Joint Optimization of Sizing and Layout for AMS Designs: Challenges and Opportunities

Ahmet F. Budak
The University of Texas at Austin, Austin, TX, USA
ahmetfarukbudak@utexas.edu

Keren Zhu
The University of Texas at Austin, Austin, TX, USA
keren.zhu@utexas.edu

Hao Chen
The University of Texas at Austin, Austin, TX, USA
haoc@utexas.edu

Souradip Poddar
The University of Texas at Austin, Austin, TX, USA
souradippddr1@utexas.edu

Linran Zhao
The University of Texas at Austin, Austin, TX, USA
lrzhao@utexas.edu

Yaoyao Jia
The University of Texas at Austin, Austin, TX, USA
yjia@austin.utexas.edu

David Z. Pan
The University of Texas at Austin, Austin, TX, USA
dpan@ece.utexas.edu

ABSTRACT

Recent advances in analog device sizing algorithms show promising results on the automatic schematic design. However, the majority of the sizing algorithms are based on schematic-level simulations and layout-agnostic. The physical layout implementation brings extra parasitics to the analog circuits, leading to discrepancies between schematic and post-layout performance. This performance gap raises questions about the effectiveness of automatic analog device sizing tools. Prior work has leveraged procedural layout generation to account for layout-induced parasitics in the sizing process. However, the need for layout templates makes such methodology limited in application. In this paper, we propose to bridge automatic analog sizing with post-layout performance using state-of-the-art optimization-based analog layout generators. A quantitative study is conducted to measure the impact of layout awareness in state-of-the-art device sizing algorithms. Furthermore, we present our perspectives on the future directions in layout-aware analog circuit schematic design.

CCS CONCEPTS

• **Hardware → Electronic design automation**; **Methodologies for EDA**.

KEYWORDS

Electronic Design Automation (EDA), analog circuit synthesis, analog sizing automation, analog layout automation, layout-aware sizing

ISPD '23, March 26–29, 2023, Virtual Event, USA

ACM ISBN 978-1-4503-9978-4/23/03.
https://doi.org/10.1145/3569052.3578929

ACM Reference Format:
Ahmet F. Budak, Keren Zhu, Hao Chen, Souradip Poddar, Linran Zhao, Yaoyao Jia, and David Z. Pan. 2023. Joint Optimization of Sizing and Layout for AMS Designs: Challenges and Opportunities. In *Proceedings of the 2023 International Symposium on Physical Design (ISPD '23), March 26–29, 2023, Virtual Event, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3569052.3578929

1 INTRODUCTION

Lacking automation tools, analog integrated circuit (IC) design is a labor-intensive process. As a crucial stage in analog circuit synthesis, device sizing has lately attracted a great deal of academic attention due to industry demands and the development of machine learning (ML)-inspired methods. The size of devices, such as transistor width and capacitor geometry, can be critical to the circuit performance, power consumption, and area (PPA). In manual design, a combination of design experience and simulator feedback often determines device sizing.

Attempts to automate analog scaling date back decades. It may be conventionally divided into equation-based and simulation-based techniques. Equation-based approaches quantify and direct the search of the optimal PPA using embedded equations. It is computationally efficient; however, the equation may be inaccurate, particularly for today's advanced technology nodes. Simulation-based methods use simulations to explore the design space. They utilize simulators to obtain precise PPA. Recent research trend in analog sizing introduces machine learning (ML) to simulation-based methodology [1]. The adoption of ML techniques has increased the optimization efficiency and improved the capability of automatic analog device sizing.

However, in addition to device sizing, analog circuit performance is also sensitive to layout implementation [2]. Layout effects exacerbate the challenges in advanced processes due to design rule complexity and large layout impacts on circuit parasitics. There have been several attempts to enable automatic sizing to be parasitic- or layout-aware. However, the existing methods either require additional efforts on implementing layout templates [3] or use less

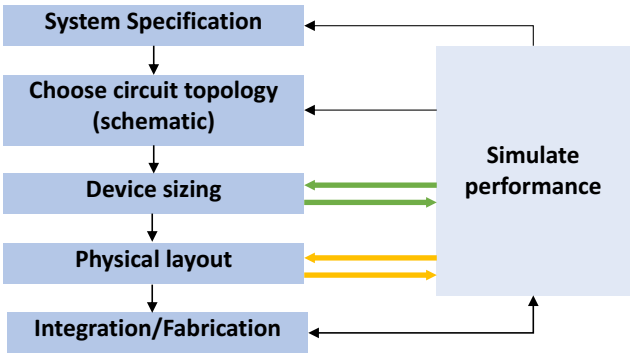


Figure 1: Analog IC Design Flow

accurate parasitic prediction model [4]. A full-scale general-purpose layout-aware analog sizing method is yet to be discovered.

In this paper, we present a comprehensive study on post-layout performance-based design parameter optimization of analog ICs. Leveraging the state-of-the-art analog sizing method [5] and automatic analog layout framework [6, 7], our framework jointly considers sizing and layout implementation. We conduct experiments on a Miller operational transconductance amplifier (OTA) and demonstrate the importance of considering layout effects in automatic sizing.

The rest of the paper is organized as follows. Sec. 2 introduces the backgrounds. Sec. 3 reviews the existing parasitic-aware and layout-aware analog sizing methods. Sec. 4 describes our joint optimization flow and presents several case studies. Sec. 5 discusses and gives our perspectives in the research of layout-aware analog sizing. Finally, Sec. 6 concludes the paper.

2 BACKGROUND

Analog IC design cycle typically follows a pattern similar to the one visualized in Figure 1. An iterative process between the human designer and computer simulations is conducted to finalize a design that satisfies the intended performance requirements. A large portion of design time is spent on the sizing and layout phases, where multiple iterations are possible due to potential loop-backs in the design flow. Therefore many automation flows with accompanying optimization algorithms are introduced to solve analog sizing and layout generation problems. This section is dedicated to reviewing such methods and preliminaries. We include a more elaborate review of the analog sizing tool, DNN-Opt, and analog layout generator, MAGICAL since these tools are leveraged in developing the joint optimization framework presented in this paper.

