PVTSizing: A TuRBO-RL-Based Batch-Sampling Optimization Framework for PVT-Robust Analog Circuit Synthesis

Zichen Kong[1], Xiyuan Tang[2,1,3*], Wei Shi[4], Yiheng Du[5], Yibo Lin[1,7], Yuan Wang[1,3,6,7*]
[1]School of Integrated Circuits, Peking University
[2]Institute for Artificial Intelligence, Peking University
[3]Key Laboratory of Microelectronic Devices and Circuits (MoE), MPW Center, Peking University
[4]Department of Electrical and Computer Engineering, The University of Texas at Austin
[5]Yuanpei College, Peking University
[6]Beijing Laboratory of Future Integrated Circuit Technology and Science, Peking University
[7]Beijing Advanced Innovation Center for Integrated Circuits

***Abstract*—With the CMOS technology advancing and the complexity of circuits growing, the demand for analog/mixed-signal design automation tools is increasing quickly. Although some tools have been developed to tackle this challenge, the performance degradation caused by process, voltage, and temperature (PVT) variations has been less considered. This paper presents PVTSizing, an optimization framework for PVT-robust analog circuit synthesis. PVTSizing adopts trust region Bayesian optimization (TuRBO) for high-quality initial datasets and reference points. Multi-task reinforcement learning (RL) is utilized for PVT optimization. Both TuRBO and RL are batch-friendly, allowing parallel sampling of design solutions. Meanwhile, critic-assisted pruning and zoom target metrics are proposed to improve sample efficiency and reduce runtime. In addition, this framework naturally supports sizing over random mismatch. On 4 real-world circuits with TSMC 28/180nm process, PVTSizing achieves $1.9\times-8.8\times$ sample efficiency and $1.6\times-9.8\times$ time efficiency improvements compared to prior sizing tools from both industry and academia.**

***Index Terms*—Bayesian optimization, Reinforcement learning, PVT variation, Analog circuit synthesis**

I. Introduction

As modern System-on-Chip (SoC) architectures increase in complexity, the imperative for advanced design automation tools intensifies. Nowadays, commercial digital circuit design automation tools have already been widely used, greatly improving the design efficiency of digital processing cores. However, modern chips still incorporate various analog signal conditioning components, such as amplifiers, references, and clocks. Due to their performance sensitivity, analog circuits heavily rely on manual design, severely limiting design efficiency and elongating time-to-market. As a result, the industry is witnessing an emergent need for sophisticated automated analog circuit design tools.

Note that analog circuit suffers from a unique challenge that is raised by process, voltage, and temperature (PVT) variations. The process variation occurs during manufacturing; the voltage variation usually comes from the non-ideal external power supply; and the temperature variation is caused by the working condition change. The analog performance is usually highly sensitive to PVT variations. For instance, the gain of an amplifier may easily vary $\pm$20dB across corners [1]. Thus, it is critical to maintain analog circuits' performance across all PVT corners to ensure chip functionality in real-world scenarios.

In recent years, machine learning (ML) methods and black box optimization algorithms like Bayesian optimization (BO) have become popular in the analog design automation field. [2], [3] implement deep neural network (DNN) models for optimization. [4]–[6] utilizes RL agent to search for solutions. [7]–[10] treat circuit sizing as black-box optimization problems and utilize BO to search design space. However, most prior works focus on the typical condition optimization [2], [3], [5], [7]–[11]. Although [7], [8], [10] and [11] take PVT variation into account in some testcases, no pruning for PVT is conducted. They simply sample every design solution under full PVT conditions without efficient PVT exploration strategies, which brings lots of unnecessary circuit simulations, severely limiting the design efficiency. For the existing PVT-aware circuit sizing tools embedded with efficient PVT exploration strategies, [4] runs PVT exploration by continuously overcoming the dominant PVT corners, but treats each PVT condition as an independent model, introducing considerable storage and computing overhead. [6] implements multi-task RL for PVT exploration, but suffers from low-quality random initial sampling, which significantly reduces the sample efficiency and success rate of optimization. Additionally, random mismatches of analog components can result in significant performance degradation, such as large DC offset, even-order distortions, and low common mode rejection ratio (CMRR). Similar to PVT variations, this challenge has not been well addressed in prior designs.

This paper proposes PVTSizing, a batch-sampling optimization framework for PVT-robust analog circuit synthesis. Experimental results demonstrate that the proposed framework achieves much higher efficiency in PVT-robust analog IC sizing problems compared to the state-of-the-art tools both from the industry and academia: *Sizing Over Corners* embedded in commercial tool Virtuoso [12] and RobustAnalog [6]. The proposed analog synthesis flow can be used for mismatch exploration directly by regarding each set of mismatch parameters as a circuit corner.

