Exploring a Machine Learning Approach to Performance Driven Analog IC Placement

Yaguang Li*, Yishuang Lin†, Meghna Madhusudan‡, Arvind Sharma‡, Wenbin Xu*, Sachin Sapatnekar‡, Ramesh Harjani‡, Jiang Hu*†
*Dept. of Electrical and Computer Engineering, Texas A&M University
†Dept. of Computer Science and Engineering, Texas A&M University
‡Dept. of Electrical and Computer Engineering, University of Minnesota

***Abstract*—Analog IC layout is usually a time-consuming manual design process. Although automated analog IC layout has been studied for decades, most of the previous works are focused on geometric constraints. As a result, there is often a performance gap compared to manual designs, which prevents the automated tools from wide applications. The recent progress on machine learning technology offers an opportunity for solving this problem. In this work, several machine learning techniques are investigated for analog IC performance prediction, which is further applied for performance driven placement. Simulation results from several amplifier designs indicate that the proposed approach can achieve performance similar to manual layout but is orders of magnitude faster.**

I. Introduction

Analog IC (Integrated Circuit) layout is usually a manual design process. This is in sharp contrast to its digital counterpart, which is highly automated. This difference is due to several intertwined factors. First, the impact of layout on analog circuit performance can be very large due to significant RC (Resistance and Capacitance) parasitic generated in layout. In Table I, a few performance characteristics from schematic (pre-layout) and post-layout simulation of an OTA (Operational Transconductance Amplifier) design are compared. One can see that layout can cause as much as 22% loss of unity gain frequency. Second, the layout effect on performance is often very complex and thus difficult to be quickly and accurately estimated. Last but not the least, analog circuit performance has much larger variety than digital circuits, and different performance metrics are used for different types of analog circuits. For example, operational amplifiers are focused on gain, bandwidth and phase margin, ADC/DAC (Analog to Digital/ Digital to Analog Converter) pays attention to linearity, and settling time is evaluated for PLL (Phase Locked Loop) circuits. Overall, analog IC layout requires experienced designers to perform multiple iterations of trial layout and simulation evaluations. Consequently, it is very time consuming and forms a design bottleneck.

Characteristic	Schematic	Layout	Change
DC Gain (dB)	39.30	37.25	-5%
Bandwidth (MHz)	10.64	10.47	-2%
Unity Gain Frequency (MHz)	440	383	-22%

TABLE I: Schematic and post-layout performance of OTA (Operational Transconductance Amplifier).

Automated analog IC layout has been studied for decades [1]. However, most of the previous works are focused on special geometric constraints for analog ICs, e.g., symmetry and common centroid [2]–[7]. Although these constraints are fundamental for dealing with performance deviations in presence of variations, they are far from being adequate for achieving desired nominal performance. Performance is considered in analog IC layout by transforming performance constraints to geometric constraints in [8]. However, the interactions among multiple performance characteristics are simplified such that the geometric constraints can be unnecessarily tight and make it very difficult to find a feasible solution. The work of [9] directly optimizes circuit performance in analog IC placement. However, the performance is estimated by a simple linear model that is hard to be generalized to many nonlinear behaviors in analog circuits. Special performance related constraints are identified and enforced in [10], [11] for analog IC placement. However, these special constraints are not for general cases. A recent work [12] applies neural network for layout migration of different sizing solutions with little attention on circuit performance. Overall, automated performance driven analog IC layout techniques are far from being well studied or mature to wide applications.

In this work, we study a machine learning approach to automated performance driven analog IC placement. Since wire parasitic is largely decided by the distance among pins, placement generates the first order effect on circuit performance. Different from constraint-based methods [8], [10], [11], our approach directly optimizes circuit performance during placement. Moreover, machine learning-based performance models can handle nonlinear behaviours and therefore is superior to the linear model used in [9]. Different machine learning techniques are investigated, including SVM (Support Vector Machine), neural network and random forest.

Our work makes the following contributions.

- Machine learning based circuit performance prediction techniques are developed.
  - Feature selection and data preparation techniques are proposed to obtain a large amount training data with almost no layout designs.
  - Machine learning model fidelity is studied in addition to its accuracy and precision. There has been very few, if not none, study on machine learning model fidelity in VLSI circuit designs.

