An Efficient Transfer Learning Assisted Global Optimization Scheme for Analog/RF Circuits

Zhikai Wang[1,2,†], Jingbo Zhou[2,*], Xiaosen Liu[1], and Yan Wang[1,*]
[1]Tsinghua University, [2]Baidu Research
wzk19@mails.tsinghua.edu.cn, zhoujingbo@baidu.com, wangy46@tsinghua.edu.cn

Abstract— Online surrogate model-assisted evolution algorithms (SAEAs) are very efficient for analog/RF circuit optimization. To improve modeling accuracy/sizing results, we propose an efficient transfer learning-assisted global optimization (TLAGO) scheme that can transfer useful knowledge between neural networks to improve modeling accuracy in SAEAs. The novelty mainly relies on a novel transfer learning scheme, including a modeling strategy and novel adaptive transfer learning network, for high-accuracy modeling, and greedy strategy for balancing exploration and exploitation. With lower optimization time, TLAGO can have a faster rate of convergence and more than 8% better performances than GASPAD.

I. Introduction

To reduce the time-to-market, previous works have reported many analog/RF circuit sizing algorithms. Among these methods, simulation-based optimizers usually leverage learning-based methods to explore design space [1]. Traditionally, model-free simulation-based algorithms, such as advanced differential evolution (DE) algorithm [2] and particle swarm optimization [3], perform abundant actual simulations to achieve good convergence. However, to achieve global optimization, these methods often require lots of SPICE simulation which leads to expensive costs.

To avoid abundant simulations when exploring design space, the parameter (inputs) to specification (outputs) surrogate model is popular for assisting optimizers as a "simulator" to predict circuit performances. Many state-of-the-art model-based optimization algorithms [4], [5] were reported in the literature. For example, GASPAD [5], using Gaussian Process Regression (GPR) as the surrogate model and DE as the global optimization engine, proposes a classical and efficient surrogate model-assisted evolutionary algorithm (SAEA). The main challenge of GPR-based global optimization methods is their huge modeling time requirement because GPR has the cubical complexity as the number of samples.

Oriented to multi-performance and stringent constrained commercial circuits, instead of using GPR, some deep learning surrogate models have attracted recent attention, such as efficient reinforcement learning-based DNN-Opt [1] and EESAB methods [6]. These algorithms show the excellent advantages of neural network (NN) methods in less optimization time and better optimization results due to low modeling costs and robust capability in high-dimensional modeling. On the other hand, PVT fluctuation, as a very critical problem, needs to be considered in optimization. Generally, we can consider the worst result under all corners as the final performance assessment. PVT simulations will increase sharply in circuit simulation costs and lead to higher non-linearity, which expand the demand for high-precision surrogate models. Thus, with regard to the NN-based methods further improving the model accuracy (evaluated by optimization results in this paper) with less modeling time has become an essential task.

Transfer Learning, which learns and transfers some knowledge from the source domain to the target domain, has been successfully applied to many modeling approaches [7], [8] in the analog/RF EDA community. Still, none is applied to online evolution sizing algorithms. In other words, although model-based evolutionary algorithms have been extensively studied, none has focused on model storage and reuse of knowledge. Our insight is that there exists a similarity between different online modeling sample tasks in SAEA. This work aims to explore how the transfer learning approach can effectively transfer knowledge across online models in SAEA algorithms.

We propose a Transfer Learning-Assisted Global Optimization scheme (named TLAGO for short). TLAGO obeys the classical SAEA structures like [5], [9] which use population elite strategy and surrogate model for sample-efficient analog/RF circuit optimizations. The key idea of TLAGO is to transfer reusable model knowledge for building high-accuracy model within the iterative optimization process.

At first, it uses a randomly initialized NN model as a surrogate model to provide well-trained source domain models. Importantly, the novelty of our approach lies in the incorporation of a specifically engineered adaptive transfer learning (ATL) network component, which is particularly crucial given that not all model knowledge is universally transferable across different domains, owing to the complexity of the circuits. To address this, the devised ATL network encompasses several key elements. Firstly, a primary network is constructed based on the source model, with the last layers removed, to facilitate continuous feature transfer. Secondly, a separate lightweight adapter is developed from scratch, enabling independent learning of the unique (private) features specific to the current target domain. Lastly, the remaining layers are responsible for predicting circuit performances.

