Machine Learning for Yield Learning and Optimization

Yibo Lin, Mohamed Baker Alawieh, Wei Ye, and David Z. Pan

ECE Department, University of Texas at Austin, Austin, TX, USA

***Abstract*—Yield learning and optimization are critical for advanced IC design and manufacturing. Recent advance in machine learning has brought a lot of new opportunities in improving the performance and efficiency of IC yield learning and optimization. This paper surveys some recent results of using various machine learning/deep learning techniques for such purpose, including performance modeling under uncertainty, lithography modeling with transfer/active learning, lithography hotspot detection, and IC mask optimization. The state-of-the-art methods are explained, and challenges/opportunities are discussed.**

I. Introduction

Recent progress in machine learning has triggered advances in various fields, such as computer vision, speech recognition, natural language processing, robotics, and autonomous driving. Tremendous experiments have demonstrated that machine learning is promising to solve data intensive tasks. Machine learning problems can be categorized to supervised learning, unsupervised learning, reinforcement learning, etc. A typical learning procedure, especially for supervised learning, builds a model upon a large training dataset, and expects good generalization on testing datasets under the same distribution.

Successful applications of machine learning techniques have been reported from various tasks. For example,

- Data classification. A data sample can be either a one-dimensional (1D) data vector, a two-dimensional (2D) image-like matrix, or even a multi-dimensional tensor. Each data sample corresponds to a label, denoting its category. The task is to predict the label given any input data sample. A variety of models, such as logistic regression, support vector machine (SVM), and neural networks, are developed to tackle this task.
- Data generation, especially image generation. This task refers to a set of generative models such as auto-encoder and generative adversarial networks (GANs) [1]. The model takes some distribution as input and generate new data following the same distribution. One example for GAN is to take a random latent code and generate images that mimic the ones in the training dataset.
- Language processing. This task corresponds to a set of recurrent models such as recurrent neural networks (RNNs) [2]. Typical tasks include separate negative and positive review comments for movies.
- Reinforcement learning. It mainly tackles problems in robotics where a machine is taught how to take action according to the environment. A famous example is the AlphaGo from Google DeepMind for the goal game [3].

The impressive effectiveness of machine learning enables fast expansion to design and manufacturing of VLSI circuits. Yield estimation and optimization are critical to production costs and design cycles. In this paper, we use *yield learning* to denote both yield estimation and optimization for brevity. Problems in yield learning usually have following characteristics.

- High dimensional. Due to complexity of modern design flow, yield is impacted by huge amount of factors, which are often difficult to model accurately. In lithography related problems, a mask clip can be viewed as an image with very high resolution.
- Data intensive. VLSI designs easily go to millions and billions of transistors. Huge amount of data can be extracted from even one design. In addition, large amount of history data is available from evolution of technology nodes.
- Computationally expensive. The modeling tasks in yield learning often involve complicated physics effects that are expensive to simulate. Monte Carlo simulation is sometimes required to achieve high accuracy. Meanwhile, the optimization tasks usually need iterative invocation of expensive simulations.

With these characteristics, yield learning is a very promising area to take advantages of the performance and efficiency of machine learning techniques. This paper will review recent successful applications of machine learning techniques to yield learning problems, e.g., performance modeling, lithography models, hotspot detection, and mask optimization. The state-of-the-art techniques are explained with motivations, empirical results, and remaining challenges.

In the next few sections, different aspects of using machine learning for yield learning and optimization will be discussed, including performance modeling, lithography modeling, lithography hotspot detection, and mask optimization. The paper is then concluded with future directions in Section VI.

II. Machine Learning for Performance Modeling

A. Statistical Performance Modeling

With the continuous scaling of integrated circuit (IC) technologies, the challenges associated with retaining robustness of state-of-art designs continue to exacerbate [4]. At deep submicron technologies, process variation prevails among the most prominent factors limiting the product yield of analog and mixed-signal (AMS) circuits [4]. Thus, it is indispensable to consider this variation in the design flow of modern ICs. Conventionally, performance modeling has been adopted to capture this variability through analytical models that can be used in various applications such as yield estimation [5]–[7] and design optimization [8], [9].

978-1-5386-8382-8/18/$31.00

Mathematically, a performance model approximates a circuit-level point of interest (PoI), e.g. gain, power, as an analytical function of the process variables:

$$y \approx f_1(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m . b_m(\mathbf{x}) \tag{1}$$

where y is the PoI, $\mathbf{x}$ is a vector containing the process variables (PV), $f_1(\mathbf{x})$ is the modeling function, $\{\alpha_m; m = 1, 2, \ldots, M\}$ contains the model coefficients, $\{b_m; m = 1, 2, \ldots, M\}$ contains the basis functions, and M denotes the total number of basis functions.