- A performance driven analog IC placement methodology is proposed. It can directly optimize analog circuit performance along with conventional objectives, including wirelength, area and geometric constraints such as symmetry constraints.
- The proposed approach is applied on the placement for three different kinds of OTA designs. The post-layout simulation results show that the proposed approach achieves significantly better performance than conventional automatic placement. It also reaches performance similar to manual layout but orders of magnitude faster.

The rest of this paper is organized as follows. Related previous works are briefly reviewed in Section II. The problem formulation and an overview of our approach is provided in Section III. The proposed machine learning techniques for circuit performance prediction are introduced in Section IV. The performance driven placement method is described in Section V. Experimental results are shown in Section VI. The conclusion is provided in Section VII.

II. Related Previous Work

The study of automated analog IC layout is early as 30 years ago [1]. Analog IC placement methods [2]–[5] have mostly followed the simulated annealing framework in floorplanning of digital designs, except that additional geometric constraints, such as symmetry and common centroid, are considered. The geometric constraints are often enforced through sequence pair [2], [3], [5] or B-tree representation [4]. Recently, analytical approaches are explored in [6], [7] without directly addressing analog circuit performance.

The layout impact on performance is noticed in [8], where performance constraints are transformed into geometric constraints for layout. Such transformation is not sophisticated enough to well capture the complicated relationship between layout and performance, and thus it faces the dilemma that the obtained geometric constraints are either overly tight or insufficient for satisfying performance specifications. A directly performance driven layout approach is proposed in [9]. However, the performance is estimated by linear approximation, which is not effective for nonlinear characteristics. In [10] and [11], analog IC performance is addressed by the constraints of monotonic current paths. However, the effectiveness of this technique is restricted to certain performance characteristics instead of being general for different metrics.

Recently, machine learning techniques are explored for analog IC designs. In [12], a neural network technique is developed for layout migration without considering performance. Knowledge mining is applied to reuse legacy design patterns [13], but again without directly considering performance. Reinforcement learning-based analog parameter tuning is proposed in [14]. Graph neural network techniques are applied to generate templates for RF passives [15].

Overall, the study on performance driven placement for analog ICs has been limited and far from sufficient.

III. Problem Formulation and Overview of the Approach

Given an analog circuit netlist (or schematic description) and process technology file, the placement is to determine the locations of all transistors and passive devices in the circuit such that design rules and geometric constraints, such as symmetry, are satisfied and a composite objective function, in terms of area, wirelength and circuit performance, is minimized. Our placement algorithm is based on simulated annealing, which has been widely used in previous analog placement works [5]. A main advantage of simulated annealing is its flexibility in incorporating different types of cost functions. Although it is not a fast algorithm in general, its runtime speed is acceptable for analog IC designs, which typically have much less elements than digital circuits. A key difference from previous works is that circuit performance is captured in the cost function through a machine learning model. That is, given a placement solution, the machine learning model predicts its performance. Performance driven analog placement is a huge challenge, and our approach is focused on solving a type of analog circuits, e.g., amplifiers, as a step toward overcoming the entire challenge.

IV. Machine Learning-Based Circuit Performance Prediction

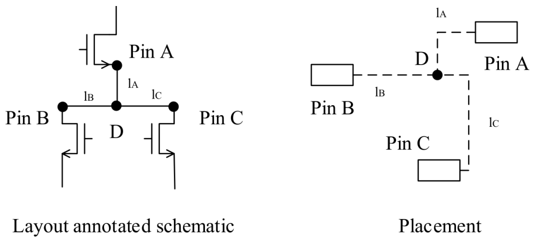
There are large variety of different machine learning techniques. In this work, three popular techniques are investigated: Neural Network (NN) [16], Random Forest (RF) [17] and Support Vector Machine (SVM) [18].