The core contributions of our work are listed below:

- We propose PVTSizing, a batch-sampling PVT-robust analog IC design automation tool with TuRBO for initial sampling and multi-task RL agents for PVT-phase optimization.
- Critic-assisted pruning, a PVT-aware pruning method, is proposed to increase convergence speed by predicting the impact brought by sampling. This pruning method uses the information provided by the Critic network, thus improving sample efficiency and reducing runtime.
- Zoom target metrics are proposed to accelerate the convergence of multi-task RL agents.
- Sizing over random mismatch is also supported to suit practical needs better.
- The proposed PVTSizing achieves $1.9\times-8.8\times$ sample efficiency and $1.6\times-9.8\times$ time efficiency improvements compared to state-of-the-art tools from both industry and academia.

The rest of the paper is organized as follows. Section II formulates the analog sizing problem and introduces the background of trust-region Bayesian optimization. Section III presents the details of the proposed PVTSizing. Section IV presents and discusses experimental results. Section V concludes the paper.

*Corresponding authors: {xitang, wangyuan}@pku.edu.cn

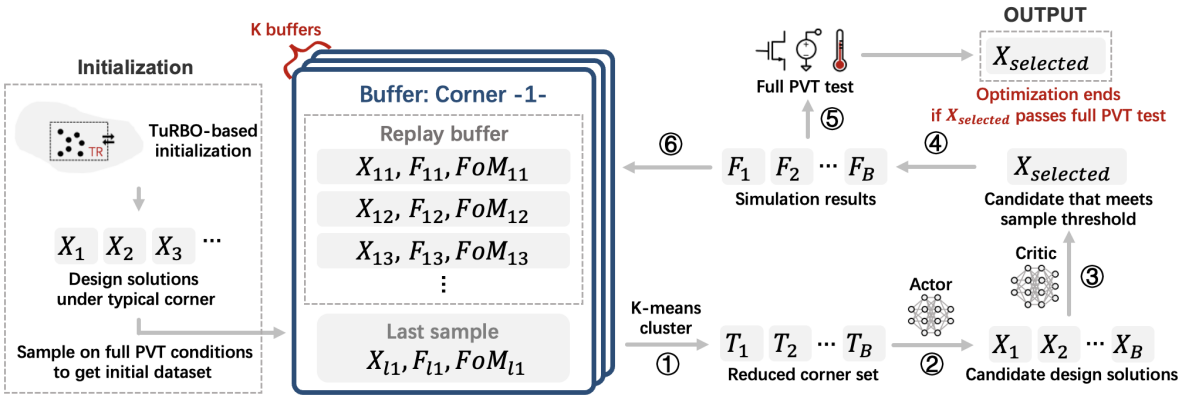

Fig. 1. The workflow of the proposed PVTSizing.

II. Preliminaries

A. Trust Region Bayesian Optimization

In the high-dimensional black-box optimization problem, it is difficult for a global surrogate model to fit the whole search space. TuRBO provides the scalable global optimization via local Bayesian optimization [13]. It selects several hypercubes within the search space as trust regions (TR). TR has the same dimension as the search space, and is centered at the best solution point within this TR. Each TR contains a local surrogate model for that local area. Assume the black box optimization problem to solve is

$$Find\ x^* \in D\ such\ that\ f(x^*) \leq f(x), \forall x \in D,$$

where $f : D \to \mathbb{R}$ and $D = [0, 1]^n$. Define the center of TR as x_c. In each iteration, a batch of candidates is selected within each TR. If x_c can continuously be improved in several consecutive iterations, one can conclude that the global minimum is outside of this TR. So, the side length of this TR will expand. Conversely, if x_c presents no improvement for past several consecutive iterations, x_c is likely to be the local or global minimum. In this case, the side length of this TR will shrink. Once a TR is smaller than a given threshold, it will be replaced by another new TR found at a random location. As a result, the sample point that meets the target value can be quickly discovered, even in high-dimensional optimization problems.

B. Problem Formulation

The analog circuit operates normally only when all performance metrics meet the target. So, the analog sizing problem can be formulated as a constraint satisfaction problem.

$$\begin{aligned} &minimize && 0 \\ &subject\ to && F_i(X|T_j) < C_i,\ j = 1, 2, ..., k \end{aligned} \tag{1}$$

where

$$\begin{aligned} &X \in D \\ &X = X_1, X_2, ..., X_n \\ &D = D_1, D_2, ..., D_n \\ &T = T_1, T_2, ..., T_k \\ &C = C_1, C_2, ..., C_m \end{aligned}$$

X is the n-dimension design solution of the sizing problem corresponding to n device sizing variables. D is the n-dimension design space. C is the m-dimension constraint vector representing m performance metrics. T is PVT corner, and there are k PVT conditions in total. F is the mapping from the circuit design solution to performance metrics. $F_i(X|T_j)$ means the i^{th} performance metrics when we simulate this circuit under design solution X and PVT condition T_j.