When starting transfer learning, the most similar models are selected from history domains to help to construct the proposed ATL network. In the process of continuous transfer learning, common model knowledge is learned adaptively and private features learned by the adapter are discarded. To balance exploration and exploitation of the design space, a greedy strategy is used to provide uncertainty for selected promising simulation designs due to the existence of prediction errors.

Moreover, TLAGO adopts a pseudo-sample strategy pro-

[†]Part of this work was done when the first author was an intern at Baidu Research under the supervision of Jingbo Zhou. [*]Corresponding authors.

posed in [6] for data enhancement. To avoid over-fitting during the continuous transfer process, we use weight decay for training. We summarize our contributions as follows.

- To the best of our knowledge, this is the first work to leverage transfer learning as a surrogate model of online evolutionary algorithms.
- We focus on knowledge reuse between online models and propose a modeling strategy. We also propose a novel ATL network that can transfer shared features and forget private features adaptively.
- In order to balance exploration and exploitation of design spaces, we leverage greedy strategy to provide uncertainty for selection.

In Section II, we show the problem definition. In Section III, we introduce the proposed optimization scheme. The details of TLAGO are shown in Section IV, and the efficiency of TLAGO is demonstrated in Section V. Finally, we conclude this work in Section VI.

II. Background

A. Problem Definition

The circuit sizing can be formulated as a constrained one-goal optimization problem as follows.

$$\begin{aligned} & minimize \quad f(\mathbf{x}) \\ s.t. \quad & c_{min} \leq c_i(x) \leq c_{max}, \forall \quad i \in 1...M, \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^d$ represents design variables of circuits (d is the number of parameters), $f(\mathbf{x})$ denotes the Figure of Merit (FOM) of circuits and $c_i(\mathbf{x})$ means the $i-th$ circuit constraints. In this paper, we formulate normalized violation error VIO as a constraint satisfaction standard as follows.

$$\begin{aligned} VIO &= \sum_{i=1}^{M} VIO_i, \\ VIO_i &= \begin{cases} 0 & \text{if } c_i(\mathbf{x}) \text{ is satisfied} \\ \frac{|c_i(\mathbf{x})^* - c_i(\mathbf{x})|}{max(|c_i(\mathbf{x})^*|, |c_i(\mathbf{x})|)} & \text{if } c_i(\mathbf{x}) \text{ is not satisfied} \end{cases} \end{aligned} \tag{2}$$

where $c_i(\mathbf{x})$ and $c_i(\mathbf{x})^*$ are $i-th$ actual simulation values and constraint values, respectively. Clearly, $VIO \geq 0$, and $VIO = 0$ mean feasible solutions for which each constrained specification is acceptable to designers.

III. The Proposed TLAGO Scheme

A. Flow of TLAGO

TLAGO (Fig. 1) works as follows. First, given a certain design space (Lower bounds (LB) and upper bounds (UB) of design parameters) of analog/RF circuits, we use the Latin Hypercube sampling (LHS) method [10] to perform initialization and obtain α design parameters. Then, we run the SPICE simulator to generate samples. These samples will make the initial database. Then, we rank all simulated design samples using the criterion in [5] (first rank feasible solutions (VIO=0), then rank all feasible solutions by minimizing the FOM value) and select λ best designs as a parent population.

Next, it will start an online iteration. First, applying the robust DE/Best/1 strategy to perform mutation and crossover operations, and generating λ child designs. Then, we select τ designs and their labels from the sample database, which are nearest to the mean value of λ child design parameters (evaluated by Euclidean distance). The actual simulated designs and their performances will generate pseudo-samples for data enhancement from [1]. To build models for training and prediction, we propose an online transfer learning architecture that includes modeling strategies and different neural network structures. Depending on the strategy, TLAGO selects an artificial neural network (ANN) or adaptive transfer learning (ATL) network to train and predict. Note that we use weight-decay to avoid over-fitting when training the model. In the process of online iteration, storing pseudo-samples and trained models in a list, knowledge is passed down in a way of weight reuse. Then, the performances of λ child designs are predicted by the trained model. We rank all these predicted designs and select the next simulation sample by the proposed greedy selection rule. Next, we simulate the most promising design and add this sample to the database. After that, a novel online iteration will begin. If meeting a maximum number of iterations T, TLAGO will output the optimal sizing result.