A. Machine Learning Feature Engineering

A placement solution $(\mathbf{x}, \mathbf{y}) = ((x_1, y_1), (x_2, y_2), ...)$ specifies locations for all circuit elements. A vector of circuit characteristics $\mathbf{z} = (z_1, z_2, ...)$ depend on $(\mathbf{x}, \mathbf{y})$. For example, the characteristics can be gain, phase margin, linearity, settling time, etc. To predict $\mathbf{z}$ through a machine learning model $\mathcal{M}$, the input features for $\mathcal{M}$ can be naïvely chosen as $(\mathbf{x}, \mathbf{y})$. However, such features require to generate a layout solution for each training sample, where the post-routing $\mathbf{z}$ values are used as labels. Generating a large number of layout solutions and performing parasitic extractions would cost a large computation runtime.

We propose an alternative approach for the model features such that almost no layout solution is needed for obtaining training data, and therefore data preparation time is greatly reduced. An observation is that the dependence of $\mathbf{z}$ on $(\mathbf{x}, \mathbf{y})$ is through post-routing wirelength $\mathbf{l} = (l_1, l_2, ...)$ among placed circuit elements, which decide RC parasitic. We suggest to approximate $\mathbf{l}$ with a pre-routing estimate $\tilde{\mathbf{l}} = (\tilde{l}_1, \tilde{l}_2, ...)$, which serve as machine learning model features. As such, no layout needs to be performed in preparing training data. To simplify the description, we assume that all wires have the same width and thus RC parasitic is determined by wirelength. This assumption is often true when the routing is mostly performed at lower metal layers.

For a 2-pin net, the pre-routing wirelength estimate is simply the Manhattan distance between the two pins. This is a lower



Layout annotated schematic

Placement

Fig. 1: Star model for multi-pin nets.

bound to post-routing wirelength and often the actual one as analog circuits are generally not congested and wire detour does not happen very often. The situation for multi-pin nets is more complex as it depends on Steiner tree constructions in routing stage. We propose to use the star model for the pre-routing estimate. This is illustrated by the 3-pin net in Figure 1. A star has a center, which is chosen to have the medium coordinates among all pins in the net. For a 3-pin net, such center is actually the Steiner point location of the optimal Steiner minimum tree, as shown by point D on right of Figure 1. In the star model, every pin has a direct connection with the center. Therefore, an m-pin net consists of m wire segments and corresponding lengths. The wirelengths of all wire segments $\tilde{\mathbf{l}} = (\tilde{l}_1, \tilde{l}_2, ...)$ form the input features to a machine learning model.

B. Training Data Preparation

The star model allows us to generate many training data samples using only one layout solution. Given an arbitrary legal placement of a circuit, routing is performed to obtain a complete layout and then parasitic extraction is conducted. Then, the wire parasitics of the extracted netlist are removed to form a template where transistor parasitics are retained. Next, a random star model is inserted at each net of the template to obtain one training sample. For example, one training sample can be obtained by assigning random values to l_A, l_B and l_C to the schematic in Figure 1, which is the template. Last, the layout annotated schematic is simulated to obtain label values for $\mathbf{z}$. By iterating with different random wirelength in the star models, many different training samples can be obtained. In this process, layout is performed only once.

C. Machine Learning Model Output

Given input feature $\tilde{\mathbf{l}} = (\tilde{l}_1, \tilde{l}_2, ...)$, a machine learning model $\mathcal{M}$ attempts to predict performance $\mathbf{z} = (z_1, z_2, ...)$. In general, the prediction can be carried in several different ways:

- **Regression**. This is to directly predict numerical values of $\mathbf{z} = (z_1, z_2, ...)$. According to our experience, it is quite challenging to obtain accurate regression.
- **Classification**. This is to classify each characteristic z_i into a certain class, e.g., to classify whether or not $z_i \geq \theta_i$, where θ_i is a given threshold or specification. For example, if z_i is the gain for an amplifier design and $\theta_i = 40dB$, the classification predicts if the gain is at least $40dB$. Such classification cannot differentiate between clear cases, where the model has a large chance to be correct, and borderline cases, where the model is more error prone.
- **Logistic regression**. This is to estimate the probability $P(z_i \geq \theta_i)$, $i = 1, 2, ...$ that performance specification θ_i is satisfied. Thus, logistic regression can be treated as soft classification and avoid the aforementioned drawback of classification.