To simplify this problem, a figure of merit (FoM) is defined to convert multiple objectives into a single target. The definition of FoM is as follows:

$$FoM = \begin{cases} r, & r < 0 \\ 0.2, & r \geq 0 \end{cases} \tag{2}$$

$$r = \begin{cases} \sum_{i=1}^{m} \min\left(\frac{F_i - C_i}{\max(F_i + C_i, C_i)}, 0\right), & \text{if we want } F_i \geq C_i \\ \sum_{i=1}^{m} \min\left(\frac{C_i - F_i}{\max(F_i + C_i, C_i)}, 0\right), & \text{if we want } F_i \leq C_i \end{cases}$$

Performance metrics are normalized during the calculation of FoM. When all the constraints are satisfied, FoM is set to 0.2. This FoM formulation is modified from [6]. The max function is added to the denominator to ensure that the denominator is always positive, which guarantees that metrics only affect the sign of FoM by changing the sign of the numerator. For simplicity, we will use FoM of a design solution X' under PVT condition T' to represent $FoM(F(X'|T'))$.

III. Proposed PVTSizing

A. Framework Overview

Fig. 1 shows the overview of PVTSizing. First, it runs TuRBO for initial sampling, generating design solutions meeting constraints under the typical condition. Then, sample these design solutions under full PVT conditions to generate the initial dataset. The dataset will be updated to replay buffers. There are k corner buffers in total; each corresponds to a PVT corner. Every buffer consists of a replay buffer for saving simulation data and a last-sample buffer for recording the last-sample data under the corresponding corner. Each iteration can be divided into 6 steps.

Step 1: Prune PVT corners to get the reduced corner set. PVT corners are grouped according to their performance metrics stored in last-sample buffers. K-means algorithm is adopted to divide performance metrics into B categories [14]. Each category will add the corner with the worst last-sample FoM to the reduced corner set.

Step 2: Get candidate design solutions. Put reduced corner set into Actor network to get B candidate design solutions.

Step 3: Get design solutions for sampling via Critic-assisted pruning. If no solution is given, go to step 6. Critic-assisted pruning finds the candidate that potentially improves the overall optimization speed the most by predicting its impact.

Step 4: Sample the selected design solution under reduced corner conditions and update corresponding buffers. Here, the simulator will run in parallel to sample design solutions under different corners.

Step 5: Run full PVT condition test if needed. If all performance metrics stored in last-sample buffers satisfy the constraints, sample the

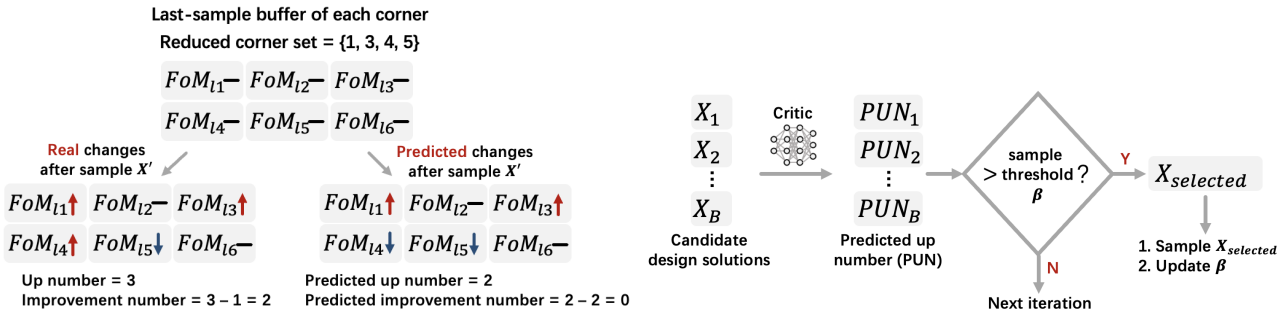

Fig. 2. Critic-assisted pruning: a) prediction by the Critic and b) the workflow overview.

latest selected design solution in step 3 under full PVT conditions. If this solution meets targets under all conditions, optimization ends. Otherwise, update the buffers once again.

Step 6: Update RL agent with data stored in buffers. Only data stored in buffers related to the reduced corner set will be selected for updating.

B. Multi-Task RL Agent

Actor-Critic method is widely used in RL agents [15]. In PVT-Sizing, we propose a multi-task RL agent to handle the optimization conflict among corners. The Actor includes a 5-layer neural network. The Actor's input is PVT states, a 7-dimensional vector, where the first five dimensions represent process variation by one-hot coding, and the last two dimensions are normalized voltage and temperature. The output of the Actor is an n-dimensional normalized design solution. The Critic is a 5-layer neural network. The input of Critic is also an n-dimensional normalized design solution, while the output of Critic is a k-dimensional vector representing the predicted FoM of input under each PVT condition.