Given a set of samples, the model coefficients in (1) are usually obtained through least-squares regression (LSR) by solving the following optimization problem [10], [11]:

$$\min_{\boldsymbol{\alpha}} \quad \|\mathbf{y} - \mathbf{B}.\boldsymbol{\alpha}\|_2^2 \tag{2}$$

where $|| \bullet ||_2$ is the ℓ_2-norm of a vector, and:

$$\mathbf{B} = \begin{bmatrix} b_1(\mathbf{x}^{(1)}) & b_2(\mathbf{x}^{(1)}) & \ldots & b_M(\mathbf{x}^{(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ b_1(\mathbf{x}^{(N)}) & b_2(\mathbf{x}^{(N)}) & \ldots & b_M(\mathbf{x}^{(N)}) \end{bmatrix} \tag{3}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \ldots & \alpha_M, \end{bmatrix}^T ; \mathbf{y} = \begin{bmatrix} y^{(1)} & \ldots & y^{(N)} \end{bmatrix}^T . \tag{4}$$

In (3)-(4), N is the total number of samples, and $\mathbf{x}^{(n)}$ and $y^{(n)}$ are the values of $\mathbf{x}$ and y at the $n-$th sample respectively.

However, LSR can build accurate models only when the number of samples is much greater than the number of unknown coefficients. Thus, given the high dimensionality of the performance models in complex AMS circuit designs, the simulation cost for building accurate models can be exorbitant. Hence, most recent performance modeling techniques incorporate additional information about the model to reduce the number of simulations needed to build accurate models [12]–[16].

B. Performance Modeling applications

One of the main applications of performance modeling is capturing the major sources of variability in the design. In a linear model, the coefficients, $\{\alpha_m; m = 1, 2, \ldots, M\}$, provide the sensitivity information for the PoI with respect to each device-level variation parameter. In practice, the magnitude of a coefficient α_i reflects the contribution of the corresponding device-level variation parameter to the variability of the PoI. In addition, statistical models can be used for worst-case corner extraction for specific application [5].

Moreover, performance modeling can be applied for yield estimation and optimization. In [6], [17], the statistical distribution of the PoI is estimated based on PoI's performance model to estimate the associated parametric yield. In addition, [8], [9] propose using the performance models to help improve the parametric yield by capturing correlations between the performance variability and the device sizes, then adjusting these sizes to improve the parametric yield.

C. Sparse Regression

In high dimensional modeling tasks, additional information about the expected nature of the model can be leveraged to compensate for the limited number of simulations. Sparse regression has been proposed as an approach to build accurate sparse performance models from underdetermined system of equations where the number of samples is fewer than that of the unknown model coefficients [12], [16].

Although the number of basis functions representing the device level variability is large, a few of these basis functions are required to accurately model a specific PoI. Hence, the vector of coefficients $\boldsymbol{\alpha}$ contains a small number of non-zero values corresponding to important basis functions [12]. This information can be incorporated in the optimization problem in (2), resulting in the following new formulation:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \|\mathbf{y} - \mathbf{B}.\boldsymbol{\alpha}\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{\alpha}\|_0 \leq \lambda \end{aligned} \tag{5}$$

where $|| \bullet ||_0$ is the "ℓ_0-norm" of a vector and λ is an upper bound on the number of non-zero coefficients.

While the formulation in (5) accurately reflects the sparse regression concept, the optimization problem is NP-hard. To address this challenge, different approaches have been proposed. In [12], the "ℓ_0-norm" constraint on $\boldsymbol{\alpha}$ is relaxed to an "ℓ_1-norm" constraint. As a results of this change, the optimization problem can be re-formulated into a convex optimization problem [12].

Other approaches proposed heuristic methods to solve the optimization problem in (5) [16]. These approaches iteratively choose a small number of important basis functions to include in the model by examining the correlation between the basis function and the performance values. As a first step, the correlations between all basis functions and the PoI are computed, then the basis function with the highest correlation is included in the set of important basis functions. At the end of each iteration, LSR is used to build a model using the important basis functions only. This process continues until the λ most important basis function are chosen or a user-defined stopping criteria is satisfied. Moreover, and since the value of λ is not known beforehand, a cross-validation based approach is adopted to arrive at its optimal value [12], [16].

D. Bayesian Methods

In literature, Bayesian analysis has been used to efficiently build accurate performance models [14], [15], [17]. Bayesian Model Fusion (BMF) leverages the fact that designing an AMS circuit typically involves multiple stages (e.g., schematic simulation, layout and post-layout simulation, etc.) [15]. Throughout the different stages, simulation data is generated to verify all performance metrics at each stage. BMF proposes to incorporate early stage simulation data to efficiently build accurate models at a late stage.