We choose to use the machine learning models in the way of logistic regression. For a neural network, there is a nonlinear sigmoid function that maps the output into probability. Typically, people use a step function to classify a positive sample if the probability is greater than a threshold. Here, we directly use the probability. For random forest, the classification probability of each sample is calculated according to classification results from all the trees in the random forest. For SVM, the probability is calibrated using Platt scaling, which is logistic regression on the SVM's scores.

D. Machine Learning Model Fidelity

A machine learning model is usually evaluated by its accuracy, precision, etc. Since the models here are applied to guide placement optimization, what matters is actually the model fidelity [19]. Roughly speaking, fidelity tells the consistency between a model and the golden reference, and indicates if the model can guide optimization moves toward correct directions. For example, there are two placement solutions p and q, with post-routing gains of $g_p = 40dB$ and $g_q = 50dB$, respectively. Model $\mathcal{M}_1$ predicts $g_{p,1} = 25dB$ and $g_{q,1} = 30dB$ while model $\mathcal{M}_2$ predicts $g_{p,2} = 46dB$ and $g_{q,2} = 44dB$. Although model $\mathcal{M}_1$ has greater errors than model $\mathcal{M}_2$, it has higher fidelity as it correctly predicts that the gain of p is smaller than q. If the optimization is to find a solution with the maximum gain, model $\mathcal{M}_2$ would incorrectly point to picking solution p even though it has smaller errors.

Given two solutions p and q, if the logistic regression results of model $\mathcal{M}$ on circuit characteristic z are $P(\tilde{z}_p \geq \theta)$ and $P(\tilde{z}_q \geq \theta)$, we say model $\mathcal{M}$ is **consistent** with golden reference $\hat{z}_p$ and $\hat{z}_q$ if and only if $P(\tilde{z}_p \geq \theta) > P(\tilde{z}_q \geq \theta) \Leftrightarrow \hat{z}_p \geq \hat{z}_q$. The percentage among N distinct solution pairs for which model $\mathcal{M}$ is consistent with the golden reference is a quantitative metric of **fidelity**. The ideal fidelity is 100%.

V. Performance Driven Placement Guided by Machine Learning

The placement algorithm follows the framework of simulated annealing like many previous works [5], [9]. The relative spatial order among circuit elements is described by sequence pairs [2], [5], which is friendly to enforcing many geometric constraints, such as symmetry and common centroid. The key difference from the previous work is that circuit performance is explicitly considered in the cost function of simulated annealing.

Usually, the performance of a circuit is evaluated by a set of characteristics $z_1, z_2, ...$, which can be gain, phase margin, etc. For each of the characteristics z_i, a *satisfaction function* is defined as

$$\psi_i(z_i) = \begin{cases} 1 & z_i \geq \theta_i \\ \frac{z_i}{\theta_i} & z_i < \theta_i \end{cases} \tag{1}$$

where θ_i is the design specification for z_i. If there are K characteristics to be considered, the performance **FOM** (Figure of Merit) is defined as

$$\Phi = \sum_{i=1}^{K} w_i \cdot \psi_i(z_i) \tag{2}$$

where w_i represents weight factors and $\sum_{i=1}^{K} w_i = 1$. If all performance specifications are satisfied, $\Phi = 1$.

The machine learning model estimates the probability that a specification is satisfied instead of directly predicting the value of each characteristic. Therefore, we define a **POD** (Probability of Demerit) based on the machine learning model inference results:

$$\Delta = \sum_{i=1}^{K} w_i \cdot P(z_i < \theta_i) \tag{3}$$

where P indicates the probability of violating specifications. Evidently, POD needs to be minimized. POD can be directly evaluated by the machine learning models, is easy to be incorporated in simulated annealing and captures the same intention as FOM in principle.

The cost function to be minimized in simulated annealing is

$$\alpha \cdot A + \beta \cdot W + \gamma \cdot \Delta \tag{4}$$

where A is normalized total area, W is normalized total wirelength estimated according to HPWL (Half Perimeter Wirelength), and α, β and γ are weighting factors that sum to 1. In addition to minimizing the cost function, our simulated annealing algorithm enforces geometric constraints, such as symmetry and common centroid, like in [5].