IV. Implement Details

A. Transfer Learning in SAEAs

Why can the transfer learning methods be incorporated into SAEAs? In TLAGO, all samples are derived from the sample database, and the samples can be repeatedly used for models. Thus, online modeling samples usually are similar, especially in the later stage local region of optimization. The existence of this similarity makes transfer learning feasible.

We consider a source domain $\mathbf{D_s} = \{\mathbf{x_{si}}, \mathbf{y_{si}}\}_{i=1}^{N_s}$ and target domain $\mathbf{D_t} = \{\mathbf{x_{ti}}, \mathbf{y_{ti}}\}_{i=1}^{N_t}$, two online evolutionary sample datasets which are not the same. In our circuit sizing task, the inputs and outputs of the source and target online model have the same feature dimension. However, different domains correspond to different circuit behaviors. In other words, the probability distribution is different in each domain, i.e., $P_s(\mathbf{x_s}, \mathbf{y_s}) \neq P_t(\mathbf{x_t}, \mathbf{y_t})$. Therefore, this research problem belongs to the domain adaptation aspect in Transfer Learning.

Fine-tuning the pre-trained model has been recognized as an efficient learning method in domain adaptation. To be more specific, the trained source domain model weights will be taken as the priority and ideal initial solutions for the target model. This work will consider when to fine-tune and how to learn for target model f_t based on trained f_s.

B. Transfer Learning Modeling Scheme

1) Modeling Strategy: The proposed modeling strategy divides online modeling into three ways: 1) training ANN in a way of random initialization, 2) selecting the most similar model from the historical models for constructing ATL, and 3) continuously adaptive transfer learning.

In the early stage of optimization, a rough model is built in a wide range and simulation samples even contain some particularly bad designs from vast design regions. At this stage, the sample similarity of different online modeling domains is low, and the model knowledge may be difficult to retain and transfer as effective information. Therefore, TLAGO first performs randomly initialized ANN modeling and store these trained model for follow-up transfer learning.

It is difficult to ensure that adjacent online samples are always the most similar because samples found by optimizers

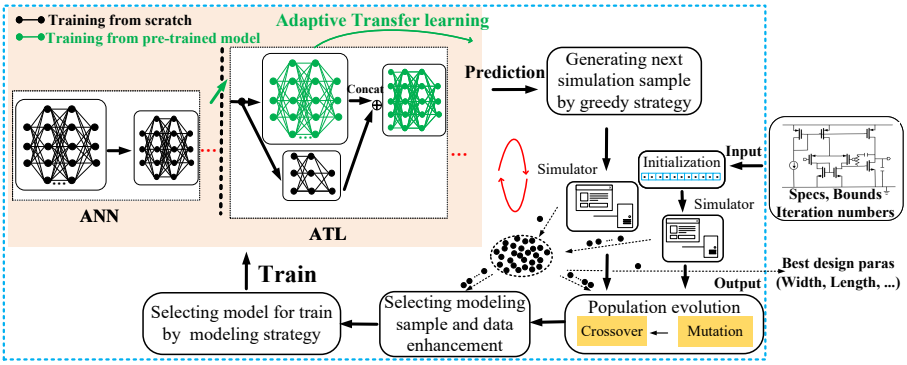


Fig. 1. The proposed TLAGO scheme

are in various regions. In this situation, transfer learning between less similar sample domains is invalid and will accumulate in the later optimization process. In order to avoid ineffective transfer learning and its cumulative negative impact, we perform model selection based on sample similarity in every iteration (introduced in Section IV-B-3). Based on the selected source domain model, we can build the ATL which will be fine-tuned.

Due to the existence of sample similarity, after the first fine-tuning from the selected model, we continuously fine-tune the last trained ATL model in each sub-region. The ATL uses the main network to perform transfer learning continuously, while learned private features by adapter are not reused. ATL method can perform adaptive knowledge transfer and cumulatively improve the accuracy of the model in each region.

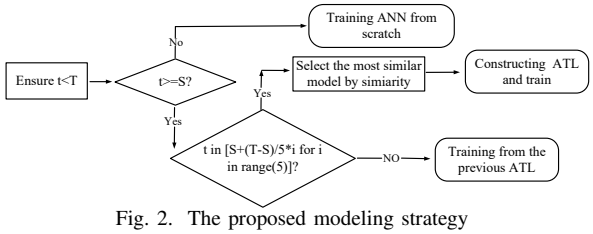


Fig. 2. The proposed modeling strategy

We divide the fine-tuned part into 5 subintervals, the modeling strategy for the $t-th$ modeling can be represented in Fig. 2. The start number S of the transfer learning process is related to modeling difficulty (dimensions of design parameters and target specifications) and modeling database. In TLAGO, we set it as $((M+1)d-\tau) \sim 2((M+1)d-\tau)$. If the calculated value of S is less than or equal to 2, it is regarded as 2.