In principal, BMF starts by building a performance model for the PoI at at the early-stage which provide a prior knowledge of the late-stage model template. Then, using a Bayesian inference framework, the model template is fused with a small number of simulation data collected at the later stage to build the target model [15].

Experimental results presented in [15] show that when applied to an SRAM, with the PoI defined as the read delay from the word line to the sense amplifier output, BMF demonstrated a

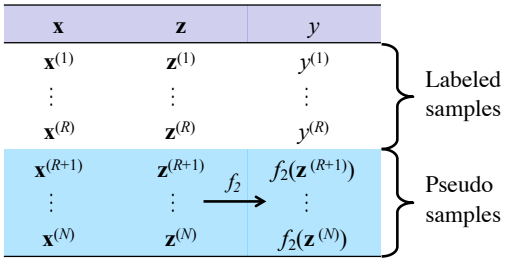

Fig. 1 The CL-BMF procedure is illustrated where $f_1(\mathbf{x})$ is fitted using labeled and pseudo samples. [14].

8x runtime speedup over sparse regression without surrendering any accuracy.

Moreover, a Co-Learning Bayesian Model Fusion (CL-BMF) approach has been proposed to leverage performance side information to efficiently build accurate performance models [14]. The key idea of CL-BMF is to build two models to estimate the same performance metric. While one of the two models is that relating the PoI to the process variation, the other model uses other performance metrics in the design to estimate the same PoI. In practice, CL-BMF passes the knowledge from the alternative low-complexity model to the high complexity model (i.e. co-learning) to reduce its training cost. The performance model of interest is treated as the high-complexity model, while the low-complexity model provides the performance side information (PSI) to reduce the training cost of the high-complexity model. These is done using a Bayesian inference framework that combines: (i) the performance side information which enables co-learning, (ii) the prior knowledge about the model coefficients, and (iii) a small number of training samples collected to build the performance model.

In practice, the PSI used in CL-BMF is typically a low-dimensional vector of alternative performance metrics in the circuit, $\mathbf{z}$, that is inexpensive to measure or simulate and can be used to accurately predict the PoI $y \approx f_2(\mathbf{z})$. With such choice of $\mathbf{z}$, f_2 can be used to generate cheap pseudo samples for y that can be fused with a set of labeled samples to build the final model f_1 as demonstrated in Fig. 1. In the figure, R samples where y was actually measured are used alongside $N-R$ samples where only $\mathbf{z}$ was measured in the circuit, and where pseudo samples for y were generated, to build the model of interest [14].

E. Semi-Supervised Learning

The aforementioned performance modeling methods all fall into the category of supervised learning. In other words, performance models are built by using labeled or, at least, partially labeled data only. Recently, a new direction, derived from semi-supervised learning, was proposed to take advantage of unlabeled data to further improve the accuracy of performance modeling for AMS designs [13], [18], [19].

The proposed technique in [13] makes use of the hierarchical structure of an AMS circuit to incorporate unlabeled data via Bayesian co-learning. In particular, the proposed approach is composed of three major components. First, the entire circuit of interest is partitioned into multiple blocks as shown in Fig. 2. Second, circuit-level performance models are built to map the block-level performance metrics to the PoI at the

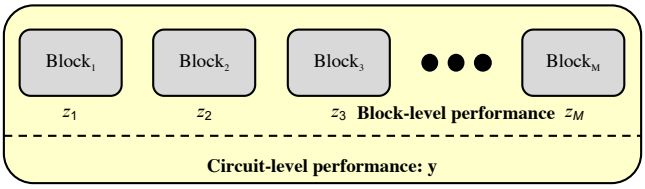

Fig. 2 Hierarchical structure of AMS is exploited to leverage semi-supervised leanring in performance modeling [13], [18].

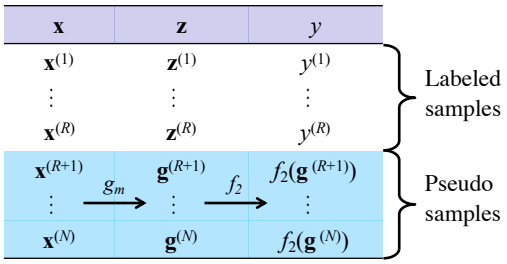

Fig. 3 The hierarchical semi-supervised Bayesian modeling framework is illustrated [13], [18].

circuit level. Such a mapping is often low-dimensional; thus it can be accurately approximated by using a small number of simulation samples. Third, by combining the aforementioned low-dimensional models and an unlabeled data set, a complex, high-dimensional performance model for the PoI can be built based on semi-supervised learning as shown in Fig. 3.

To implement this modeling technique, a Bayesian inference is formulated to integrate the aforementioned three components, along with the prior knowledge on model coefficients, in a unified framework. Experimental results shown in [13] demonstrate that the proposed semi-supervised leaning approach can achieve upto 3.6x speedup when compared to sparse regression based approach.