VI. Experiment Results

The experiments are conducted on a Linux machine using Xeon (R) E5-2680 V2 processor with 2.8GHz frequency and 256G memory. The machine learning models are implemented in Python. Our analog IC placer and a router based on [20] were programmed in C++. SPICE simulations are performed on extracted netlists after after layout and design rule checking. The ASAP 7nm process technology [21] is employed in the experiment.

The proposed techniques are evaluated on three different OTA designs: 5-transistor OTA, cascode OTA and current mirror OTA. For each design, 1000 layout annotated schematics are generated by randomly varying wirelength for each net and then simulated to obtain training data. Among each 1000 samples, 600 samples are employed for training and the others are used for testing. During the model construction, cross validation is performed to tune model hyperparameters.

A. Evaluation of Machine Learning Models

The neural network structure after tuning has three hidden layers and the number of nodes in each layer is equal to input feature size. Each random forest in the experiment has no more than 500 trees, each of which has height no greater than 10. For SVM, radial basis function kernel is employed and the regularization parameter is set to be 1.

	TPR	FPR	Accuracy	Precision	F1 Score
NN	87.1%	9.72%	88.7%	90.3%	88.7%
RF	76.3%	24.4%	76.2%	76.4%	76.3%
SVM	77.3%	20.8%	78.4%	79.5%	78.2%

TABLE II: Average model performance among all four characteristics of all three OTA designs.

The evaluation of a machine learning model usually uses the following terms:

- True Positive (TP): the number of samples for which the model correctly classifies them to positive class.
- True Negative (TN): the number of samples for which the model correctly classifies them to negative class.
- False Positive (FP): the number of samples for which the model falsely classifies them to positive class.
- False Negative (FN): the number of samples for which the model falsely classifies them to negative class.

Then, a machine learning model is often evaluated by the following metrics:

- **TPR** (True Positive Rate), a.k.a. **recall**: $\frac{TP}{TP+FN}$.
- **FPR** (False Positive Rate): $\frac{FP}{FP+TN}$.
- **Accuracy**: $\frac{TP+TN}{TP+TN+FP+FN}$.
- **Precision**: $\frac{TP}{TP+FP}$.
- **F1 score**: $\frac{2 \times Precision \times TPR}{Precision+TPR}$.

The NN, RF and SVM results on these metrics are summarized in Table II. These are the average results among the four circuit performance characteristics and the three OTA designs. In order to apply these metrics, the logistic regression results from the models, which are probabilities, are rounded into binary classifications. One can see that NN has the best result with 87% TPR at about 9.7% FPR.

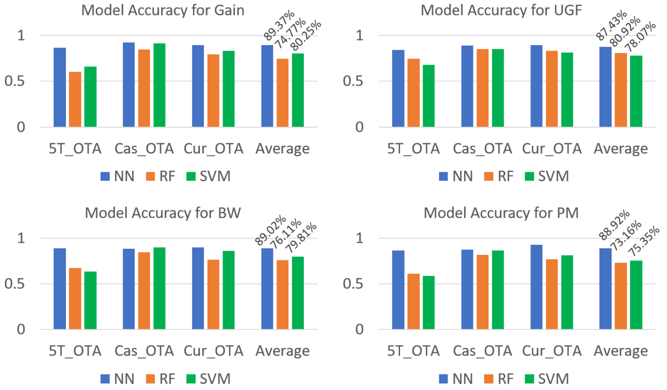


Fig. 2: Machine learning model accuracy on gain, unity gain frequency (UGF), bandwidth (BW) and phase margin (PM).

We further analyze the accuracy of these models on different circuit characteristics of different OTA designs. The results are plotted in Figure 2. NN outperforms both RF and SVM on every case. SVM is better than RF on gain, bandwidth and phase margin.

Table III shows results of machine learning model fidelity, which is defined in Section IV-D, with respect to simulation results on layout annotated schematic, which is the golden reference and illustrated on the left of Figure 1. These results are obtained from 43071 distinct solution pairs. In this case, the machine learning model input features are the same as the

	5-transistor OTA				Cascode OTA				Current Mirror OTA				Average
	Gain	UGF	BW	PM	Gain	UGF	BW	PM	Gain	UGF	BW	PM	
NN	89.7%	99.2%	98.7%	99.0%	98.1%	97.5%	98.7%	97.5%	99.1%	98.7%	99.4%	99.0%	97.9%
RF	92.5%	96.1%	92.8%	91.8%	82.3%	92.0%	86.1%	83.3%	94.9%	96.5%	94.7%	94.7%	91.5%
SVM	90.5%	95.2%	92.1%	89.2%	81.7%	88.8%	84.1%	79.3%	95.7%	96.4%	96.0%	94.8%	90.3%

TABLE III: Machine learning model fidelity with respect to simulation of layout annotated schematics.