2.1 Problem Formulation

We formulate the analog circuit sizing task as a constrained optimization problem succinctly as below.

$$\begin{aligned} &\text{minimize} \;\; f_0(\mathbf{x}) \\ &\text{subject to} \; f_i(\mathbf{x}) \leq 0 \quad \text{for } i = 1, \ldots, m \end{aligned} \tag{1}$$

where, $\mathbf{x} \in \mathbb{R}^d$ is the parameter vector and d is the number of design variables of sizing task. Thus, $\mathbb{R}^d$ is the design space. $f_0(\mathbf{x})$ is the objective performance metric we aim to minimize. Without loss of generality, we denote i^{th} constraint by $f_i(\mathbf{x})$.

Through this paper, we will evaluate the quality of a design by defining a Figure of Merit (FoM) in the following form:

$$\text{FoM}(\mathbf{x}) = w_0 \times f_0(\mathbf{x}) + \sum_{i=1}^{m} \min\left(1, \max(0, w_i \times f_i(\mathbf{x}))\right) \tag{2}$$

where w_i is the weighting factor. Note, a $\max(\cdot)$ clipping is used for equating designs after constraints are met, and $\min(\cdot)$ is used to prevent single constraint violation from dominating FoM value.

2.2 Analog Sizing

Different approaches have been adapted to solve analog sizing tasks in the past. One adaption is to use domain expertise where designer experience and device characteristics are translated into equations and procedures[8],[9]. However, this approach depends highly on accumulated experience and requires custom modifications for each topology and technology node. Therefore, such methods are not scalable and inefficient.

Another mainstream approach is to map the sizing problem into an optimization problem where circuit performances are handled as objective and constraint functions. This stream has two classes: equation-based methods and simulation-based methods. Equation-based methods approximate performance functions with convex or non-convex regression models, then numerical optimization methods are utilized as solvers to find optimal input variables. In the past, Geometric Programming[10], [11] and Semidefinite Programming (SDP) relaxations [12] are used to find optimal sizing of analog circuits by employing posynomials as performance models. However, with the advanced device technology and complex circuit structures, regressed equations highly deviate from accurate simulation values. On the contrary, simulation-based methods evaluate iteration points via real simulations and, therefore, have become much more popular among recent algorithms.

In the past, population-based methods are used to solve the sizing problem. Among those, particle swarm optimization (PSO) [13] and advanced differential evolution [14] were examples of popular approaches. These methods provide proximity to the global optima, but they burn a large number of expensive simulations to achieve this. Due to their sample-inefficient nature, these methods are not applicable to designs with time-expensive simulations. To mitigate sample inefficiency issue, model-based and learning-based methods are becoming increasingly popular in the field. Since these methods employ/train proxies between the solution space and performance space, they are more efficient in exploring the solution space. Usually, the proxy method for model-based methods is called a surrogate model. A typical surrogate model is Gaussian Process Regression (GPR) [15], which is a well-studied model in Bayesian Optimization (BO) field [16] and is adapted by several analog sizing algorithms. For example, GASPAD is a hybrid algorithm using a combination of evolutionary space exploration and GPR surrogate-based selection [17]. WEIBO method also employs GPR as a surrogate and introduces a Bayesian Optimization framework where a weighted acquisition function is tailored to comply with the performance-constrained nature of sizing problem [18]. One main drawback of GPR modeling is that it has cubic complexity in

the number of samples, $O(N^3)$, which brings complications when the collected sample number gets larger.

Other model-based optimization methods utilized ANN for circuit performance modeling. In [19], ANN is used as a proxy for the circuit simulator, and performance estimates obtained by the trained model are used to guide an evolutionary optimization framework efficiently. Another ANN-boosted evolutionary method is presented in [3] where the trained ANN is employed as a performance classifier. In [20], authors proposed to enhance the efficiency of a Genetic Algorithm based optimizer where the local minimum search is conducted via the help of trained ANN. Another method that utilizes neural nets is ESSAB [21], where ANN training is combined with a data augmentation method to reduce the need for the required number of samples for accurate training. A different approach is adapted for building an ANN model in [22], where circuit specifications are used at the input nodes, and design variables are taken from the output nodes. To train and work with such a model, a large dataset is generated for offline training of the ANN model.

Recently supervised and reinforcement learning (RL) algorithms have been applied in the field [5, 21, 23, 24]. GCN-RL [23] is a Graph Neural Network (GNN) algorithm where state representation is built via device index, type, and selected electrical properties. They also propose methods to transfer the optimization experience between different topologies and processes. AutoCkt [24] is a discrete action space policy gradient method. The RL agent is trained on different optimization tasks where the task is randomly sampled from a predefined set. The trained agent is then tested for the particular tests during deployment. They both provide competitive results compared to other black-box methods; however, GCN-RL requires thousands of simulations to converge, and AutoCkt needs to be trained before deployment, which is SPICE intensive. DNN-Opt [5] is introduced as an RL-inspired supervised learning optimization method that shows high sample efficiency and can be trained during optimization.

While learning-based methods (RL, DNN, Bayesian) target sample efficiency, in practice, SPICE simulations may be very costly, and the whole synthesis process is time budgeted, so a time-efficient method is sought. One way to gain efficiency is to make use of computationally cheap simulations. In [25], a multi-fidelity surrogate optimization technique is proposed where it uses fast RC simulations to reduce the cost of electromagnetic simulations. Another approach is to impose custom conditions or procedures before running particular simulations. In [26], an example of this approach is applied as a PVT exploration strategy, which is helpful for checking pass/fail conditions of specifications across several corners. In [27], the APOSTLE framework is introduced in the sizing problem, where real-time efficiency is provided via an asynchronously parallel optimization framework and a theory-based expensive simulation bypassing strategy.