While the proposed approach in [13] assumes a hierarchical structure for the AMS design, a more general semi-supervised framework was proposed in [19] which makes no assumption about the AMS circuit structure. The proposed framework incorporates a co-learning technique that leverages multiple views of the process variability to efficiently build a performance model. The first is the device level variations such as $\Delta_{V_{TH}}$ or $\Delta_{w_{eff}}$, while the second view is the underlying set of independent random variables, referred to as process variables. Traditionally, performance modeling targets expressing the PoI as an analytical function of PV; however, [19] proposes to capitalize on the information provided by the device level variability as an alternative view to efficiently build the performance model for the PoI.

As shown in Fig. 4, the key idea is to use a small number of labeled samples to build an initial model for each of the views of the data, then attempt to iteratively bootstrap from the initial models using unlabeled data. In other words, initial models can be used to give pseudo labels for unlabeled data, then the most confident predictions from a particular model are used as pseudo samples for the other model. In each iteration step, *highly-confident* pseudo samples are fused with the small number of available labeled samples to build a new model. The experimental results demonstrated upto 30% reduction in

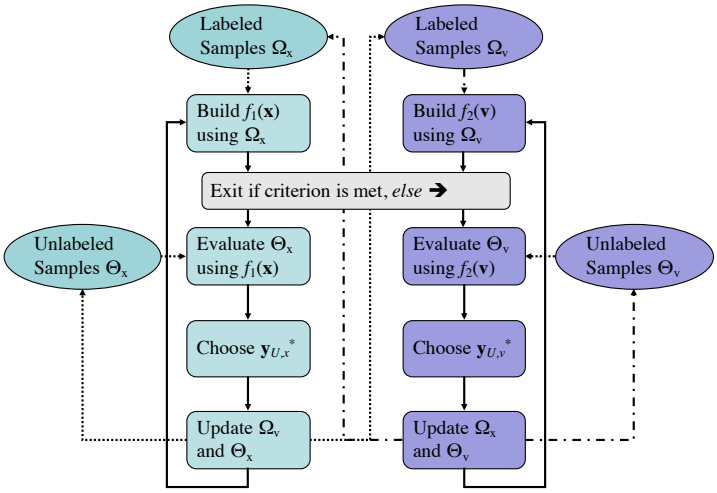

Fig. 4 The semi-supervised col-learning modeling framework is illustrated [19].

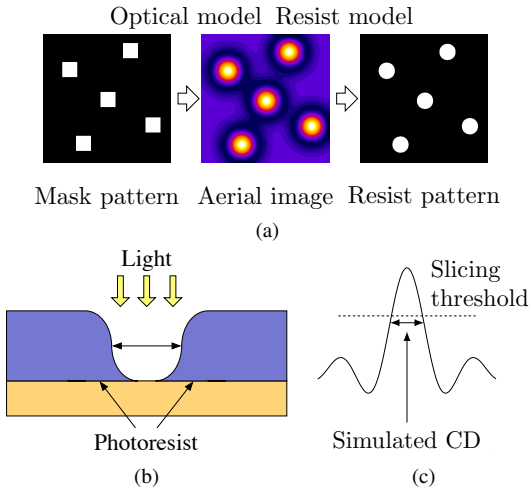

Fig. 5 (a) Process of lithography simulation with optical and resist models [20]. (b) Photoresist exposed to light. (c) Thresholds for aerial image determine simulated CD, which should match manufactured CD.

simulation cost when compared to sparse regression based approaches.

F. Challenges

Recent works starting from the CL-BMF framework in [14], where partially labeled samples are utilized in the modeling task, and then the leverage of unlabeled samples in [18] has demonstrated a new trend in the performance modeling field. In fact, the focus has shifted from the computationally expensive *supervised learning* paradigm to the *semi-supervised learning* paradigm where a smaller number of simulations is needed to build accurate models. This shift was mainly driven by the technology scaling which is continuously making the modeling task more challenging. Despite the advancement in the performance modeling field, some challenges are still present, with the two major ones being (i) the nonlinear behaviour and (ii) dimensionality.

Most of the techniques used to address the performance modeling task use polynomial models which are in practice parametric models. With the continuous scaling of IC technology, performance models are becoming increasingly nonlinear, and the polynomial models will soon fall short of capturing such highly nonlinear behavior.

While using non-parametric models can help address the nonlinearity challenge, it further exacerbates the dimensionality problem. Hence, new techniques are needed to address the dimensionality problem to pave the may to non-parametric modeling. The first attempt towards this goal was seen with the semi-supervised learning techniques; however, more work is needed at this front.