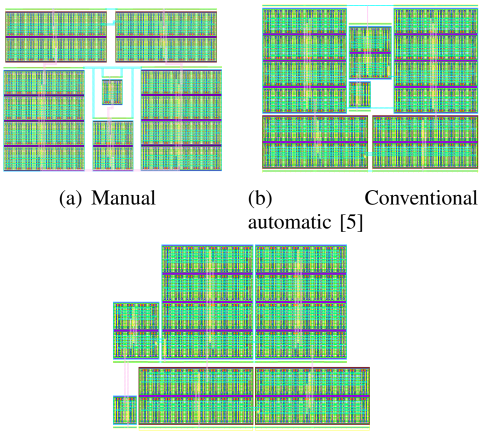
	5-transistor OTA				Cascode OTA				Current Mirror OTA				Average
	Gain	UGF	BW	PM	Gain	UGF	BW	PM	Gain	UGF	BW	PM	
NN	84.8%	82.9%	87.7%	87.9%	88.1%	79.0%	84.3%	80.6%	84.1%	81.6%	85.5%	85.2%	84.3%
RF	86.3%	85.2%	85.1%	81.6%	88.6%	84.9%	90.3%	85.3%	82.3%	81.3%	79.8%	81.8%	84.4%
SVM	81.6%	80.7%	89.1%	89.5%	85.6%	79.5%	87.3%	85.7%	86.5%	80.5%	79.9%	81.5%	83.9%

TABLE IV: Machine learning model fidelity with respect to simulation of post-layout circuits.

star model wirelength (see Figure 1). Hence, inconsistency between model prediction and the golden reference is due to the errors on predicting circuit characteristics. The average fidelity results of all three models are above 90%, which is greater than their accuracy reported in Table II. Like the results in Table II, NN outperforms RF and SVM.

	Schematic	Manual	Conventional Automatic [5]	NN	RF	SVM
Gain (dB)	32.43	32.39	26.5	31.86	31.49	32.15
UGF (MHz)	1105	870.37	656.9	718.9	785.4	721.6
BW (MHz)	26.45	21.0	34.23	18.17	20.68	17.65
PM (degree)	86.47	85.05	85.53	93.84	94.1	94.07
FOM	1.00	0.89	0.76	0.81	0.84	0.82
Area (μm^2)	-	18.72	16.41	19.73	17.97	17.95
Wirelength (μm)	-	12.40	7.36	10.38	8.13	9.16

TABLE V: Results of 5-transistor OTA.



(a) Manual

(b) Conventional automatic [5]

(c) SVM based

Fig. 3: SVM-guided layout of 5-transistor OTA.

The golden reference for the fidelity results in Table IV are obtained from post-layout simulation. In this case, the actual wirelength after layout might be different from the machine learning model input features. Thus, the inconsistency may be attributed to this discrepancy besides errors on predicting circuit characteristics. To certain extent, the models here attempt to predict post-layout wirelength as well. Interestingly, all three models achieve very similar fidelity of about 84%. Such results tell that RF and SVM are better in predicting post-layout wirelength although they are not as good as NN in predicting circuit performance.

	Schematic	Manual	Conventional Automatic [5]	NN	RF	SVM
Gain (dB)	37.0	33.01	23.71	37.0	33.73	26.1
UGF (MHz)	1522.9	1167	947.6	1025	873.7	1266
BW (MHz)	21.82	26.75	56.02	13.0	17.76	58.42
PM (degree)	82.1	80.66	108.5	105.4	95.69	103.3
FOM	1.00	0.85	0.71	0.82	0.77	0.78
Area (μm^2)	-	26.52	24.12	31.68	29.45	27.36
Wirelength (μm)	-	20.69	8.17	13.79	21.73	19.09

TABLE VI: Results of cascode OTA.