The details of the agent updating method are given in Algorithm 1. Here, each task is defined as searching for a design solution that meets constraints under a certain PVT condition. PCGrad algorithm is implemented for multi-task agent updating to handle the gradient conflict among tasks [16]. In each iteration, only tasks that are related to the reduced corner set contribute to the final gradient for agent updating. The Actor and the Critic are updated alternately, but have different update frequencies according to N_A and N_Q.

Algorithm 1: Update Method of RL Agent

Given the i^{th} PVT corner of all PVT conditions T_i ;
Given the reduced corner set S_{Tr};
Given actor network $A(T|\theta^A)$ and critic network $Q(X|\theta^Q)$. Q_i refers to the i^{th} dimension of its output;
Given replay buffer B_r, and replay buffer for i^{th} corner B_r^i;
Given update times for Actor N_A, and update times for Critic N_Q;
for *update = 1,* M_{update} **do**
  if *update* $< M_{update}/2$ **then**
    Set $N_Q = 5, N_A = 1$;
  else
    Set $N_Q = 1, N_A = 2$;
  end
  for *updateQ = 1,* N_Q **do**
    For every $T_i \in S_{Tr}$, sample a batch of $(\widehat{X}^i, \widehat{FoM}^i)$ from $B_r^i \in B_r$;
    Update the Critic by minimizing MSE losses with PCGrad:
    $L_{Qi} = MSELoss(\widehat{FoM}^i, Q_i(\widehat{X}^i|\theta^Q) + bias)$;
  end
  for *updateA = 1,* N_A **do**
    Update the actor by minimizing MSE losses with PCGrad:
    $L_{Ai} = MSELoss(0.2, Q_i(A(T_i|\theta^A)|\theta^Q) + bias)$
  end
end

C. Critic-Assisted Pruning

In this subsection, we will first show the overview of proposed Critic-assisted pruning, and then present the details of formula derivation. The workflow of Critic-assisted pruning is shown in Fig. 2. Given that B candidate solutions are generated in each iteration, but only one solution will be sampled, a pruning strategy is needed. Meanwhile, there is a trade-off on whether to sample if the candidates in an iteration are not good enough. Based on the above demands, we propose Critic-assisted pruning, a PVT-aware method.

Note that the Critic is able to predict the FoM of candidate solutions. Experiments show that the Critic is good at predicting the relative quality of two solutions instead of absolute FoMs. The prediction accuracy for relative quality is 66% on average, which is sufficient for the pruning purpose.

Meanwhile, PVTSizing utilizes last-sample buffers to record the optimization progress of each corner. When a solution X is sampled under several PVT conditions, the optimization progress of corresponding corners will be changed. We define "up number" of a certain X as the number of corners being better after sampling X, "down number" of a certain X as the number of corners being worse, and "improvement number" of a certain X as "up number" minus "down number." In each iteration, the Critic will predict the "up number" of each candidate solution, and randomly select a candidate whose predicted "up number" is greater than a certain sample threshold β. Note that the sample threshold β is updated in each iteration.

To accelerate the convergence speed of PVT-phase optimization, a greater "improvement number" is preferred for every unit time. In other words, a higher "improvement speed" is preferred, which is given in equation (3).

$$G(\beta, P_{inc}, P_a) = Improvement\ speed = \frac{E(Improvement\ number)}{E(Sample\ interval)} \tag{3}$$

$$\beta = argmax(G(\beta, P_{inc}, P_a)) \tag{4}$$

Here, P_{inc} and P_a reflect the current quality of the Actor and the Critic. The typical values of P_{inc} and P_a are 0.61 and 0.66, respectively. The improvement speed is shown in Fig. 3. By sweeping

TABLE I
NOTATIONS IN CRITIC-ASSISTED PRUNING

Notation	Definition*	formula
f_r	The real FoM of X' under T'	$f_r = FoM(F(X'\|T'))$
f_p	The predicted FoM of X' under T'	$f_p = Critic(X')\|_{T=T'}$
f_l	The FoM of last sample under T'	/
P_{inc}	The probability of f_r being better than f_l	$P_{inc} = P(f_r > f_l)$
P_a	The Critic's accuracy in predicting FoM changes	$P_a = P(\{\{f_r > f_l\} \cap \{f_p > f_l\}\} \cup \{\{f_r < f_l\} \cap \{f_p < f_l\}\})$
P_H	The probability of f_p being better than f_l	$P_H = P(f_p > f_l)$
P_L	The probability of f_p being worse than f_l	$P_L = P(f_p < f_l)$
P_s	The probability of the predicted "up number" of $X' > \beta$	Given in equation (7)
I_H	Expected "improvement number" when f_p is better than f_l	$I_H = \frac{P(\{f_p>f_l\}\cap\{f_r>f_l\}) - P(\{f_p>f_l\}\cap\{f_r<f_l\})}{P(f_p>f_l)}$
I_L	Expected "improvement number" when f_p is worse than f_l	$I_L = \frac{P(\{f_p<f_l\}\cap\{f_r>f_l\}) - P(\{f_p<f_l\}\cap\{f_r<f_l\})}{P(f_p<f_l)}$
t_u	Time taken for Actor & Critic updating within an iteration	/
t_s	Time taken for sampling within an iteration	/

*Assume there is a design solution X' given by the Actor and a reduced PVT corner set S_{Tr}. Corner T' is randomly selected within S_{Tr}.