III. Machine Learning for Lithography Modeling

Lithography modeling takes mask design as input and simulates the printed patterns of the lithography system. Fig. 5(a) shows the typical process on how a lithography model operates for contact layers. The first step uses the optical model to compute the light intensity on the photoresist. Photoresist exposed to strong light will be removed, as shown in Fig. 5(b). The degree to which photoresist will be removed depends on various factors, i.e., the property of the photoresist material, lithography system, and mask patterns. Thus in the second step, the resist model is used to compute the locations, shapes and sizes of the printed pattern according to the light intensity map, i.e., aerial image, from previous step. One intuitive understanding about the resist model is to determine the threshold to cut through the light intensity map for each layout pattern, such that the simulated pattern matches the actually printed one, as shown in Fig. 5(c). Accurate lithography modeling is critical to guarantee yield, but time consuming as well.

A. Accuracy-Driven Modeling

It is usually difficult to directly measure the light intensity on the photoresist, so the objective of lithography modeling is to match the eventual output patterns with the manufactured ones. Several problem formulations have been proposed for this problem. Shim et al [21] proposed an artificial neural networks (ANN) to predict the 3D height of photoresist at any given location in the layout, as shown in Fig. 6. The model takes a feature vector sampled from a layout clip centered by the point of interest and feeds to the ANN. Each feature vector consists of 49 density values from 6 concentric circles with 10nm intervals. With 5 hidden layers and 7 hidden nodes for each layer, they demonstrate an accuracy about 5% root mean square (RMS) error of the initial resist height.

Another problem formulation of lithography modeling is to predict the 2D printed resist patterns instead of the resist height. The output will be a 2D image like the last figure in Fig. 5(a). This problem can be tackled in two ways. Given a layout clip, we can compute its light intensity and then predict its slicing threshold for each feature. Once the threshold is known, recovering the printed patterns using light intensity map is possible. This method is also called various threshold approach (VTR). The other method is to modulate the light intensity map with Gaussian kernels, such that a constant threshold

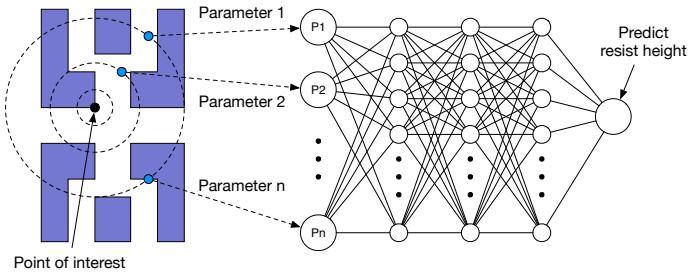

Fig. 6 ANN model to predict resist height [21].

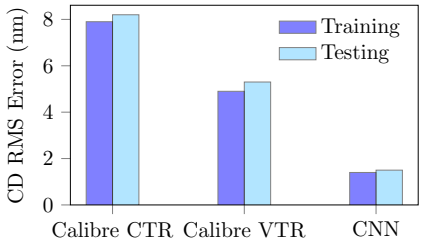

Fig. 7 Comparison between CNN model and Calibre CTR and VTR models [22].

can be used for any input layout clip. This is called constant threshold approach (CTR). Watanabe et al [22] adopted the VTR model and construct a convolutional neural networks (CNN) to predict the thresholds for the center contact of a given layout clip. With 3 convolution layers and 2 fully connected layers, they demonstrate 70% smaller prediction errors compared with conventional VTR and CTR models in Mentor Graphics Calibre [23], as shown in Fig. 7.

B. Data Efficient Modeling

Lin et al [20] argue that the cost of data preparation is very high for lithography modeling, i.e., requiring measurement from manufactured wafers. Meanwhile, the accuracy and generality of a model highly depend on the amount of training data available. Therefore, reducing the amount of data required to achieve high modeling accuracy is necessary. By observing the potential similarity between datasets from neighboring technology nodes, they integrate both transfer learning and active training data selection into the VTR modeling problem. Fig. 8 shows the flow of training an N7 model with the assistance of N10 data. There are two assumptions in the flow.

- Large amount of N10 data is available and there are correlation between N10 and N7 data.
- N7 dataset is initially unlabeled and querying the labels for N7 data samples is possible.

The knowledge transfer is realized with a TF_k scheme in Fig. 9 in which an N10 model is first trained with N10 data and then weights are fine-tuned with N7 data to obtain the N7 model. The first k layers are fixed in fine-tuning. Smaller parameter k indicates more flexibility in fine-tuning, while larger k means less flexibility. The active data selection tries to further reduce the amount of training data required by selecting representative N7 data samples for training, as shown in Fig. 10. A K-Medoids clustering technique is developed to increase the coverage of

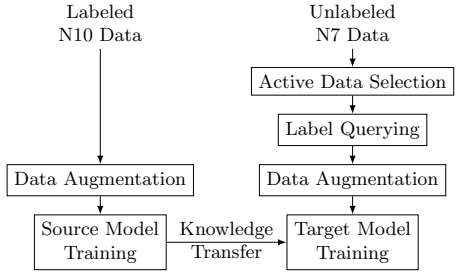

Fig. 8 Training flow with transfer learning and active learning [20].