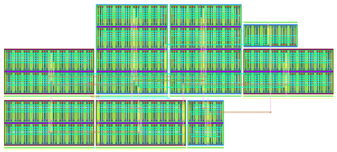


Fig. 4: SVM-guided layout of cascode OTA.

	Schematic	Manual	Conventional Automatic [5]	NN	RF	SVM
Gain (dB)	32.57	32.73	33.05	32.64	33.04	32.87
UGF (MHz)	531.03	542.96	451.0	481.1	499.8	497.4
BW (MHz)	12.5	12.96	10.19	11.42	11.33	11.51
PM (degree)	82.82	79.39	78.48	77.96	77.62	77.14
FOM	1.00	0.99	0.90	0.94	0.95	0.95
Area (μm^2)	-	15.75	14.49	21.47	17.06	20.46
Wirelength (μm)	-	18.40	10.1	19.42	13.36	14.00

TABLE VII: Results of current mirror OTA.

B. *Results of Performance Driven Placement*

The performance driven placement guided by machine learning is evaluated by comparison with manual layout and conventional automatic placement [5]. Individual circuit performance characteristics are obtained from post-layout simulation. The overall performance of a layout solution is evaluated by the FOM defined in Equation (2). The performance result from simulating schematics is also provided as a reference.

Time	Training	5-transistor OTA			Cascode OTA			Current mirror OTA		
		Placement	Routing	Total	Placement	Routing	Total	Placement	Routing	Total
Manual (min)	-	90			110			160		
Conventional automatic [5] (s)	-	0.22	1.04	1.26	0.60	1.67	2.27	0.20	1.18	1.38
NN (s)	27.75	7.18	1.19	8.37	7.40	3.13	10.53	7.93	1.40	9.33
RF (s)	4.03	12.93	1.52	14.45	13.09	2.96	16.05	13.73	1.23	14.96
SVM (s)	3.84	5.14	1.27	6.41	5.37	2.07	7.44	5.82	1.05	6.87

TABLE VIII: Run-time comparison.

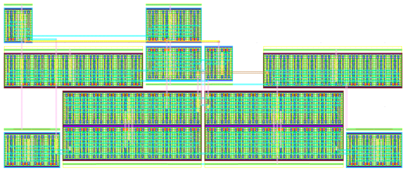


Fig. 5: RF-guided layout of current mirror OTA.

1) Results of 5-Transistor OTA: The main results of the 5-transistor OTA design are summarized in Table V. RF guided placement achieves FOM of 0.84, which is close to the FOM of 0.89 by the manual layout and significantly better than conventional automatic method [5]. At the same time, RF achieves the same area and much shorter post-layout wirelength compared to the manual layout. Please note all symmetry constraints are satisfied in all these layout designs. Sample layout pictures of the 5-transistor OTA design are shown in Figure 3.

2) Results of Cascode OTA: The results of cascode OTA are shown in Table VI and Figure 4. In this case, NN achieves the best FOM of 0.82 among the three machine learning models and this FOM is close to the 0.85 of manual layout. Its area is bigger than manual layout but its post-layout wirelength is much shorter.

3) Results of Current Mirror OTA: The results of current mirror OTA are shown in Table VII and Figure 5. All three models achieve FOM values similar to the manual layout and better than the conventional automatic layout. They tend to result in greater area but shorter wirelength. The similar FOMs from the three models can be attributed to the similar fidelity among them.

4) Runtime Cost: Computation runtime comparison is provided in Table VIII. The machine learning-based layout time is orders of magnitude faster than manual design. For example, the total layout times for SVM-guided placement are 842X, 887X, and 1397X faster than manual layout in the three OTA designs.

VII. Conclusion and Future Research

A new approach to performance driven analog IC placement guided by machine learning is proposed. In this methodology, layout design is largely avoided in preparing machine learning model training data. The model fidelity is analyzed in addition to accuracy. Experimental results show that the proposed approach outperforms conventional automatic method and can obtain circuit performance similar to manual layout but is orders of magnitude faster. In future research, we will study how to apply machine learning across different kinds of analog circuits other than OTAs.

