$ for all i. For circuit cases, we set the weights as follows,

$$\mu_i = \frac{\exp\left(\frac{L_k - L_i}{\Delta}\right)}{\sum_{i=1}^{\lceil N_s/2 \rceil} \exp\left(\frac{L_k - L_i}{\Delta}\right)}, \tag{16}$$

where L_k is the Lagrangian of center $\boldsymbol{x}_k$, $\Delta = |L_k - L_1|$ is the absolute value of the biggest improvement. The deviation σ is updated based on the normalized distance of $\boldsymbol{x}_{k+1}$ and $\boldsymbol{x}_k$ as follows,

$$\sigma_{k+1} = \sigma_k \cdot \exp\left(\frac{l_k(\boldsymbol{x}_{k+1})}{\chi(d)\sqrt{\sum \mu_i^2}} - 1\right). \tag{17}$$

C. Gradient sketching

Inspired by a quasi-gradient method for federated learning [17], we use gradient sketching to approximate the delayed gradients with the simulation data of BO candidates. The directional derivatives, or called conditional gradients, along a unit vector $\boldsymbol{a}$ and a small step h can be written as follows,

$$f(\boldsymbol{x} + h\boldsymbol{a}) - f(\boldsymbol{x}) = h\boldsymbol{a}^T \cdot \nabla f(\boldsymbol{x}), \tag{18}$$

where $\nabla f(\boldsymbol{x})$ is the gradient at $\boldsymbol{x}$. When there are a batch of sketching points, the equation set can be written as follows,

$$f(\boldsymbol{x} + h\boldsymbol{A}) - f(\boldsymbol{x}) = h\boldsymbol{A}^T \cdot \nabla f(\boldsymbol{x}). \tag{19}$$

In practice, (19) is usually under-determined because we have less than d simulation points in each iteration. By introducing an alternative step in the place of the small step h, we calculate the approximate gradient $\tilde{\boldsymbol{g}}_k$ using the simulation data of BO candidates $\boldsymbol{X}_k^{(BO)}$ as follows,

$$\begin{aligned} \tilde{\boldsymbol{g}}_k := \underset{\boldsymbol{g}}{\operatorname{argmin}} \ & |\boldsymbol{g} - \tilde{\boldsymbol{g}}_{k-1}|, \\ \text{s.t. } & \boldsymbol{\Psi}_k^T \cdot \boldsymbol{g} = \Delta \boldsymbol{L}_k, \end{aligned} \tag{20}$$

where $\boldsymbol{\Psi}_k = \boldsymbol{X}_k^{(BO)} - \boldsymbol{x}_k$ is the difference matrix, $\Delta \boldsymbol{L}_k = L(\boldsymbol{X}_k^{(BO)}) - L(\boldsymbol{x}_k)$ is the Lagrangian difference vector and $\tilde{\boldsymbol{g}}_{k-1}$ denotes the approximate gradient at $\boldsymbol{x}_{k-1}$ serving as the prior. The explicit solution of (20) can be derived as,

$$\tilde{\boldsymbol{g}}_k = \tilde{\boldsymbol{g}}_{k-1} + \boldsymbol{\Psi}_k(\boldsymbol{\Psi}_k^T\boldsymbol{\Psi}_k)^{-1}(\Delta \boldsymbol{L}_k - \boldsymbol{\Psi}_k^T \cdot \tilde{\boldsymbol{g}}_{k-1}). \tag{21}$$

Notably, the gradient sketching is calculated using simulation data of BO candidates around the previous center, hence it is referred as delayed gradients. Consequently, we highlight the effect of gradients in (10) to promote convergence and mitigate systematic errors with two distinct approximation approaches.