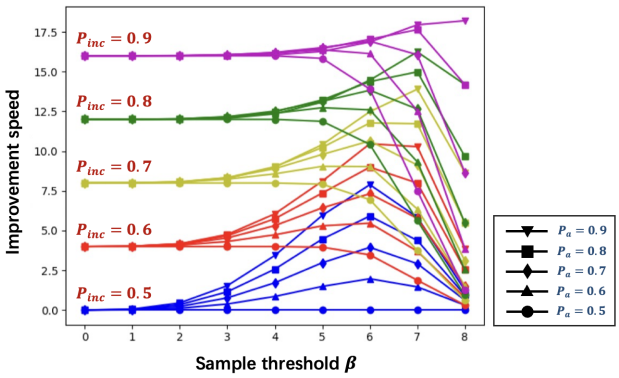

Fig. 3. Improvement speed under different P_a and P_{inc}

β from 0 to B, one can find the β with the highest improvement speed in each iteration.

Formula derivation. The notations used in formula derivation are defined in Tab. I. According to the definition, we have

$$E(Sample\ interval) = t_s + \frac{t_u}{1-(1-P_s)^B}, \tag{5}$$

$$E(Improvement\ number) = \sum_{i=\beta}^{B} \frac{\binom{B}{i} P_H^i P_L^{B-i}}{P_s} [iI_H + (B-i)I_L], \tag{6}$$

where

$$P_s = \sum_{i=\beta}^{B} \binom{B}{i} P_H^i P_L^{B-i}, \tag{7}$$

$$P_H = P_{inc}P_a + (1-P_{inc})(1-P_a), \tag{8}$$

$$P_L = P_{inc}(1-P_a) + (1-P_{inc})P_a, \tag{9}$$

$$I_H = \frac{P_{inc} + P_a - 1}{P_H}, \tag{10}$$

$$I_L = \frac{P_{inc} - P_a}{P_L}. \tag{11}$$

Combine equation (3)-(11) to get improvement speed, which is proved to be a function of sample threshold β. t_u, t_s, P_{inc}, and P_a are all easy to obtain. t_u and t_s are measured directly in each iteration. P_{inc} and P_a are estimated and soft updated after each sampling operation.

D. Zoom Target Metrics

The proposed zoom target metrics can increase optimization speed by setting two sets of target metrics. The tighter ones, which are usually 10% tighter than the targeted constraints, are used during the RL agents updating to accelerate convergence. The looser ones, which are the targeted constraints, are used in the 5^{th} step of iteration to decide whether to run a full PVT test and determine whether a solution passes this test.

By setting zoom target metrics, the agent is more likely to find and sample solutions with better metrics. Meanwhile, the tight set of constraints makes performance metrics less likely to oscillate around targeted constraints, which improves sample efficiency eventually.

E. Batch Sampling

Note that both TuRBO and PVT-phase optimization need batch sampling. TuRBO samples in batches to update the center point. PVT optimization samples in batches to simulate a design solution under several PVT conditions. In this case, the simulator can perform parallelly for time-saving.

F. Mismatch Exploration Strategy

Given the performance degradation caused by random mismatches, Monte Carlo (MC) sampling should be performed during the design phase to ensure its performance after fabrication. The proposed analog synthesis flow in PVTSizing can be used for mismatch exploration directly, for each set of random mismatch parameters can be regarded as a circuit corner. Once the PVT-phase optimization is completed, the trained RL agent $N(A, Q, k)$ is got, where A is the Actor network, Q is the Critic network, and k is the total number of PVT corners. The design solution X_{sel} that passes the full PVT condition test will be adopted as the starting point for the mismatch exploration. The mismatch exploration can be divided into 5 steps.

Step 1: Implement X_{sel} for MC sampling. The number of MC sampling points is k', which is about twice as much as k. Each MC sampling point represents a mismatch corner.

Step 2: Select k MC sampling points with the worst k FoMs, and replace the k PVT conditions in $N(A, Q, k)$ with k mismatch corners of k selected MC sampling points.

Step 3: Run multi-task RL as in PVT-phase optimization and get a new design solution X'_{sel} passing all mismatch corners.

Step 4: Test X'_{sel} with another k' MC sampling points. If X'_{sel} fails to pass all these k' mismatch corners, go back to step 2.

Step 5: Test X'_{sel} with full PVT conditions. If X'_{sel} fails to pass all PVT conditions, run PVT-phase optimization again and then return to step 1 of mismatch exploration.