2.3 DNN-Opt

DNN-Opt [5] is a deep neural network analog sizing framework. Being an RL-inspired optimization algorithm, DNN-Opt searches a continuous action space to find an optimal solution for the sizing problem. In this paper, we employ DNN-Opt as a black-box optimization algorithm. We first use it in its original form to size the circuit based on schematic-level simulations. Then, we modify the circuit performance generation procedure and use DNN-Opt to optimize design variables based on post-layout performance.

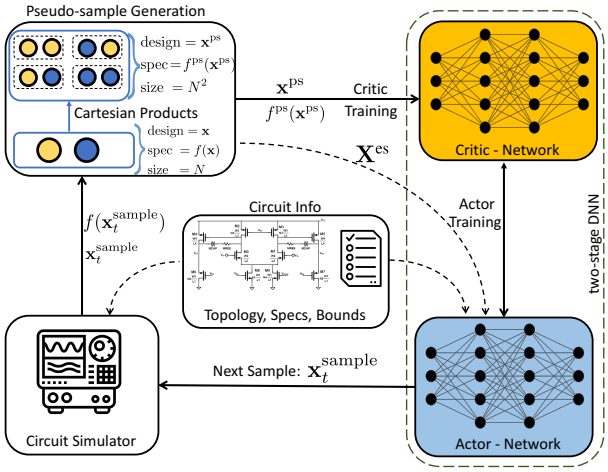


Figure 2: DNN-Opt Framework [5]

The core architecture for DNN-Opt is provided in Figure 2. It inputs a circuit topology, a set of design variables and search bounds, and performance criteria to optimize analog circuits. It interacts with the circuit simulator to obtain the circuit performance through optimization iterations, and the final output is the optimized design parameter values.

The search algorithm of DNN-Opt consists of two neural networks: actor and critic networks. The actor-network conducts a guided search in the sample space and determines the next sample point. The critic-network serves as a proxy to the real circuit simulations and is trained by the data created via *pseudo-sample* generation mechanism. Since the critic-network is the regression network for circuit performance metrics, it is trained by the mean squared error expressions obtained by the difference between network predictions and real simulation outputs. Once the critic training is done, the actor is trained based on the fixed critic. The analogy here is that the critic-network behaves as a surrogate model for the design space, and the actor-network searches within this model to find potential *good* designs. In that perspective, the state representation of the environment is the vector of design variables, and actions are the change vector for the design variables.

2.4 Analog Layout Automation

The physical layout implementation stage in the current analog IC design is still mostly manual, labor-consuming, and error-prone, placing constraints on turnaround time. Procedural layout generation and optimization-based layout synthesis are two paradigms that describe AMS Layout automation techniques [2].

Utilizing pre-designed parameterized layout templates, procedural layout generators migrate layouts for various manufacturing technologies and device sizes. Berkeley Analog Generator (BAG) [28] is a framework of procedural layout generation. This method typically requires designers to parameterize device placement and route (such as routing topology, metal layer, and via cuts), allowing

designers to modify device sizing. Nonetheless, it needs a large amount of human labor to develop generic layout templates or PCELLs, where device layouts are still manually coded to be placed and routed.

In generating layouts, optimization-based techniques employ place-and-route (PNR) algorithms to optimize area, power, and certain performance metrics. These methods have varying degrees of generality and are designed for various application scenarios. In contrast to template-based layout creation, optimization-based methods produce fully automated layout solutions without the need for additional effort to build layout templates. To assure layout quality, it is necessary to recognize constraints such as device symmetry, building block symmetry, and common-centroid matching.

Developing open-source end-to-end frameworks, such as MAGICAL [6] and ALIGN [29], is an emerging trend in optimization-based AMS circuit layout automation. MAGICAL and ALIGN both include a complete module generator, analog placer, and detailed router. They have demonstrated success in automating building block-level analog circuits with minimal or no human intervention. These general-purpose analog layout generators can generate the layouts in automatic sizing flow and, therefore analog sizing to be layout-aware. In this paper, we leverage the open-source MAGICAL framework as a bridge to study the joint optimization of sizing and layout for AMS designs.

2.5 MAGICAL

MAGICAL is an open-source end-to-end framework for AMS layout synthesis that aims to automatically build layouts from netlists with no-human-in-the-loop. In a few years, the MAGICAL system has evolved from the initial version with limited capability [6] to the silicon-proven MAGICAL 1.0 release [7].

MAGICAL automatically extracts layout constraint, generates the device layouts, places the modules, and routes the wires. Figure 3 illustrates the flow of MAGICAL for generating the layout for a block-level analog circuit. The automatic symmetry constraint generation is based on heuristic algorithm for device-level constraints [6] and graph similarity [30] for the system-level. The placement engine is based on a non-linear programming-based global placer and a linear programming-based legalizer [31]. The MAGICAL detailed router is based on an obstacle-aware path-finding algorithm [32]. The paper [33] gives a detailed overview of the MAGICAL algorithm.

3 RECENT DEVELOPMENTS

There have been efforts to enhance automatic sizing with parasitic or layout awareness. The existing methods use post-layout performance as the optimization objective of simulation-based analog sizing algorithms so that the device sizing can take layout effect into consideration. Their methodologies mainly fall into two categories.

(1) *Layout-aware sizing with procedural layout generation.* An automatic analog sizer is combined with a procedural layout generator. The flow generates the layouts based on a template for a set of sizing, and the post-layout performance is obtained from the generated results.
(2) *Parasitic-aware sizing with parasitic prediction.* A machine learning model to predict layout parasitics from pre-layout netlists is trained. The automatic analog sizing flow can then use the prediction model as a vehicle to model the layout effect.