D. Initial setup

For the first iteration, we set a random initial point as the first center for Alg.1 if no initial points given. Otherwise, we start from the center point using the best quarter part of initial dataset D_0 as follows.

$$\boldsymbol{x}_1 = \begin{cases} RNG(d) \in S, & |D_0| = 0, \\ \sum_{i=1}^{\lceil |D_0|/4 \rceil} \mu_i \boldsymbol{x}_i^D, & |D_0| > 0, \end{cases} \tag{22}$$

where $\boldsymbol{x}_i^D$ is the initial samples sorted by simualtion data and the weight μ_i is calculated as (16). Then we will invoke simulation at $\boldsymbol{x}_1$ and compute the prior gradient as follows.

$$\tilde{\boldsymbol{g}}_0 = \begin{cases} \boldsymbol{0}, & |D_0| = 0, \\ \boldsymbol{\Psi}_0(\boldsymbol{\Psi}_0^T\boldsymbol{\Psi}_0)^{-1}\Delta \boldsymbol{L}_0, & |D_0| > 0. \end{cases} \tag{23}$$

IV. Experiments

We implement tSS-BO[1] with PyTorch. The performances are tested on synthetic benchmark functions and real-world analog circuits. Since conventional analog circuit optimization methods always yield failure in high-dimensional problems, the tSS-BO method is compared to state-of-the-art high-dimensional methods TuRBO[2] [10] and CMA-ES[3] [13]. TuRBO with n trust regions is denoted as TuRBO-n. To ensure the robustness, we run experiments 10 times. All the methods are run with batch/population size of 20 and simulations are invoked in parallel. All the aspects of the algorithms, i.e. initialization, optimization and evaluation/simulation, are counted in the total runtime. The experiments are conducted on a Linux server with Intel Xeon 6230 CPUs.

A. Synthetic benchmark functions

We use two multi-modal Ackley, Griewank functions and one valley-shaped Rosenbrock function with 100 and 200 independent variables to test the algorithm performances.

We also test methods labeled as tSS only and CMA-ES+BO. The tSS only uses the truncated subspace sampling technique and gradient sketching, without the guidance of BO selection and GP model gradient. CMA-ES+BO is a trivial combination of the CMA-ES and BO selection as in Section III-B. The average results of 10 independent runs are plotted in Fig.2.

The proposed tSS-BO method achieves the best solutions in all the experiments. In terms of evaluation efficiency, the tSS-BO method achieves $1.68 \sim 2.20\times$ speedups with respect to CMA-ES and $3.40 \sim 4.93\times$ speedups with respect to TuRBO.

In terms of runtime, for all cases with 3000 evaluation budget, the tSS-BO method takes total 159s in average while other average runtime is 4867s, 5578s and 24s for TuRBO-1, TuRBO-5 and CMA-ES respectively, which is over $30.6\times$ algorithm comlexity reduction compared to the TuRBO method. Notably, for tSS only method, it takes 1.9s in average.

The BO selection technique can improve the evaluation efficiency incrementally and it shows more advantages in problems with higher dimensions. Without BO selection, the truncated subspace sampling method still outperforms other algorithms in solution quality, evaluation efficiency and runtime.

[1] https://github.com/PascalInn/tSS-BO
[2] https://github.com/uber-research/TuRBO
[3] https://github.com/CyberAgentAILab/cmaes