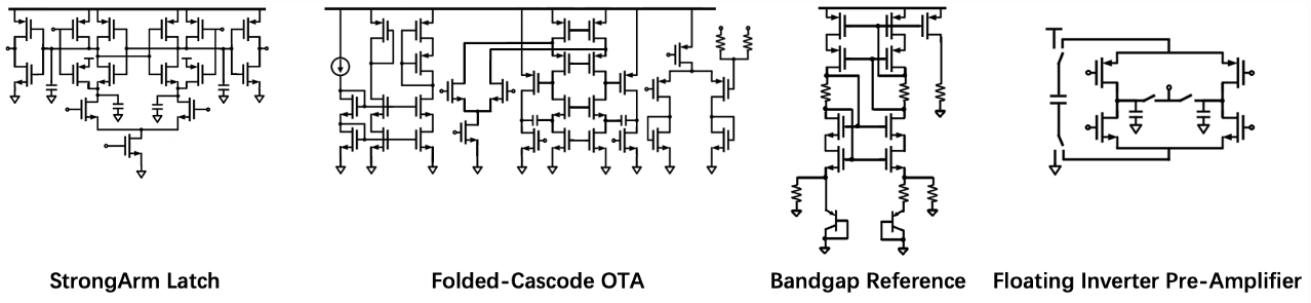

Fig. 4. Topologies of testcases.

The constraints of performance metrics in mismatch exploration differ slightly from those in PVT-phase optimization. For a statistical metric F_s, such as DC offset, that follows Gaussian distribution, if we want $F_s \sim N(\mu, \sigma)$, we can set the constraint to $F_s \in [\mu - 2\sigma, \mu + 2\sigma]$.

IV. Experimental Results

A. Testcases

Four real-world design examples are adopted to test the design efficiency of PVTSizing and compare it with other state-of-the-art tools. The testcases include the folded-cascode operational transimpedance amplifier (FC OTA), the strongArm latch (StrongArm), the bandgap voltage reference (BGR), and the floating inverter amplifier (FIA). The topologies of testcases are shown in Fig. 4.

Note that the testcases are chosen carefully to fully verify the capability of the proposed PVTSizing across various conditions. As for the technology node, StrongArm is designed with TSMC 28nm, while others are with TSMC 180nm. The advanced process typically has larger variations. Regarding circuit operation conditions, FC OTA and BGR are considered static circuits, which operate under static operating points. In contrast, StrongArm and FIA are fully dynamic and suffer more from various variations. For parameter sensitivity, BGR is highly sensitive to design variables. Fine adjustments are usually needed to cancel out the temperature coefficient of the output, even in manual design flow. Regarding the design space size, FC OTA has the largest design space, which is about 10^{55}. Others' vary from 10^{17} to 10^{24}.

Folded-cascode OTA. The FC OTA testcase is a high-gain static amplifier implemented in TSMC 180nm technology with 30 PVT conditions. It has 20 design variables, including 7 transistor widths, 7 transistor lengths, 4 transistor ratios, and 2 capacitor values. It has about 10^{55} possible points in the design space. The 30 PVT conditions consist of $\{TT, SS, FF, SF, FS\} \times \{1.6V, 1.8V\} \times \{-40\,^\circ\mathrm{C}, 27\,^\circ\mathrm{C}, 125\,^\circ\mathrm{C}\}$. The target metrics are listed below, which are much more advanced than the prior demonstration in [6].

$$C = \begin{cases} Gain \geq 100\mathrm{dB}, & UGB & \geq 30\mathrm{MHz}, \\ PSRR \geq 100\mathrm{dB}, & Phase\ margin & \geq 60\,^\circ, \\ CMRR \geq 100\mathrm{dB}, & Noise & \leq 30\mathrm{mV}, \\ Power \leq 1\mathrm{mW}. \end{cases}$$

StrongArm. The StrongArm Latch testcase is a dynamic comparator implemented in TSMC 28nm technology with 30 PVT conditions. It has 14 design variables, including 6 transistor widths, 6 transistor lengths, and 2 capacitor values. It has about 10^{24} possible points in the design space. The 30 PVT conditions consist of $\{TT, SS, FF, SF, FS\} \times \{0.81V, 0.9V\} \times \{-40\,^\circ\mathrm{C}, 27\,^\circ\mathrm{C}, 80\,^\circ\mathrm{C}\}$. The target metrics are listed below, with much lower reset/set delay compared to [6].

$$C = \begin{cases} Power \leq 40\mu\mathrm{W}, \\ Reset\ delay \leq 4\mathrm{ns}, \\ Set\ delay \leq 4\mathrm{ns}, \\ Noise \leq 120\mu\mathrm{V}. \end{cases}$$