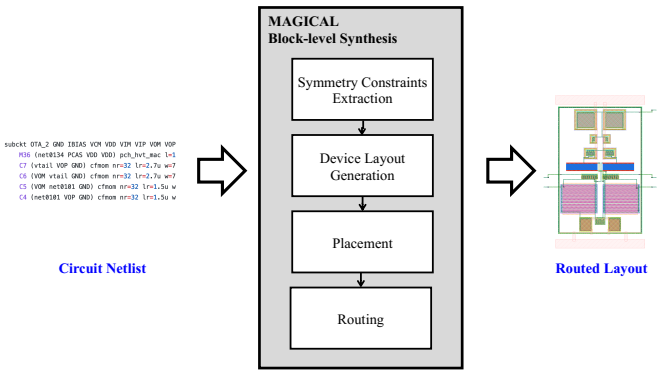


Figure 3: The MAGICAL flow for block-level layout generation.

The rest of this section gives an overview of the recent developments in academia.

3.1 With Procedural Layout Generation

A procedural analog layout generator uses a manually scripted layout generator to enable automatic sizing to generate the layouts for a template schematic. Both two recent methods, AutoCkt [24, 34] and BagNet [3], leverage the BAG [28] framework as the layout generator.

AutoCkt educates RL agents to be knowledgeable of the full design space. It uses BAG to enable layout awareness. Consequently, the agent is able to select the optimal course of action at a particular moment in the design process. Multiple trajectories teach the agent how to create a circuit. The agent either reaches the target or ends the trajectory because the maximum number of steps has been reached. Following training, the agent is assigned to an unseen target. The more successful the training, the higher the success rate agent can achieve during the deployment.

BagNet, on the other hand, is based on an evolutionary algorithm with a deep neural network (DNN)-based oracle to boost the search efficiency. It contains three core components: (1) an evolutionary engine to generate offspring, (2) the BAG layout generator, and (3) a DNN model acting as an oracle. Based on the existing population, the evolutionary algorithm produces the following generation of children. The DNN-based oracle evaluates the produced candidates and eliminates those with lower scores. After the discriminating step, the resulting offspring are implemented in layouts using BAG and simulated to acquire the metrics. The simulated performance is then utilized to fine-tune the oracle and direct the next generation in the evolutionary process.

The procedural layout generator-assisted analog sizing can obtain accurate post-layout performance. However, they require manually scripted layout templates.

3.2 With Parasitic Prediction

Recent advances in machine learning have enabled statistical and data-driven techniques to predict layout parasitics directly from circuit schematics with high accuracy. ParaGraph [35] converts circuits into heterogeneous graphs and uses a graph neural network (GNN) model to predict net parasitic capacitance. MLParest [36] uses Random Forest models to predict resistance and lumped parasitic capacitance.

The work [4] leverages the Paragraph prediction model to enable parasitic-aware sizing. It further proposes an improved performance surrogate model using graph embeddings from the pre-trained parasitic graph neural network as additional parasitic information. The ParaGraph GNN model encodes the latent information of the circuit parasitics with graph embeddings. With the performance model, With the surrogate model, the automatic sizing framework can therefore enable parasitic awareness and mitigate the layout effect.

The ML-based parasitic prediction and performance prediction provides an efficient vehicle to take layout effect into the sizing loop. However, the accuracy of predicting layout effects is still a challenge without layout implementation in both the parasitic prediction [37] and performance prediction [38] tasks.

4 LAYOUT IN THE LOOP SIZING

The majority of the research to automate analog design flow (Fig 1) focuses on individual phases and does not address the effects of preceding and following phases on each other. As reviewed in Section 2, analog automation literature is populated with mostly sizing-only or layout-only efforts. This practice has three major undesired consequences:

(1) Being agnostic of other phases damages the reliability of automation.
(2) The total scope of optimization is restrained as certain phases are excluded from the automated flow.
(3) Isolating these phases does not reflect the industry practice and, therefore, discourages the adaptation of such tools by the industry.

In this section, we quantitatively study the importance and potential of the layout-aware analog sizing framework. We combine the state-of-the-art analog sizing algorithm, DNN-Opt [5], and the general-purpose analog layout generator, MAGICAL [6, 7, 33]. Leveraging the joint framework, the analog sizer can obtain accurate post-layout circuit performance with the requirement of manually scripted layout templates. We conduct a case study on a representative analog circuit and evaluate the joint framework.

4.1 A Case Study

In order to demonstrate the importance of layout effects on the final performance, we perform experiments on a Miller OTA circuit designed in 40nm technology (Fig. 4). The transistors of the circuit are parameterized so that an automation flow to optimize its performance can be conducted. The optimization problem has 17 independent design variables to size length, width, and number of fingers of the transistors.

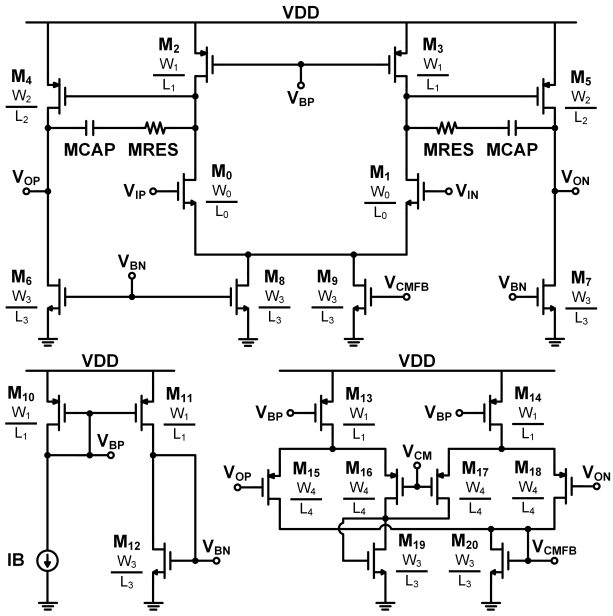


Figure 4: Schematic of the Miller OTA

The optimization problem for the Miller OTA consists of an objective and several performance constraints and is expressed as below:

$$
\begin{aligned}
&\text{minimize Power}\\
&\text{s.t.}\quad \text{DC Gain} > 45\ \text{dB} \qquad \text{Settling Time} < 100\ \text{ns}\\
&\qquad \text{CMRR} > 55\ \text{dB} \qquad \text{Saturation Margins} > 50\ \text{mV}\\
&\qquad \text{PSRR} > 55\ \text{dB} \qquad \text{Unity Gain BW.} > 40\ \text{MHz}\\
&\qquad \text{Out. Swing} > 1\ \text{V} \qquad \text{RMS Noise} < 400\ \text{uV}_{\text{rms}}\\
&\qquad \text{Static error} < \%2 \qquad \text{Phase Margin} > 60\ \text{deg.}
\end{aligned}
\tag{3}
$$