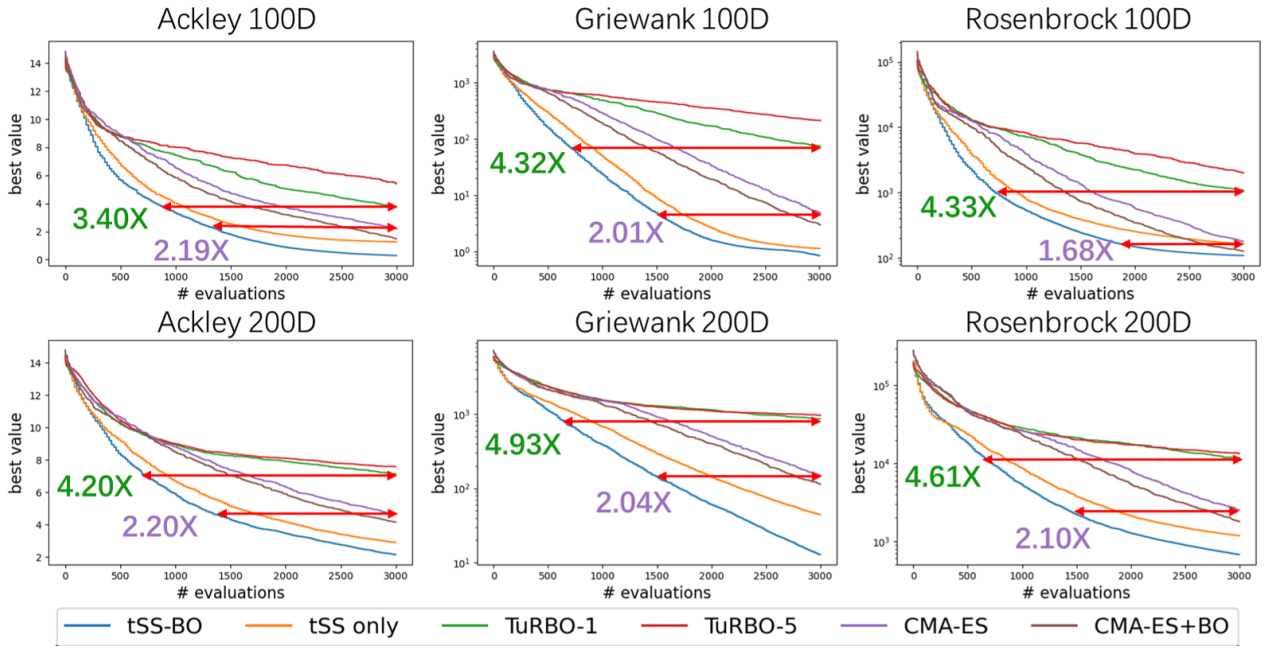

Fig. 2: The convergence curves of algorithms on the Ackley, Griewank and Rosenbrock functions. The dimensionalities for each experiment are shown in the sub-figure titles. The nominal optimization range for these functions are $[-5, 10]^d$, $[-600, 600]^d$ and $[-2.5, 2.5]^d$ respectively. All the algorithms are initialized randomly with one sample. The initial σ and covariance for CMA-ES are 0.3 and identity matrix respectively. The setup of TuRBO is same as in [10].

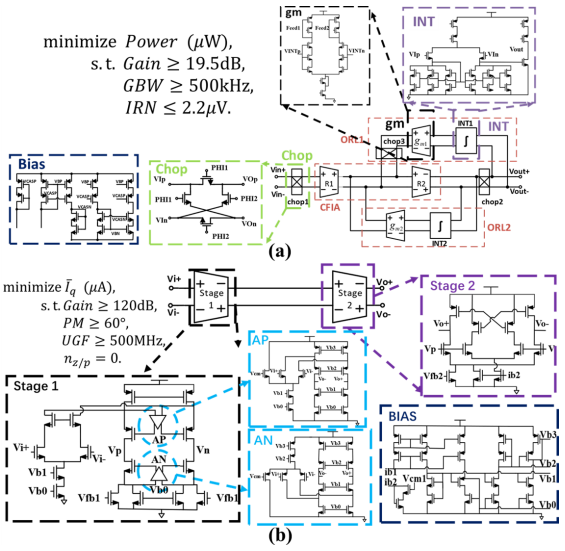

Fig. 3: The schematics and optimization formulations of **(a)** the ACCIA circuit and **(b)** the OTA circuit.

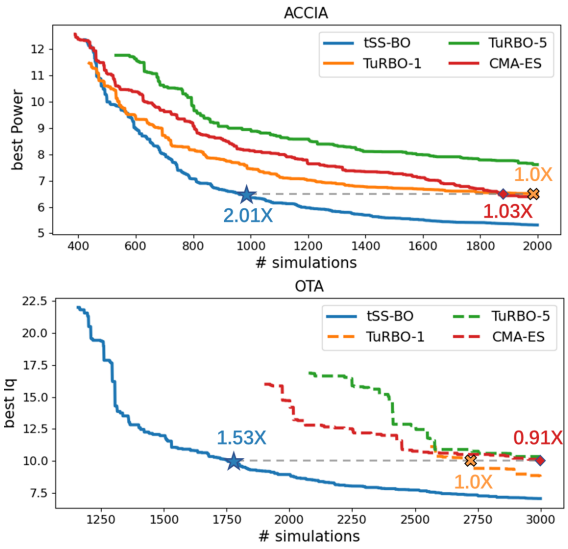

Fig. 4: The convergence curves of optimization results of two real-world circuit ACCIA and OTA.