2) Artificial Neural Network: For training ANN model more efficiently, we leverage a pseudo-sample method to provide more information instead of actual simulation samples. TLAGO uses an ANN model to predict all specifications (a vector $\mathbf{y}$ with $M+1$ dimensions, including one FOM and M constraints) by pseudo-sample inputs (a vector $\mathbf{x}$ with $2d$ dimensions). For τ modeling designs, we can obtain τ^2 modeling pseudo-samples.

$$x^{ij} = (x^i, x^j - x^i), y^{ij} = y^{j1}, y^{j2}, ..., y^{j(M+1)} \\ i = 1, 2, ..., \tau, j = 1, 2, ..., \tau \tag{3}$$

We leverage L-layer fully connected (FC) neural networks as ANN structure and use Exponential Linear Units (ELU) as the activation function of each layer.

3) Adaptive Transfer Learning Network: ATL consists of a pre-trained model and an adapter trained from scratch. The source pre-trained weights may be copied from historic source models (ANN or ATL). The pre-trained model can provide a prior initial solution for the current training task so as to greatly improve the modeling accuracy. Parallel adapter can learn private features to bridge the differences between the target domain and the source domain. In other words, the adapter is trained specifically for the target task. ATL also adopts a pseudo-sample method for data enhancement. mathematically, target domain model f_t can be obtained by the following relation:

$$\mathbf{y} = g_3(Concat(g_2(\mathbf{x}), g_1(\mathbf{x}))) \tag{4}$$

where g_1, g_2, and g_3 are FC networks with the ELU activation. g_1 is trained by fine-tuning source pre-trained model weight f_s (deleting the last layer for ANN) and represents well-learned features of the source domain. g_2 is a lightweight adapter generated by random initialization and represents the complementary features. g_3, a single layer, processes the output features learned by g_1 and g_2 and predicts final circuit performances. If the selected source model is ANN, g_3 can not be copied because of unequal dimensions and will be trained from scratch. However, it also will be copied and then fine-tuned when the source model is ATL.

In TLAGO, the similarity between different modeling samples is evaluated by the average distance of design parameters. We first compute the average of the inputs to the current modeling pseudo-sample. Note that the average pseudo-sample inputs actually represent the average of the design parameters, which is demonstrated in Eq. (6).

$$\frac{1}{\tau}\sum_{i=0}^{\tau}(x^i, 0) = \frac{1}{\tau^2}\sum_{i=0}^{\tau}\sum_{j=0}^{\tau}(x^i, x^j - x^i) \tag{5}$$

Next, we calculate the Euclidean distance between the mean value of the current sample and mean value of the historical modeling database. Based on the selected model which has the minimum value of Euclidean distance, we can construct the ATL structure.

When transferring knowledge between different design domains, some knowledge needs to be forgotten and other knowledge needs to be passed on due to the diversity of circuit behavior. In ATL, the main network g_1 and g_3 is responsible for transferable knowledge, and the parallel adapter g_2 needs to be forgotten. Moreover, we do not just freeze the pre-trained model f_s and leverage the parameter-efficient method to only train the adapter model for target domain modeling. Firstly, there may be huge differences and fluctuations in circuit performances between the early and final design spaces. We cannot simply pass on invariable early knowledge for all design spaces. Therefore, the model parameters must be adjusted to learn for the new data set. Moreover, the source

domain and the target domain have the same number of training samples. Because of the use of pseudo samples, we can easily obtain thousands of training samples. Therefore, these data can support us in retraining model weights that are copied from the source domain. Finally, the ANN model adopted has a low training cost, allowing us to fine-tune all model parameters.

Specifically, compared with g_2 which has a small network, g_1 has a more complex network depth and number of neurons. For g_1, two deep layers are sufficient, the number of neurons is set as 128, the input dimension is $2d$, and output dimension is 128. For g_2, the number of neurons equal is 32, and only one single FC layer with ELU. Thus, g_2 can be considered as a slave layer for g_1 network. The g_3 is a single-layer network.