Our experimental study includes the following two steps:

(1) We first use the sizing automation tool DNN-Opt to optimize the performance metrics. This step is based-on schematic-level electrical simulations and is layout agnostic. We then auto-generate the layout of the optimized design using MAGICAL and run post-layout simulations to obtain performance values, including the layout effects.
(2) We modify DNN-Opt algorithm to optimize the post-layout performance of the circuit. To facilitate this approach, we utilize MAGICAL as the layout generator.

4.2 Experiments with Layout Agnostic Loop

In this part of the study, we investigate how much performance degradation is introduced to the layout agnostic flow, i.e., the amount of suffering from completely delayed layout considerations during sizing. To do a fair analysis, we do not use the designer's after-layout intent to formulate the problem, but we refine it by using the designer's schematic-level performance. This way, we match the designer's overdesign effort during sizing.

Table 1: DNN-Opt (Schematic) vs. Designer Performance Comparison Based on Pre-Layout Performance

Schematic Optimized	Intent	DNN-Opt	Designer
Power (mW)	minimize	**0.51**	0.53
Output Swing (V)	≥ 1	**0.99***	0.92
Gain (dB)	≥ 46	**48.1**	46.7
CMRR (dB)	≥ 55	**66.1**	56.2
PSRR (dB)	≥ 55	**63.7**	55.8
Phase Margin (deg)	≥ 57	**62.1**	57.2
RMS Noise (uV)	≤ 400	**380**	390
Rise Time (ns)	≤ 50	**21.3**	22.2
Static Error (%)	≤ 1.2	**1.08**	1.19
UGB (MHz)	≥ 85	**85.9**	85.0

We compile the experiment results in Table 1. We included the new objective and constraint values used for the schematic-level optimization. DNN-Opt was able to provide a design that is very close to satisfying all design intent except for a minor miss for output swing. We further observe that the schematic-level result found by DNN-Opt overperforms designer-crafted design in every metric.

After obtaining optimized results on schematic-level simulations, we use MAGICAL to create layouts for both DNN-Opt and designer solutions. Layout generation is followed by parasitic extraction and post-layout simulations to obtain the post-layout performance of Miller OTA testcase. The post-layout simulations results of DNN-Opt optimized design and designer-created design are collected at Table 2.

Table 2: DNN-Opt (Schematic) vs. Designer Performance Comparison Based on Post-Layout Performance

Schematic Opt + Layout	Intent	DNN-Opt	Designer
Power (mW)	minimize	0.53	0.53
Output Swing (V)	≥ 1	0.96*	**0.97***
Gain (dB)	≥ 45	17.9*	**47.8**
CMRR (dB)	≥ 55	25.6*	**53.7***
PSRR (dB)	≥ 55	25.7*	**55.6**
Phase Margin (deg)	≥ 60	**75.1**	69.9
RMS Noise (uV)	≤ 400	370	370
Rise Time (ns)	≤ 100	**23.0**	110*
Static Error	≤ 2	**1.07**	2.52*
UGB (MHz)	≥ 40	41.6	**42.0**

We observe that, except for the phase margin and rms noise, all other performance metrics degrade in value. The metrics such as the Gain, CMRR, and PSRR are severely reduced for DNN-Opt generated design after layout effects. Together with output swing, these metrics can not satisfy the design intent after layout. On the other hand, the designer's design demonstrates much better resilience against the layout effects. Its Gain and PSRR values meet the constraint threshold, and CMRR is very close to meeting the threshold. Overall, our analysis shows that the performance values of a circuit deviate significantly once a layout is generated. Therefore, the layout effects must be included in the parameter optimization phase in order to generate a robust sizing solution.

4.3 Layout in the Loop Automation Flow

To include the post-layout effects on the performance during sizing, we tailor the classical sizing flow. Instead of optimizing the design variables based on the schematic level simulations, we utilize the layout automation tool MAGICAL to modify performance evaluation steps. The suggested flow is shown in Fig 5. To obtain the post-layout performance of each new design, first, an automated layout is generated via MAGICAL. This step is followed by parasitic extraction, and circuit simulations are run on the updated netlist with parasitic elements.

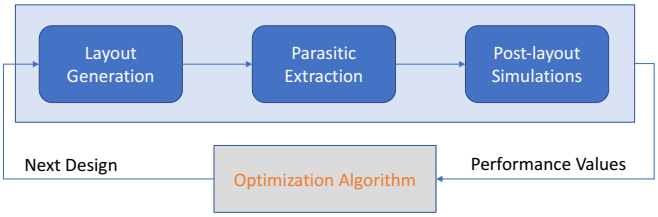


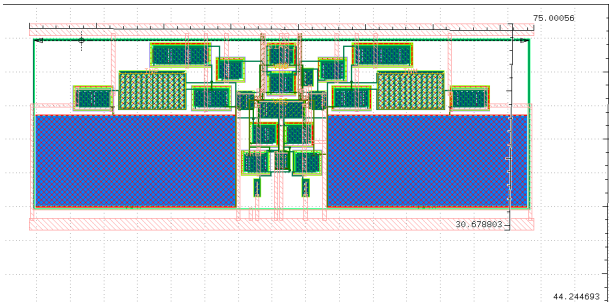
Figure 5: Post-Layout Performance Based Optimization

We evaluate the effectiveness of the layout-in-the-loop optimization flow by revisiting the Miller OTA case. The same optimization algorithm DNN-Opt is used to optimize the same set of parameters using the post-layout performance values instead of the schematic (pre-layout) simulations. The optimized post-layout performance values and their comparison to designer performance are included in Table 3.