B. Real-world analog circuits

We test the proposed method with two real-world analog circuits. We feed the same random initial points for all algorithms. The optimized variables include the size of the transistors, values of resistors and capacitors. The symmetry parameters (e.g., of differential pairs and current mirrors) are considered and represented by the same optimized variables. The results are shown in Fig.4 and Table I. We only consider feasible designs that meet the constraints and hence convergence curves may start from different points in Fig.4. The FOM (figure of merit) column in Table I gives the mean and standard deviation values of objective circuit performances of feasible designs. The speedup data in Table I are calculated comparing to TuRBO-1. We show the results of algorithms with failure runs using average performances of their success runs. These results are shown with dashed curves in Fig.4 and with star superscript markers in Table I.

1) ACCIA: The AC-Coupled Instrumentation Amplifier (ACCIA) consists with current feedback instrumentation amplifier (CFIA), transconductor and offset reduction loops (ORL) [18] as shown in Fig.3(a). A single simulation of ACCIA takes about 17.5 minutes and 68 independent variables are optimized with 50 initial points and total 2000 simulations. Power consumption is the objective performance and it must

TABLE I: Optimization Results of Real-World Analog Circuits

Circuit	Algo.	FOM	# simu	# simu speedup	runtime	runtime speedup	# success
ACCIA	proposed	$\mathbf{6.35 \pm 0.98}$	**997**	**2.01×**	**14.69h**	**2.00×**	**10/10**
	TuRBO-1	6.51 ± 0.76	2000	1×	29.41h	1×	10/10
	TuRBO-5	7.61 ± 0.83	2000	1.00×	29.37h	1.00×	10/10
	CMA-ES	6.43 ± 0.26	1933	1.03×	28.42h	1.03×	10/10
OTA	proposed	$\mathbf{9.91 \pm 1.23}$	**1776**	**1.53×**	**1.35h**	**1.98×**	**10/10**
	TuRBO-1*	9.94 ± 2.53	2715	1×	2.68h	1×	5/10
	TuRBO-5*	10.57 ± 1.92	3000	0.91×	2.59h	1.03×	7/10
	CMA-ES*	10.15 ± 1.36	3000	0.91×	2.08h	1.28×	8/10

meet constraints of gain, GBW, and input referred noise (IRN, the root mean square of noise voltage at $0.5 \sim 100$Hz). Achieving a similar FOM value, the proposed tSS-BO achieves a $2.01\times$ simulation number speedup and a $2.00\times$ runtime speedup compared to TuRBO-1. The aforementioned speedups are $1.94\times$ and $1.93\times$ compared to CMA-ES respectively.

2) OTA: The operational trans-impedance amplifier circuit with cascode compensation (OTA) was originally implemented in a pipeline ADC [19]. The schematic of the OTA circuit and its main specifications are shown in Fig.3(b). A single simulation of OTA takes about 50 seconds and 103 independent variables are optimized with 100 initial points and total 3000 simulations. $\bar{I}_q$ is the mean power of circuit with small-swing and large-swing. $n_{z/p}$ is the number of zero/pole within the range of UGF. Other constraints guarantee the large signal behavior and operating points. The TuRBO-1 method fails to find the feasible design in 5 out of 10 runs. TuRBO-5 and CMA-ES also fail in 3 and 2 out of 10 runs respectively. Comparing to success runs of other algorithms, the proposed tSS-BO still achieves an $1.53\times$ simulation number speedup and an $1.98\times$ runtime speedup compared to TuRBO-1. The aforementioned speedups are $1.68\times$ and $1.54\times$ compared to CMA-ES respectively.

V. CONCLUSION

A novel scalable Bayesian optimization method with truncated subspace sampling (tSS-BO) is presented. The proposed approach can adaptively reduce the effective sampling dimensionality down to a constant upper bound with truncated subspace sampling technique, incoporating gradients, center update steps and prior subspace features. The BO candidates are selected from subspace samples to enhance the efficiency. To save simulation budget and to reduce approximate errors, gradients are approximated through gradient sketching and local GP models. Experiments have demonstrated advantages of the tSS-BO method. Specifically, on synthetic benchmark functions, the tSS-BO method exhibits $1.68 \sim 2.20\times$ speedups with respect to CMA-ES and $3.40 \sim 4.93\times$ speedups with respect to TuRBO in terms of evaluation efficiency, and achieves a remarkable over $30.6\times$ algorithm complexity reduction compared to TuRBO. On real-world analog circuits, we achieve $1.53 \sim 2.01\times$ simulation number speedups and $1.98 \sim 2.00\times$ runtime speedups with respect to TuRBO.