4) Training Methods: In TLAGO, all models adopt the same train method. We train all ANN models with the Loss Back Propagation algorithm by calculating the mean squared error distance. The Adam optimizer is leveraged in TLAGO. Weight decay is employed as a regularization strategy to prevent overfitting by inducing a predefined reduction in the model's weights before each gradient descent iteration. The value of weight decay in Adam optimizer is set as 1e-6. In our numerical experiments, weight decay can effectively help TLAGO achieve better optimization results. All modeled specifications of analog/RF circuits are re-scaled to [0,1]. All models are trained with only 5 epochs (a very low cost), and the initial learning rate lr of Adam optimizer is set as 0.001.

C. Greedy selection

In TLAGO, we adopt the sorting rule proposed by [5] for predicted population designs. However, due to the existence of model prediction error, we cannot just pick the best solution and must provide a certain uncertainty to explore unknown design solution spaces (also avoid local optimum). ANN method cannot provide prediction uncertainty with the help of expected improvement and a lower confidence bound. To solve this problem, this Section proposes a new selection rule based on a greedy strategy.

When selecting the next simulation design based on ranked results, there is a small ε probability for the random selection from top-n potential values (including the most promising solution) and 1 -ε probability for the most promising solution. In TLAGO, the t+1-th simulation point is selected by the following equation based on the ranked population designs.

$$x_t^* \leftarrow \arg\max_{x_t} \quad x_t^\lambda$$

$$\mathrm{x}_{t+1} \leftarrow \begin{cases} x_t^* & \text{if } random(0,1) >= \varepsilon \\ random(top_n(x_t^\lambda)) & \text{if } random(0,1) < \varepsilon \end{cases} \quad (6)$$

where x_t^λ represents the $t-th$ child designs, x_t^* is the most promising solution, and x_{t+1} is the next selected simulation design. $n = 3$ and $\varepsilon = 0.01$ in this study.

V. Experimental Results

We will show the practical effectiveness of the TLAGO by three analog/RF circuits, two-stage operational amplifier under 180 nm process, lower noise amplifier (LNA) under 65 nm process, and three-stage Operational Transconductance Amplifier (OTA) under 65 nm process.

We compare our method with the well-known GASPAD and conventional DE methods. We can not use pseudo-sample technology for GASPAD because it clearly leads to a significant increase in modeling costs. For GASPAD, we use the authors' recommendations to determine the number of modeling samples, perform hyperparameter optimization, and build a single model for each performance. For TLAGO and GASPAD, we limit the maximum number of simulations to 500 ($\alpha + T$, initialization samples are included) in the first two cases. We increased the simulation budget to 600 for the third complex circuit. For DE, the simulation budget is set to 10000. We keep two decimal points for all design parameters during searches. All experiments are carried out 10 times to average out the random fluctuations.

A. Two-stage Operational Amplifier

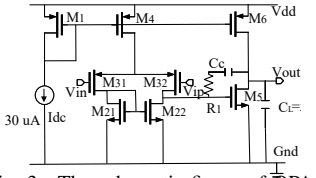
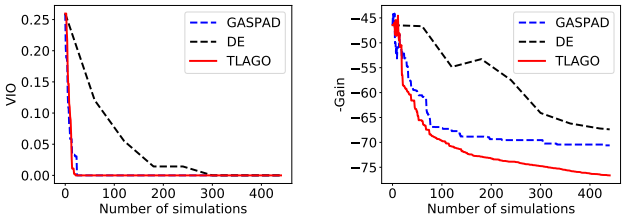


Fig. 3. The schematic figure of OPA

The first test case (shown in Fig. 3) is an OPA with 14 design parameters. The parameters and search regions are in Table I. The synthesis problem of OPA circuit is:

$$\begin{aligned} &minimize && -Gain \\ &s.t. && PM \geq 60^\circ \quad UGF \geq 40MHz \end{aligned} \quad (7)$$

For both TLAGO and GASPAD, the parameter settings: λ=α=τ=60. S of TLAGO is 2. They both perform 440 online iterations. Fig. 4 (a) shows the constraint convergence process



(a) (b)

Fig. 4. The optimized values vary simulations with different methods.

of all algorithms in 440 online simulations. We observe that all methods can search the feasible solutions (VIO=0). However, these surrogate model-assisted approaches look for feasible solutions much earlier than DE. Constraint specifications found by TLAGO: PM=61.27 deg and UGF=41.14M Hz.