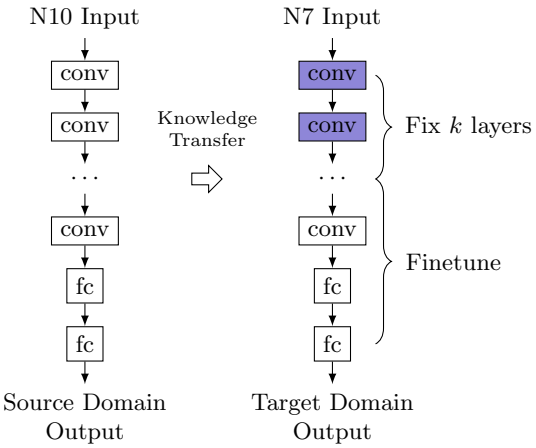

Fig. 9 Transfer learning scheme with the first k layers fixed when training for target domain, denoted as TF_k [20].

the entire data space. They also prove that with assumptions on the Lipschitz continuity, some extent of generality can be guaranteed.

Fig. 11 shows the amount of training data required given a target accuracy in the RMS errors of critical dimensions (CD). "CNN" denotes training with pure N7 data like that in [22], "CNN TF_0" denotes the incorporation of transfer learning to CNN, "ResNet TF_0" denotes the incorporation of transfer learning to residual neural networks (ResNet), and "ResNet TF_0 + AL" denotes that ResNet is trained with both transfer learning and active data selection. Transfer learning can achieve $2-10$X reduction on the amount of required training data. Integrating active data selection can further reduce the amount for some target CD RMS errors such as 1.5nm and 2.25nm. Eventually $3-10$X reduction can be achieved.

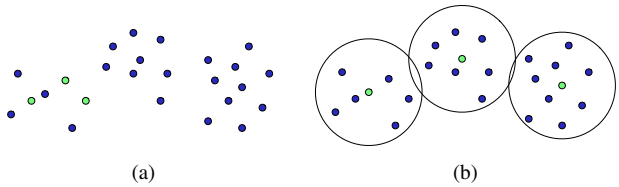

Fig. 10 Example of (a) bad data selection and (b) K-Medoids clustering selection in 2D space [20]. Three selected points are highlighted. Circles denote three clusters centered by selected points.

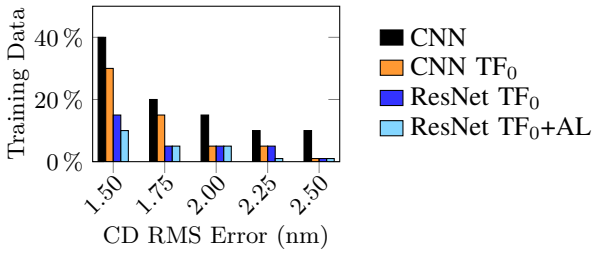

Fig. 11 Amount of training data required for N7 given target CD RMS errors with enough N10 data available [20].

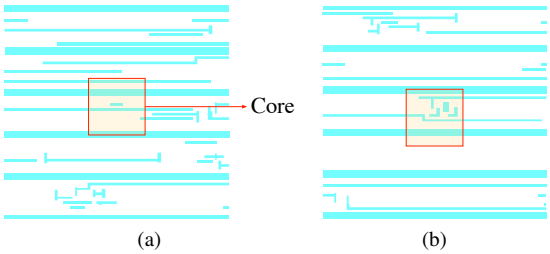

Fig. 12 Example of hotspot patterns [24].

C. Challenges

Lithography modeling remains to be an active area for machine learning applications. There are still various challenges. For instance, the formulations aforementioned are mostly regression tasks and most of the time it still requires the calculation of light intensity map (aerial image in Fig. 5(a)). Aerial image simulation is still time consuming. Eventually end-to-end learning that takes a mask pattern as input and produce the corresponding contours of the resist pattern as output is desired. This requires generative models such as auto-encoder or generative adversarial networks [1]. However, the challenge lies in the high resolution of images. The lithography critical dimension is usually around $2-3\mu m$, which means a feature may be impacted by features $2-3\mu m$ away from it. Suppose the target RMS error is 1nm for an end-to-end model. Then we need to sample at least a 4000×4000 image as the input to the model and only 1 pixel of mis-prediction is allowed near the contours of the printed patterns. The large image sizes lead to complicated networks and slow inference. Meanwhile, the requirement of high accuracy is difficult to achieve as well. Therefore, compressed representation of the images or network sketching techniques may be useful to tackle this problem.