ACKNOWLEDGEMENTS

This research is supported partially by National Key R&D Program of China 2020YFA0711900, 2020YFA0711901, partly by National Natural Science Foundation of China (NSFC) research projects 62141407, 62304052, 61974032, 61929102, 61674042, 62090025, and partly by Beijing Natural Science Foundation (Grant NO. Z220003).

REFERENCES

[1] R. A. Rutenbar, G. G. E. Gielen, and J. Roychowdhury, "Hierarchical modeling, optimization, and synthesis for system-level analog and rf designs," *Proceedings of the IEEE*, vol. 95, no. 3, pp. p.640–669, 2007.

[2] Y. Li, Y. Wang, Y. Li, R. Zhou, and Z. Lin, "An artificial neural network assisted optimization system for analog design space exploration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2640–2653, 2020.

[3] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, and F. V. Fernández, "Analog circuit optimization system based on hybrid evolutionary algorithms," *Integration*, vol. 42, no. 2, pp. 137–148, 2009.

[4] B. Liu, D. Zhao, P. Reynaert, and G. G. E. Gielen, "Gaspad: A general and efficient mm-wave integrated circuit synthesis method based on surrogate model assisted evolutionary algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 2, pp. 169–182, 2014.

[5] W. Lyu, P. Xue, F. Yang, C. Yan, Z. Hong, X. Zeng, and D. Zhou, "An efficient bayesian optimization approach for automated optimization of analog circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 6, pp. 1954–1967, 2017.

[6] M. Liu, W. J. Turner, G. F. Kokai, B. Khailany, D. Z. Pan, and H. Ren, "Parasitic-aware analog circuit sizing with graph neural networks and bayesian optimization," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021, pp. 1372–1377.

[7] A. F. Budak, M. Gandara, W. Shi, D. Z. Pan, N. Sun, and B. Liu, "An efficient analog circuit sizing method based on machine learning assisted global optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 5, pp. 1209–1221, 2022.

[8] H. Wang, K. Wang, J. Yang, L. Shen, N. Sun, H.-S. Lee, and S. Han, "Gcn-rl circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2020, pp. 1–6.

[9] 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 Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 7, pp. 2080–2093, 2023.

[10] D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek, "Scalable global optimization via local bayesian optimization," *Advances in neural information processing systems*, vol. 32, 2019.

[11] K. Touloupas, N. Chouridis, and P. P. Sotiriadis, "Local bayesian optimization for analog circuit sizing," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 1237–1242.

[12] K. Touloupas and P. P. Sotiriadis, "Locomobo: A local constrained multiobjective bayesian optimization for analog circuit sizing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 9, pp. 2780–2793, 2022.

[13] N. Hansen, *The CMA Evolution Strategy. A Comparing Review*. Berlin: Springer, 2006, vol. 192, pp. 75 – 102.

[14] Y.-x. Yuan and J. Stoer, "A subspace study on conjugate gradient algorithms," *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 75, no. 1, pp. 69–77, 1995.

[15] 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.

[16] Y.-x. Yuan, "A review on subspace methods for nonlinear optimization," in *International Congress of Mathematicians*, vol. 4, 2014, pp. 807–827.

[17] R. M. Gower, P. Richtárik, and F. Bach, "Stochastic quasi-gradient methods: Variance reduction via jacobian sketching," *Mathematical Programming*, vol. 188, pp. 135–192, 2021.

[18] X. Pu, Z. Hui, Y. Qin, and Z. Hong, "An 8-channel readout front-end for long-term sleep quality monitoring," in *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2011.

[19] G. Shu, M. Fan, C. Shu, C. Chen, N. Li, and J. Ren, "A 12-bit 50-ms/s pipelined analog-to-digital converter in 65nm cmos," in *2010 10th IEEE International Conference on Solid-State and Integrated Circuit Technology*, 2010, pp. 563–565.