From Fig. 4 (b), we can see FOM value may increase in the early stage because no feasible solution is found and the optimization task is to minimize the violation value (VIO). After 440 online simulations, TLAGO and GASPAD can search for better solutions than DE. However, compared with the two methods, TLAGO can obtain the optimal results (It is far superior to other methods). Moreover, as design spaces shrink, the similarity between modeling samples becomes stronger and stronger, and the advantages of TLAGO become more and more obvious.

TABLE I
THE SOLVED DESIGN PARAMETERS AND THEIR SEARCH REGIONS.

Parameters	LB	UB	Parameters	LB	UB
Width of M1∼ $M6$	1u	100u	Length of M1∼ $M6$	180n	600n
R1	100	500	C_c	1p	10p

In Table II, we list the mean, best and worst values of FOM. Note that "Success" means the number of runs finding feasible solutions in 10 total experiments. We have three observations from Table II. First, compared with the conventional DE method, the total run-time cost is greatly reduced, and TLAGO can achieve even better optimization results with 20X sample reduction and 13.2x run-time speed up. Moreover, TLAGO can find a higher gain (6.26 dB, 8.9%) compared to GASPAD with the same number of iterations. In addition, GASPAD spends more on modeling costs (1.66x) than TLAGO because of the cube time complexity of GPR (close to simulation time).

TABLE II
THE RESULTS ABOUT TLAGO AND COMPARED METHODS.

Technique	**TLAGO**	GASPAD	DE
Mean Gain (dB)	**76.86**	70.60	76.02
Max Gain (dB)	**78.39**	72.25	76.79
Min Gain (dB)	**74.72**	67.18	75.21
Mean VIO	0	0	0
Success	10/10	10/10	10/10
Number of simulations	500	500	10000
Modeling time	**0.098h**	0.163h	0
Simulation time	0.189h	0.189h	3.78h
Total run-time	**0.287h**	0.352h	3.78h

B. Lower Noise Amplifier

The second test case is RF LNA circuit (Fig. 5) with 12 design parameters shown in Table III. 9 corners (TT/SS/FF&27° C/0° C/$-45°$ C) are considered in this circuit. We added more

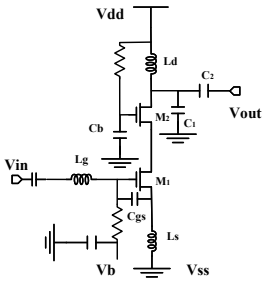


Fig. 5. The schematic figure of LNA

constraint specifications to verify the efficiency of TLAGO. The synthesis problem of LNA is:

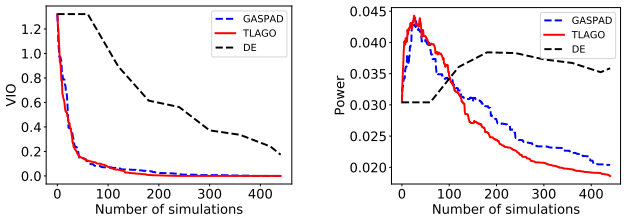
$$\begin{aligned} &minimize \quad Power \\ &s.t. \quad S21 \geq 8dB, \quad S11 \leq -8dB \\ &S22 \leq -8dB, \quad S12 \leq -30dB, \quad NF \leq 3dB \end{aligned} \tag{8}$$

For both TLAGO and GASPAD, as case 1, λ=α=τ=60.

TABLE III
THE SOLVED DESIGN PARAMETERS AND THEIR SEARCH REGIONS.

Parameters	LB	UB	Parameters	LB	UB
Width of M1/M2	1u	5u	Finger of M1/M2	5	50
Width of C1∼ $C3$	5u	30u	Width of Cgs	5u	30u
Width of Lg-Ls-Ld	4u	10u	Radius of Lg-Ls-Ld	20u	60u

We can make a novel observation from Fig. 6. Given the same simulation budget, TLAGO can find a feasible solution more quickly and still find a better FOM value compared to other reference methods. TLAGO, using greedy strategy and benefiting from continuous model accuracy improvement, can find the optimal global design solution.



(a) (b)

Fig. 6. The optimized values vary simulations with different methods.