Acknowledgement

This work is supported by the DARPA ERI IDEA program.

References

[1] J. M. Cohn, D. J. Garrod, R. A. Rutenbar, and L. R. Carley, “KOAN/ANAGRAM II: New tools for device-level analog placement and routing,” *IEEE Journal Solid-State Circuits*, vol. 26, no. 3, pp. 330–342, 1991.

[2] F. Balasa and K. Lampaert, “Symmetry within the sequence-pair representation in the context of placement for analog design,” *IEEE TCAD*, vol. 19, no. 7, pp. 721–731, 2000.

[3] E. F. Young, C. C. Chu, and M. Ho, “Placement constraints in floorplan design,” *IEEE TVLSI*, vol. 12, no. 7, pp. 735–745, 2004.

[4] P.-H. Lin, Y.-W. Chang, and S.-C. Lin, “Analog placement based on symmetry-island formulation,” *IEEE TCAD*, vol. 28, no. 6, pp. 791–804, 2009.

[5] Q. Ma, L. Xiao, Y.-C. Tam, and E. F. Young, “Simultaneous handling of symmetry, common centroid, and general placement constraints,” *IEEE TCAD*, vol. 30, no. 1, pp. 85–95, 2010.

[6] B. Xu, S. Li, X. Xu, N. Sun, and D. Z. Pan, “Hierarchical and analytical placement techniques for high-performance analog circuits,” in *Proc. ISPD*, 2017, pp. 55–62.

[7] B. Xu, S. Li, C.-W. Pui, D. Liu, L. Shen, Y. Lin, N. Sun, and D. Z. Pan, “Device layer-aware analytical placement for analog circuits,” in *Proc. ISPD*, 2019, pp. 19–26.

[8] U. Choudhury and A. Sangiovanni-Vincentelli, “Automatic generation of parasitic constraints for performance-constrained physical design of analog circuits,” *IEEE TCAD*, vol. 12, no. 2, pp. 208–224, 1993.

[9] K. Lampaert, G. Gielen, and W. M. Sansen, “A performance-driven placement tool for analog integrated circuits,” *IEEE Journal Solid-State Circuits*, vol. 30, no. 7, pp. 773–780, 1995.

[10] P.-H. Wu, M. P.-H. Lin, Y.-R. Chen, B.-S. Chou, T.-C. Chen, T.-Y. Ho, and B.-D. Liu, “Performance-driven analog placement considering monotonic current paths,” in *Proc. ICCAD*, 2012, pp. 613–619.

[11] C.-W. Lin, J.-M. Lin, C.-P. Huang, and S.-J. Chang, “Performance-driven analog placement considering boundary constraint,” in *Proc. DAC*, 2010, pp. 292–297.

[12] D. Guerra, A. Canelas, R. Póvoa, N. Horta, N. Lourenço, and R. Martins, “Artificial neural networks as an alternative for automatic analog IC placement,” in *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, 2019, pp. 1–4.

[13] P.-H. Wu, M. P.-H. Lin, and T.-Y. Ho, “Analog layout synthesis with knowledge mining,” in *European Conference on Circuit Theory and Design*, 2015, pp. 1–4.

[14] H. Wang, J. Yang, H.-S. Lee, and S. Han, “Learning to design circuits,” *arXiv preprint arXiv:1812.02734*, 2018.

[15] G. Zhang, H. He, and D. Katabi, “Circuit-GNN: Graph neural networks for distributed circuit design,” in *ICML*, 2019, pp. 7364–7373.

[16] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 1097–1105, 2012.

[17] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[18] V. Vapnik, “Statistical learning theory,” *New York*, vol. 1, 1998.

[19] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins, “Near-optimal critical sink routing tree constructions,” *IEEE TCAD*, vol. 14, no. 12, pp. 1417–1436, 1995.

[20] L.-T. Wang, Y.-W. Chang, and K.-T. T. Cheng, *Electronic design automation: synthesis, verification, and test*. Morgan Kaufmann, 2009.

[21] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, “Asap7: A 7-nm finfet predictive process design kit,” *Microelectronics Journal*, vol. 53, pp. 105–115, 2016.