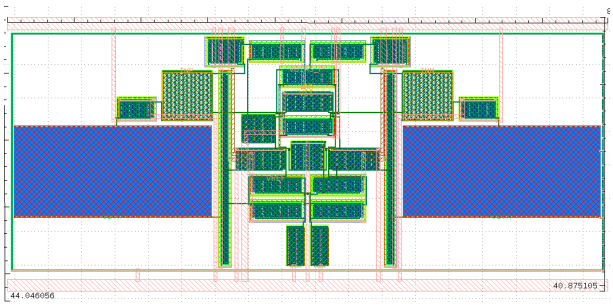
Table 3: DNN-Opt (Post-Layout) vs. Designer Performance Comparison on Post-Layout Performance

Post-Layout Optimized	Intent	DNN-Opt	Designer
Power (mW)	minimize	**0.39**	0.53
Output Swing (V)	≥ 1	**1.11**	0.97*
Gain (dB)	≥ 45	46.1	**47.8**
CMRR (dB)	≥ 55	**56.7**	53.7*
PSRR (dB)	≥ 55	**58.9**	55.6
Phase Margin (deg)	≥ 60	**70.7**	69.9
RMS Noise (uV)	≤ 400	370	370
Rise Time (ns)	≤ 100	**26.9**	110*
Static Error	≤ 2	**1.2**	2.52*
UGB (MHz)	≥ 40	31.3*	**42.0**

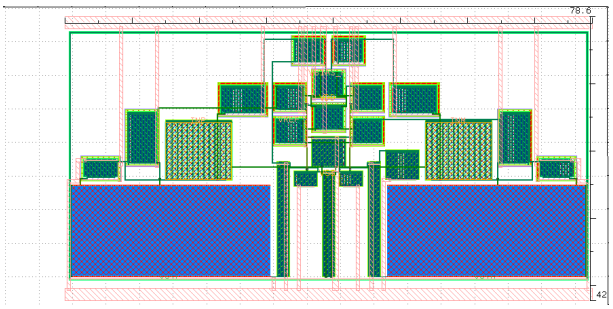
The tool-generated design satisfies all constraints except the unity gain bandwidth (UGB). Considering that the designer's design fails to meet four metrics proves that the proposed flow is very effective. Further, layout-in-the-loop sized solution overperforms the designer's solution in seven metrics and falls behind only in two metrics (Gain, UGB).



(a) Designer optimized design's layout



(b) Post-layout performance-based optimized design's layout



(c) Schematic-level performance-based optimized design's layout

Figure 6: Layouts for different design flows

We further elaborate our analysis by including the layouts generated by MAGICAL in Figure 6.

Analysis for Figure 6a: It is the most compact layout among all generated layouts. The designer has used their experience to provide a layout-friendly design. The total area is around $75\mu m \times 30\mu m$.

Analysis for Figure 6b: Despite the good performance, the layout has some issues in terms of compactness and area consumption compared to the designer's layout. Compared to the designer's layout, the layout-in-the-loop DNN-Opt generated result consumes around 60% larger area as its total area is $90\mu m \times 40\mu m$. Additionally, the lack of dummy devices for critical devices increases the risk of performance degradation in the actual silicon due to fabrication-related non-linearities and issues that are not modeled in the simulation. To improve the final design quality, the flow can be optimized to decrease area consumption and automatically generate dummy devices to increase reliability.

Analysis for Figure 6c: The area consumed by the schematic-level performance-optimized design is around $79\mu m \times 42\mu m$ and 47% larger than designer layout. Also, it resulted in some key devices with abnormal sizes, such as long channel lengths and very short widths. This can cause deviation in these devices' transconductance and output resistance when post-layout effects are introduced into the simulation, leading to a significant degradation in performance parameters such as gain, CMRR, and PSRR. The abnormal-sized devices may also increase parasitic capacitance and resistance, further degrading the UGB (Unity Gain Bandwidth).

5 CHALLENGES AND OPPORTUNITIES

Our quantitative case study demonstrates the importance of considering layout effects in analog sizing. The need for efficient layout-aware analog sizing calls for further research. In this section, we give our perspectives on the challenges and opportunities in future research in layout-aware analog sizing.

5.1 Challenges

Cost of Computation and Data Generation: The most recent algorithms introduced in analog sizing and layout problems utilize machine learning algorithms. The amount and quality of the data are crucial for ML model training; therefore, the quality of the sizing/layout algorithm strongly depends on the same requisites. The complication of this requirement is that it can introduce a huge time cost since the data generation for EDA problems is generally subject to calling EDA simulations, where a single simulation can take days to run. One way to mitigate this issue is to utilize parallelization facilities in the computing environment. However, the user may be subject to commercial license constraints.

Another related challenge is that analog automation lacks open-sourced benchmarks. This introduces an entry barrier for potential developers and prevents collaborations.

Scalability: The ability to scale to larger designs is a critical property of analog sizing/layout algorithms. Working on smaller designs is advantageous for the sake of agile algorithm development and concept proving; however, the end goal of these algorithms is to help in industrial applications, which are typically larger than academic building blocks. Therefore, at some point, automation

algorithms should demonstrate how to handle designs with a large number of devices, nodes, and design parameters.

Another type of scalability concern is the scalability in time. The time requirement for the initial dataset collection and the algorithm run should scale well for various problems.

Reliability: Reliability is a big concern for analog design. The designs have to maintain certain specs under varying environmental conditions. Therefore, the designs are tested against the process, voltage, and temperature (PVT) variations. For a complete end-to-end flow, it is expected that the automation flow also considers these variations, and the final design is robust against variations.

Adaptability Across Technology Nodes: As the design process for IC design changes rapidly, the automation tools need instruments to synchronize with that change. One current limitation of MAGICAL is that it only supports certain technologies, and transforming it with a developer has a time cost.