TABLE IV
THE RESULTS ABOUT TLAGO AND COMPARED METHODS.

Technique	**TLAGO**	GASPAD	DE
Mean power (A)	**0.0186**	0.0204	0.0209
Max power (A)	**0.0241**	0.0278	0.0242
Min power (A)	**0.0155**	0.0155	0.0160
Mean VIO	0	0	0
Success	10/10	10/10	10/10
Number of simulations	500	500	7200
Modeling time	**0.133h**	0.279h	0
Simulation time	2.65h	2.65h	38.2h
Total run-time	**2.78h**	2.93h	38.2h

From Table IV, we can see that TLAGO is still efficient in optimization time and solution quality compared with other methods. Moreover, for six specifications, lots of modeling time is spent by GASPAD. With better average optimization results (8.8%) and the same simulation numbers, TLAGO can significantly reduce power consumption compared to GASPAD and at a small modeling cost (0.279h to 0.133h). Finally, performing over 120 generations of evolution (7200 simulations), DE costs more than 38.2h to find comparable results to TLAGO. This also shows that it is very difficult to optimize power in complex RF circuit with PVT experiments. However, TLAGO can obtain even better performances and save more than 13.74x total run-time than DE. There, constraints found by TLAGO: S21=13.36 dB, S11=-9.08 dB, S22=-8.21 dB, S12=-31.33 dB, and NF=2.83 dB.

C. Three-stage Operational Transconductance Amplifier

The final test case is a three-stage OTA shown in Fig. 7. The circuit has 24 design parameters (Table V). We run simulations

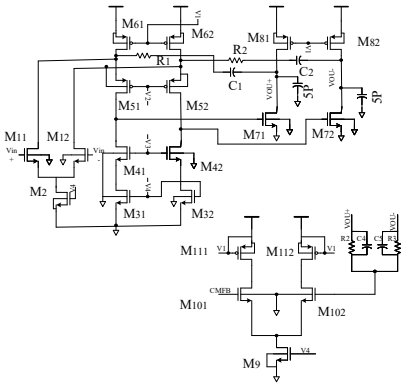


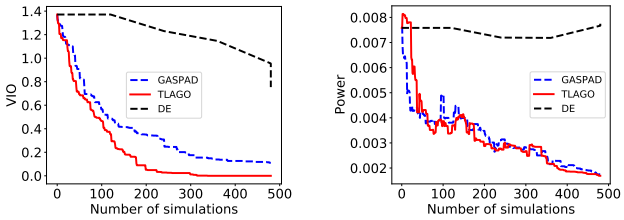
Fig. 7. The schematic figure of OTA

under 9 corners to obtain performances. The synthesis problem of OTA is:

$$\begin{aligned} &minimize \quad Power \\ &s.t. \quad Gain \geq 50dB, \quad PM \geq 50° \\ &UGF \geq 40MHz, \quad Loop \quad Gain \geq 50dB \\ &PM\&CM \geq 50°, \quad UGF\&CM \geq 60MHz \end{aligned} \tag{9}$$

TABLE V
THE SOLVED DESIGN PARAMETERS AND THEIR SEARCH REGIONS.

Parameters	LB	UB	Parameters	LB	UB
Width of M11∼ $M112$	1u	200u	Length of M11∼ $M112$	100n	300n
R1/R2	100	500	C1/C2	1p	5p



(a) (b)

Fig. 8. The optimized values vary simulations with different methods.

For both TLAGO and GASPAD, the parameter settings: λ=α=τ=120. S of TLAGO is 50. First of our observations from Fig. 8 and Table VI, given the upper limit of 600 simulations, consuming less modeling time compared with GASPAD, TLAGO can find 10 times feasible solutions in this difficult task. There, mean constraints found by TLAGO: Gain=54.17 dB, PM=77.54 deg, UGF=58.38 MHz, Loop Gain=62.52 dB, PM&CM=55.3 deg, and UGF&CM=67.45 MHz. However, in this complex high-dimension circuit, GASPAD has the worst performances (VIO>0) and cannot find feasible solutions in all runs. It can be seen from Fig. 8 (a) that TLAGO is far superior to GASPAD in the convergence rate of constraint satisfaction. The power value (Fig. 8 (b)) found by GASPAD is invalid because the constraints are not satisfied. Finally, TLAGO is much better than other methods considering total time and optimization results and can find a globally optimal solution similar to DE with 7320 simulations. However, the search for such a solution is a long and tedious process using DE (11.65x run-time than TLAGO).