Bandgap reference. The BGR testcase is a reference voltage source implemented in TSMC 180nm technology with 15 PVT conditions. It has 8 design variables, including 2 transistor widths, 2 transistor lengths, 3 resistor values and one resistor ratio. It has about 10^{23} possible points in the design space. The 15 PVT conditions consist of $\{TT, SS, FF, SF, FS\} \times \{3V, 4V, 5V\}$. The target metrics are listed below. Note that temperature variation has already been considered in the temperature coefficient.

$$C = \begin{cases} Temperature\ coefficient \leq 100\mathrm{ppm}, \\ Average\ current \leq 2\mu\mathrm{A}, \\ 1V \leq Output\ voltage \leq 2\mathrm{V}. \end{cases}$$

Floating inverter amplifier. The FIA testcase is an emerging dynamic amplifier implemented in TSMC 180nm technology with 30 PVT conditions. It has 6 design variables, including 2 transistor widths, 2 transistor lengths, and 2 capacitor values. It has about 10^{17} possible points in the design space. The 30 PVT conditions consist of $\{TT, SS, FF, SF, FS\} \times \{1.1V, 1.2V\} \times \{-40\,^\circ\mathrm{C}, 27\,^\circ\mathrm{C}, 80\,^\circ\mathrm{C}\}$. The target metrics are listed below, with similar energy consumption and noise requirements compared to the state-of-the-art design in [18].

$$C = \begin{cases} Energy/conv. \leq 1.5\mathrm{pJ}, \\ Noise \leq 70\mathrm{mV}. \end{cases}$$

We apply PVTSizing to these four testcases and record the number of samples and runtime. To provide a good baseline, *Sizing Over Corners* embedded in Virtuoso [12] from industry and RobustAnalog [6] from academia are also applied for the same specs. For PVTSizing, the batch size for training and sampling is set to 20 and 8, respectively. All circuits are simulated by Cadence Spectre. And PVTSizing is implemented with PyTorch.

B. Mismatch Exploration

To showcase the mismatch exploration capability of the proposed framework, FC OTA is sized over random mismatch with constraints modified as follows.

$$C = \begin{cases} Gain \geq 100\mathrm{dB}, & UGB & \geq 30\mathrm{MHz}, \\ PSRR \geq 80\mathrm{dB}, & Phase\ margin & \geq 60\,^\circ, \\ CMRR \geq 84\mathrm{dB}, & Noise & \leq 30\mathrm{mV}, \\ Power \leq 1\mathrm{mW}, & DC\ offset & \sim N(0, 0.8mV). \end{cases}$$

TABLE II
OPTIMIZATION RESULTS ON FOUR REAL-WORLD CIRCUITS

Testcases		FC OTA	StrongArm	BGR	FIA
# of samples	PVTSizing	**538**	**174**	**365**	**334**
	Virtuoso	1023	674	3232	764
	RobustAnalog	4202	2033	8238	2043
Sample efficiency improvement		**1.9×**	**3.8×**	**8.8×**	**2.2×**
Runtime	PVTSizing	**0.68h**	**0.25h**	**0.43h**	**0.29h**
	Virtuoso	1.14h	2.45h	1.63h	0.88h
	RobustAnalog	3.87h	3.00h	8.59h	2.93h
Time efficiency improvement		**1.6×**	**9.8×**	**3.7×**	**3.0×**
Success rate	PVTSizing	100%	100%	90%	100%
	Virtuoso	100%	100%	100%	100%
	RobustAnalog	80%	100%	20%	100%

In tests with success rate less than 100%, only data with successful optimization are included.

TABLE III
RESULTS OF ABLATION STUDY

Testcases		FC OTA	StrongArm	BGR	FIA
# of samples	Proposed	**538**	**174**	**365**	334
	w/o TuRBO	1035	375	902	386
	w/o pruning[1]	551	253	581	**301**
	w/o ZTM[2]	705	187	472	524
Runtime	Proposed	**0.68h**	**0.25h**	**0.43h**	0.29h
	w/o TuRBO	1.71h	0.68h	1.04h	0.35h
	w/o pruning	0.73h	0.45h	0.60h	**0.22h**
	w/o ZTM	1.08h	0.30h	0.57h	0.50h
Success rates	Proposed	**100%**	**100%**	**90%**	**100%**
	w/o TuRBO	90%	100%	10%	80%
	w/o pruning	100%	100%	70%	70%
	w/o ZTM	100%	100%	70%	90%

In tests with success rate less than 100%, only data with successful optimization are included.
[1]refers to Critic-assisted pruning.
[2]refers to zoom target metrics.

A statistical metric, DC offset, is added, and the constraints for PSRR and CMRR are relaxed. Note that both PSRR and CMRR are highly sensitive to random mismatch. When 1 σ and 3 σ of mismatch appear at the input stage, the ideal CMRR of this FC OTA is around 94dB and 85dB, respectively. So, making CMRR better than 84dB across all mismatch corners is not easy.