Transferability: Transferring the knowledge harvested from one task to another is a known concern in all ML applications. Considering the cost of data generation and time-to-market requirements for analog design, transferring the learning experience can significantly help productivity.

There are multiple ways to offer transferability in analog sizing/layout problems. One is to use the knowledge obtained in one design on a new topology. Another one is to transfer the experience between different technology nodes.

Joint Optimization Metrics: To facilitate a joint optimization of sizing and layout, using the post-layout circuit performance alone may not be sufficient. It is noted in our case study that the layout with imperfections can still result in optimized post-layout performance. Therefore, strong feedback between the layout and design parameter optimization algorithm should be constructed. The sizing algorithm and layout generation should adopt a mechanism to process the previously generated layout and respective performance metrics. This mechanism might require additional metrics that assess the quality of the layout and the final performance together. Quantifying the layout quality is challenging since many non-systematic practices are involved.

5.2 Opportunities and Future Directions

ML for Analog Design: EDA is an important application area for ML as there has been a massive interest in recent years [39, 40]. Also, EDA companies have developed their own AI platforms, such as Cadence Cerebrus [41] and Synopsys DSO.ai [42]. For analog sizing, ML-based modeling reduced the optimization time significantly compared to traditional randomized approaches.

Community and Hardware: Being an important ML application area, there is an excellent opportunity to build a larger community for analog design. Analog design benefits from being a part of a very active ML community. Academy-industry collaboration is stronger, and there is more effort from companies to contribute to the open-source environment by publishing their research, sharing code, and design IPs.

Another significant development is that analog design automation benefits from the advances in hardware technology. It helps in two aspects of analog automation. One is that the algorithms that are computationally expensive in the past are now feasible to run. The second aspect is that the availability of parallel computing resources and faster simulations helps significantly in making recent algorithms scalable and practical.

Towards efficient layout-aware analog sizing: In the near future, we expect some degree of integration between sizing and layout automation. One drawback of integrating design phases is that the need for simulation feedback increases. To tackle this problem, certain blocks in the automation flow can utilize surrogate models to reduce the need for simulation. Such surrogates can either replace the real expensive simulations or can serve as guiding models with lower fidelity.

The flow we presented in our case study lacks in processing the quality of the generated layout. A simple extension is to include the total layout area and device shapes as a part of the optimization problem. Another approach is to avoid running certain simulations by following a hierarchy of simulations and certain decision steps. For example, it may become apparent after the placement that the layout does not satisfy desired qualities and the algorithm can decide not to run the remaining costly steps: routing, parasitic extraction, and post-layout simulations.

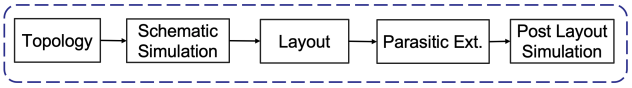


Figure 7: Design phases to obtain post-layout performance

Towards joint analog synthesis: A longer-term future direction for analog automation is to adapt such flows that integrate topology selection, device sizing, and layout generation. As shown in Fig 7, the joint analog synthesis requires several blocks to be automated. The end-to-end flow can output an optimized design based on system-level specifications.

One bottleneck for this flow is that the topology search research is not as mature as sizing and layout automation. There have been tree-based and machine-learning approaches for topology search problems [43, 44], but generalizable methods are still awaited.

6 CONCLUSION

In this paper, we analyze the importance of layout-induced effects on analog device sizing. We reviewed recent research efforts on analog sizing and analog layout automation. Then, we provide a case study using the Miller OTA circuit to quantify the gap between layout-agnostic and layout-in-the-loop sizing approaches. We further discuss the challenges and opportunities in analog design automation.

ACKNOWLEDGEMENT

This work is supported in part by the DARPA IDEA program, NSF under grants 1704758 and 2112665, SRC Task 3160, and Synopsys.

REFERENCES

[1] A. F. Budak, S. Zhang, M. Liu, W. Shi, K. Zhu, and D. Z. Pan, "Machine learning for analog circuit sizing," in *Machine Learning for Analog Circuit Sizing*, H. Ren and J. Hu, Eds. Springer, 2022, pp. 307–335.

[2] H. Chen, M. Liu, X. Tang, K. Zhu, N. Sun, and D. Z. Pan, "Challenges and opportunities toward fully automated analog layout design," *Journal of Semiconductors*, vol. 41, no. 20070021, p. 111407, 2020.

[3] K. Hakhamaneshi, N. Werblun, P. Abbeel, and V. Stojanović, "BagNet: Berkeley analog generator with layout optimizer boosted with deep neural networks," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019.

[4] 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 *IEEE/ACM Proceedings Design, Automation and Test in Eurpoe (DATE)*, 2021.

[5] 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 *ACM/IEEE Design Automation Conference (DAC)*, 2021.

[6] B. Xu, K. Zhu, M. Liu, Y. Lin, S. Li, X. Tang, N. Sun, and D. Z. Pan, "MAGICAL: Toward fully automated analog IC layout leveraging human and machine intelligence," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019.

[7] H. Chen, M. Liu, X. Tang, K. Zhu, A. Mukherjee, N. Sun, and D. Z. Pan, "MAGICAL 1.0: An open-source fully-automated ams layout synthesis framework verified with a 40-nm 1 GS/s ΔΣ ADC," in *IEEE Custom Integrated Circuits Conference (CICC)*, 2021.

[8] N. Horta, "Analogue and mixed-signal systems topologies exploration using symbolic methods," *Analog Integr. Circuits Signal Process.*, p. 161–176, 2002.

[9] "Iprail—intellectual property reuse-based analog ic layout automation," *Integration*, 2003, analog and Mixed-signal IC Design and Design Methodologies.

[10] W. Daems, G. Gielen, and W. Sansen, "Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2003.

[11] M. d. Hershenson, S. P. Boyd, and T. H. Lee, "Optimal design of a cmos op-amp via geometric programming," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2001.