IV. Machine Learning for Hotspot Detection

Various design for manufacturability (DFM) techniques have been proposed to bridge the wide gap between design demands and manufacturing limitations introduced by the current mainstream 193nm lithography. However, due to the complexity of lithography systems and process variation, failure to print specific patterns still happens, known as lithography hotspot. Examples of two hotspot patterns are shown in Fig. 12.

Lithography hotspot detection often requires expensive lithography simulation. Therefore, efficient and accurate lithography hotspot detection is desired for layout finishing and design closure. A general problem formulation for hotspot detection is as follows.

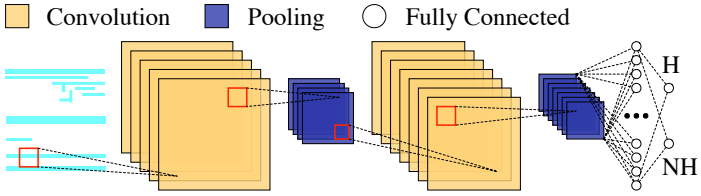

Fig. 13 Example illustration of convolutional neural network architecture for hotspot detection [48]. "H" denotes hotspot and "NH" denotes non-hotspot.

Problem 1 (Hotspot detection)**.** *Given two sets with hotspots and non-hotspots layout clips, one as training set, the other as testing set, the task of hotspot detection is to construct a model based on the training set, such that the model can classify hotspots on the testing set with maximum accuracy and minimum false alarms.*

The *detection accuracy* is the ratio between the number of correctly-detected hotspots and the number of real hotspots, while the *false alarm* is defined as the number of non-hotspots that are wrongly recognized as hotspots.

Existing hotspot detection methods mainly fall into three categories: lithography simulation, pattern matching, and machine learning. Lithography simulation is reliable but time consuming [25], [26]. Pattern matching methods first build a library of hotspot patterns. Any new pattern to detect is compared with the existing patterns in the library and marked as hotspot if a match was found [27], [28]. This type of techniques including fuzzy pattern matching lacks the capability to predict never-before-seen hotspot patterns [29], [30]. On the other hand, machine learning approaches have demonstrated good generalization capability to recognize unseen hotspot patterns [31]–[39]. These methods generally perform one-time training on a labeled dataset to obtain a machine learning model and predict whether a new layout pattern is a hotspot or not efficiently.

A. Machine Learning Models

Various machine learning models have been used as hotspot detection kernels including SVM [40], [41], ANN [40], and boosting methods [42], [43]. Zhang et al [43] also propose an online learning scheme to verify newly detected hotspots and incrementally update the model. Deep neural network (DNN) classifier has been adopted for hotspot detection [44], [45]. DNN is able to take the high-dimensional layout and perform automatic feature extraction during training, which avoids the manual efforts to reduce select feature extraction methods. Promising empirical results have been observed with DNN in several papers [44]–[47]. Fig. 13 gives a typical configuration of DNN structure.

In spite of the convenience in automatic feature extraction, the performance of DNN highly relies on manual efforts to tune the networks, e.g., the number and types of layers. Matsunawa et al [45] propose a DNN structure that can achieve low false alarms. Yang et al [48] propose Discrete Cosine Transform (DCT) based

feature representation to reduce the image size for DNN with a biased learning to improve accuracy and decrease false alarms.

B. Challenges

Machine learning techniques have achieved tremendous success in lithography hotspot detection. However, to label enough layout patterns in preparation for the training set, a large number of lithography simulations are required. Especially for layout designs at the early stage of a new technology node, the amount of labeled data samples is limited. Hence, it remains challenging to build machine learning models with a much smaller number of training instances while maintaining high detection performance of the models, i.e., increasing the data efficiency.

There are several options to improve the data efficiency for hotspot detection such as semi-supervised learning and active learning. Semi-supervised learning can be a viable option to reduce the upfront cost and time associated with training a model. It leverages both labeled and unlabeled samples to help the model training and alleviate the dependency to a large amount of labeled training data. It is being actively explored in image recognition, neural language processing, etc [49].

Besides the data efficiency issue, [50] shows that layout datasets are highly imbalanced because the number of lithography hotspots is much less than the number of non-hotspot patterns. With such a setup, a large number of samples are needed to guarantee enough hotspot samples for building accurate classification models. This translates to an enormous computational cost associated with running a large number of lithography simulations. Active learning is a special case of semi-supervised learning, and has the ability to query instances based upon past queries and the labels from those queries. It determines which new instances to be labeled and added into the training set [51], [52]. For example, it is preferred to select more hotspots for training to improve the accuracy.