The effectiveness and efficiency of PVTSizing on mismatch exploration is evaluated and compared with *Sizing Over Corners* embedded in Virtuoso. They use the same X_{sel} as the starting point of mismatch exploration.

C. Results and Discussions

Tab. II shows the optimization results. Among all 4 testcases, PVTSizing achieves the highest sample efficiency and lowest runtime. Compared to Virtuoso and RobustAnalog, the sample efficiency and time efficiency are increased by 1.9×-8.8× and 1.6×-9.8×, respectively. The sample efficiency of the BGR case is improved significantly (by 8.8× compared to Virtuoso). However, its success rate is slightly lower than Virtuoso (90% for PVTSizing). This is because BGR is extremely sensitive to device sizes. In this case, the global searching method adopted by Virtuoso may realize a slightly higher success rate at the cost of low sample efficiency.

To further demonstrate the improvement brought by the proposed techniques, the ablation study is conducted in Tab. III. It shows that TuRBO and zoom target metrics both contribute to the sample efficiency and success rate. One may notice that the Critic-assisted pruning improves the success rate but reduces sample efficiency in the FIA case. This is because only data with successful optimization are included for sample efficiency and runtime calculation. As a result, the sample number with a success rate less than 100% is actually underestimated. If the success rate is taken into account, the sample efficiency with the Critic-assisted pruning is increased in all cases.

TABLE IV
RESULTS OF MISMATCH EXPLORATION ON FC OTA

Framework	# of samples	Runtime	Successed?
PVTSizing	**823**	**3.73h**	Yes
Virtuoso	5267	5.27h	Yes
Improvement	6.4×	1.4×	/

As shown in Tab. IV, the proposed PVTSizing demonstrates higher sample efficiency and lower runtime in mismatch exploration tasks.

V. CONCLUSION

In this paper, we proposed PVTSizing, a batch-sampling optimization framework for the PVT-robust analog sizing problem. PVTSizing utilizes TuRBO for initial sampling and multi-task RL agents for PVT optimizations. The Critic-assisted pruning and zoom target metrics are proposed to improve sample efficiency and reduce runtime. In addition to PVT corners, the proposed framework naturally supports mismatch exploration. Experiments on real-world circuits demonstrate that PVTSizing offers higher sample efficiency and time efficiency than prior tools from industry and academia.

REFERENCES

[1] J. Lan et al., "Effective Gain Analysis and Statistic Based Calibration for Ring Amplifier With Robustness to PVT Variation," TCAS-II, 2022.
[2] A. F. Budak, et al., "DNN-Opt: An RL Inspired Optimization for Analog Circuit Sizing using Deep Neural Networks," DAC, 2021.
[3] A. F. Budak, D. Smart, B. Swahn, and D. Z. Pan, "APOSTLE: Asynchronously Parallel Optimization for Sizing Analog Transistors Using DNN Learning," ASPDAC, 2023.
[4] K.-E. Yang et al., "Trust-Region Method with Deep Reinforcement Learning in Analog Design Space Exploration," DAC, 2021.
[5] H. Wang et al., "GCN-RL Circuit Designer: Transferable Transistor Sizing with Graph Neural Networks and Reinforcement Learning," DAC, 2020.
[6] W. Shi, et al., "RobustAnalog: Fast Variation-Aware Analog Circuit Design Via Multi-task RL," MLCAD, 2022.
[7] 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," TCAD, 2022.
[8] K. Touloupas and P. P. Sotiriadis, "LoCoMOBO: A Local Constrained Multiobjective Bayesian Optimization for Analog Circuit Sizing," TCAD, 2022.
[9] M. Liu, et al., "Parasitic-Aware Analog Circuit Sizing with Graph Neural Networks and Bayesian Optimization," DATE, 2021.
[10] B. He, S. Zhang, F. Yang, C. Yan, D. Zhou, and X. Zeng, "An Efficient Bayesian Optimization Approach for Analog Circuit Synthesis via Sparse Gaussian Process Modeling," DATE, 2020.
[11] Y. Yang et al., "Smart-MSP: A Self-Adaptive Multiple Starting Point Optimization Approach for Analog Circuit Synthesis," TCAD, 2018.
[12] Virtuoso Analog Design Environment User Guide, Cadence Design Systems, Inc., San Jose, CA, USA, 2020.
[13] D. Eriksson, et al., "Scalable global optimization via local Bayesian optimization." NeurIPS, 2019.
[14] J. MacQueen et al., "Some methods for classification and analysis of multivariate observations," in Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1. Oakland, CA, USA, 281–297, 1967.
[15] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," NeurIPS, 1999.
[16] T. Yu, et al., "Gradient surgery for multi-task learning," NeurIPS, 2020.
[17] B. Razavi, "The StrongARM latch [a circuit for all seasons]," IEEE Solid-State Circuits Magazine, 2015.
[18] X. Tang, et al., "An energy-efficient comparator with dynamic floating inverter amplifier," JSSC, 2020.