D. Other Study

Based on the TLAGO, we change the modeling method and perform more experiments on the three circuit examples with (1) **ANN**: Greedy selection is not used, a surrogate model is ANN (Section IV-B-2)) trained with scratch. (2) **ANN + G**: Greedy selection is used, surrogate model is ANN trained with scratch. (3) **TL**: Greedy selection is not used, surrogate model is the proposed transfer learning modeling scheme. To be fair, they also use pseudo-sample and weight decay techniques. We perform the same hyperparametric effort for all methods.

TABLE VI
THE RESULTS ABOUT TLAGO AND COMPARED METHODS.

Technique	**TLAGO**	GASPAD	DE
Mean power (A)	**0.00169**	0.00175	0.00169
Max power (A)	**0.00269**	0.00235	0.00284
Min power (A)	**0.00097**	0.00106	0.00104
Mean VIO	0	0.11	0
Success	10/10	7/10	10/10
Number of simulations	600	600	7320
Modeling time	**0.43h**	0.47h	0
Simulation time	9.14h	9.14h	111.45h
Total run-time	**9.57h**	9.61h	111.45h

TABLE VII
STUDY ON TRANSFER LEARNING SHCEME AND GREEDY STRATEGY.

	Performances	ANN	ANN+G	TL	**TLAGO**
Case 1	Mean Gain(dB)	70.61	70.78	76.33	**76.86**
	Mean VIO	0	0	0	0
Case 2	Mean Power (A)	0.0265	0.0251	0.0196	**0.0186**
	Mean VIO	0.0275	0.0194	0	0
Case 3	Mean Power (A)	0.001825	0.001828	0.001977	**0.00169**
	Mean VIO	0.1	0.0157	0.0073	0

From Table VII, if only ANN is used, all examples achieve the worst optimization results due to inadequate modeling. In the second and third cases, feasible solutions can even not be found. However, once the proposed model architecture is used, the improvement of modeling precision can lead to great effects on final circuit performances. It also can be seen that using a greedy strategy can help ANN or TL to find better solutions by balancing exploration and exploitation (avoid falling into local optimality). The greedy strategy can help ATL architecture to achieve better global optimizations.

VI. CONCLUSIONS

This paper has proposed a transfer learning surrogate model-assisted global optimization algorithm which has been demonstrated as an efficient sizing method for analog and RF circuits. We make experiments on three circuits even considering PVT corners. With lower optimization time, TLAGO has a faster rate of convergence and more than 8% better performances than GASPAD. TLAGO can achieve even better optimization results with a few simulations and has a more than 11x run-time speedup than conventional DE methods.

ACKNOWLEDGMENTS

Thanks to the mentors and interns of Baidu Research for their invaluable assistance. This work was supported by a grant from the National Key R&D Program of China (*Project No. 2022YFB4401100*).

REFERENCES

[1] A. F. Budak, P. Bhansali, B. Liu, N. Sun, D. Z. Pan, and C. V. Kashyap, "DNN-Opt: An RL Inspired Optimization for Analog Circuit Sizing using Deep Neural Networks," in *58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 1219–1224.

[2] 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.

[3] R. A. Vural and T. Yildirim, "Analog circuit sizing via swarm intelligence," *AEU: Archiv fur Elektronik und Ubertragungstechnik: Electronic and Communication*, no. 9, p. 66, 2012.

[4] 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.

[5] B. Liu, D. Zhao, P. Reynaert, and G. G. 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.

[6] A. Budak, M. Gandara, W. Shi, D. Pan, 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. PP, no. 99, pp. 1–1, 2021.

[7] Q. Zhang, S. Su, J. Liu, and M. S.-W. Chen, "CEPA: CNN-based early performance assertion scheme for analog and mixed-signal circuit simulation," in *Proceedings of the 39th International Conference on Computer-Aided Design (ICCAD)*, 2020, pp. 1–9.

[8] J. Liu, M. Hassanpourghadi, Q. Zhang, S. Su, and M. S.-W. Chen, "Transfer learning with Bayesian optimization-aided sampling for efficient AMS circuit modeling," in *Proceedings of the 39th International Conference on Computer-Aided Design*, 2020, pp. 1–9.

[9] 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, 2021.

[10] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.