[12] Y. Wang, M. Orshansky, and C. Caramanis, "Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization," in *ACM/IEEE Design Automation Conference (DAC)*, 2014.

[13] A. Vural *et al.*, "Analog circuit sizing via swarm intelligence," *AEU - International Journal of Electronics and Communications*, 2012.

[14] B. Liu, G. Gielen, and F. V. Fernndez, *Automated Design of Analog and High-frequency Circuits: A Computational Intelligence Approach.* Springer, 2013.

[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[16] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Conference on Neural Information Processing Systems (NIPS)*, vol. 25, 2012.

[17] 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 (TCAD)*, Feb 2014.

[18] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Multi-objective bayesian optimization for analog/rf circuit synthesis," in *ACM/IEEE Design Automation Conference (DAC)*, 2018.

[19] G. İslamoğlu, T. O. Çakici, E. Afacan, and G. Dündar, "Artificial neural network assisted analog ic sizing tool," in *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, 2019.

[20] 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*, 2020.

[21] A. Budak, M. Gandara, W. Shi, D. 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 (TCAD)*, 2021.

[22] N. Lourenço, J. Rosa, R. Martins, H. Aidos, A. Canelas, R. Póvoa, and N. Horta, "On the exploration of promising analog ic designs via artificial neural networks," in *2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, 2018.

[23] H. Wang, K. Wang, J. Yang, L. Shen, N. Sun, H. Lee, and S. Han, "GCN-RL circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning," *ACM/IEEE Design Automation Conference (DAC)*, 2020.

[24] K. Settaluri, A. Haj-Ali, Q. Huang, K. Hakhamaneshi, and B. Nikolić, "AutoCkt: deep reinforcement learning of analog circuit designs," *IEEE/ACM Proceedings Design, Automation and Test in Eurpoe (DATE)*, 2020.

[25] Y. Wang, P. D. Franzon, D. Smart, and B. Swahn, "Multi-fidelity surrogate-based optimization for electromagnetic simulation acceleration," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, aug 2020.

[26] K. Yang *et al.*, "Trust-region method with deep reinforcement learning in analog design space exploration," in *ACM/IEEE Design Automation Conference (DAC)*, 2021.

[27] A. F. Budak, D. Smart, B. Swahn, and D. Z. Pan, "APOSTLE: asynchronously parallel optimization for sizing analog transistors using dnn learning," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2023.

[28] J. Crossley, A. Puggelli, H.-P. Le, B. Yang, R. Nancollas, K. Jung, L. Kong, N. Narevsky, Y. Lu, N. Sutardja, E. J. An, A. L. Sangiovanni-Vincentelli, and E. Alon, "BAG: A designer-oriented integrated framework for the development of ams circuit generators," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2013.

[29] K. Kunal, M. Madhusudan, A. K. Sharma, W. Xu, S. M. Burns, R. Harjani, J. Hu, D. A. Kirkpatrick, and S. S. Sapatnekar, "ALIGN: Open-source analog layout automation from the ground up," in *ACM/IEEE Design Automation Conference (DAC)*, 2019.

[30] M. Liu, W. Li, K. Zhu, B. Xu, Y. Lin, L. Shen, X. Tang, N. Sun, and D. Z. Pan, "S^3DET: Detecting system symmetry constraints for analog circuits with graph similarity," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2020.

[31] K. Zhu, H. Chen, M. Liu, X. Tang, N. Sun, and D. Z. Pan, "Effective analog/mixed-signal circuit placement considering system signal flow," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2020.

[32] H. Chen, K. Zhu, M. Liu, X. Tang, N. Sun, and D. Z. Pan, "Toward silicon-proven detailed routing for analog and mixed signal circuit," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2020.

[33] K. Zhu, H. Chen, M. Liu, and D. Z. Pan, "Tutorial and perspectives on MAGICAL: A silicon-proven open-source analog IC layout system," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2022.

[34] K. Settaluri, Z. Liu, R. Khurana, A. Mirhaj, R. Jain, and B. Nikolic, "Automated design of analog circuits using reinforcement learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2022.

[35] H. Ren, G. F. Kokai, W. J. Turner, and T.-S. Ku, "ParaGraph: Layout parasitics and device parameter prediction using graph neural networks," in *ACM/IEEE Design Automation Conference (DAC)*, July 2020.

[36] B. Shook, P. Bhansali, C. Kashyap, C. Amin, and S. Joshi, "MLParest: Machine leaning based parasitic estimation for custom circuit design," in *ACM/IEEE Design Automation Conference (DAC)*, July 2020.

[37] K. Zhu, H. Chen, W. Turner, G. Kokai, P.-H. Wei, D. Z. Pan, and H. Ren, "TAG: Learning circuit spatial embedding from layouts," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2022.

[38] M. Liu, K. Zhu, J. Gu, L. Shen, X. Tang, N. Sun, and D. Z. Pan, "Towards decrypting the art of analog layout: Placement quality prediction via transfer learning," in *IEEE/ACM Proceedings Design, Automation and Test in Eurpoe (DATE)*, 2020.

[39] H. Ren, B. Khailany, M. Fojtik, and Y. Zhang, "Machine learning and algorithms: Let us team up for eda," *IEEE Design Test*, vol. 40, 2023.

[40] A. F. Budak, Z. Jiang, K. Zhu, A. Mirhoseini, A. Goldie, and D. Z. Pan, "Reinforcement learning for electronic design automation: Case studies and perspectives: (invited paper)," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2022.

[41] N. Hanchate, "Improving chip design performance and productivity using machine learning," in *Proceedings of the 2022 International Symposium on Physical Design*, ser. ISPD '22.

[42] "Synopsys dso.ai," https://www.synopsys.com/implementation-and-signoff/ml-ai-design/dso-ai.html.

[43] H. Y. Koh, C. H. Sequin, and P. R. Gray, "Opasyn: a compiler for cmos operational amplifiers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1990.

[44] M. Rotman and L. Wolf, "Electric analog circuit design with hypernetworks and a differential simulator," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.