V. Machine Learning for Mask Optimization

Mask synthesis takes a significant amount of time in the back-end design flow. A typical flow of mask synthesis is shown in Fig. 14(a). Sub-resolution assist feature (SRAF) generation and optical proximity correction (OPC) are two key steps in mask synthesis. SRAF refers to the small features that will not be actually printed on wafers but may help the printing of target features. OPC refers to shifting the edge segments of features for robust lithography printing. Both SRAF and OPC need to pass mask rule check (MRC) and lithography compliance check (LCC).

The robustness of lithography printing is usually evaluated with edge placement error (EPE) and process variation band (PVBand) by performing lithography simulation under different {focus, dose} conditions. Fig. 14(b) gives an example for definitions of EPE and PVBand. Under various lithography conditions, an inner contour, an outer contour, and a nominal contour can be obtained for each feature. EPE is defined as the edge difference between the nominal contour and the design target. PVBand is defined as the band between inner and outer contours, measuring the sensitivity to process variations. The objective of mask synthesis is to minimize EPE and PVBand.

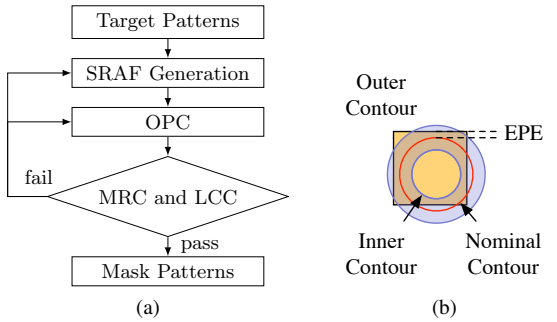

Fig. 14 (a) Mask synthesis flow; (b) EPE and PVBand.

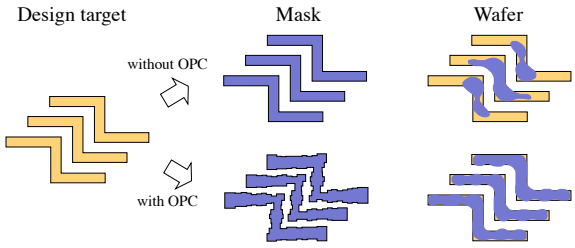

Fig. 15 Motivation of OPC [53].

Fig. 15 illustrates the motivation of OPC. Due to the extremely small feature sizes, the actual printed patterns may be completely off the design target, while OPC shifts the edge segments of target features such that the printed patterns approach the design targets. OPC alone is not enough to achieve robust lithography printing. Fig. 16 demonstrates the necessity of SRAF generation, because the robustness to process variation also needs to be minimized. Fig. 16(b) shows the PVBand of printing contours for a contact. By inserting SRAFs in Fig. 16(c), PVBand can be decreased. At the same time, the SRAFs are too small to be printed on the wafer, which means eventual wafer will only contain design targets.

A. Optical Proximity Correction

Conventional model-based OPC requires iterative and massive calls of lithography simulation [55]. Although it is able to generate high quality solutions, it is time consuming and the convergence is slow. Previous work proposes regression based approaches to achieve fast full-chip OPC with acceptable performance loss [53], [56], [57]. In these approaches, design targets are fragmented and features are extracted from the fragmented layout for model training and prediction. Due to the complicated

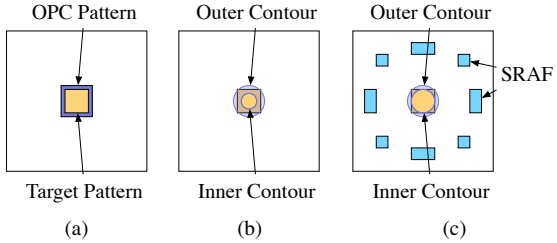

Fig. 16 Motivation of SRAF [54]. (a) An isolated contact and its OPC pattern; (b) printed pattern with OPC only; (c) printed pattern with both SRAF and OPC.

optical proximity effects, the regression models need to be complex, but the models often suffer from over-fitting issues. It is difficult to find general yet accurate ones.

To overcome the over-fitting issue, Matsunawa et al [58] proposed a hierarchical Bayes model (HBM) with concentric circular area sampling (CCAS) as the feature extraction technique. HBM trains a generalized linear mixed model to consider various edge types, including normal, convex, concave, and line-end edge, by regarding each edge type as a random effect with a random variance. HBM can generate solutions with comparable quality to that from 10 iterations of conventional model-based approach, while it is much more efficient. Thus, HBM can be used as a starting point for model-based approach and speedup the convergence.

Recently, Yang et al [59] presented GAN-OPC, i.e., the first generative adversarial neural networks (GAN) for OPC. They incorporate conditional GAN [60] and jointly train a generator and a discriminator for OPC. Fig. 17 shows the architecture of GAN-OPC. The generator has an auto-encoder structure [61], which takes a design target and outputs a mask clip. The discriminator is a classifier that differentiates generated masks and reference masks.

Consider target patterns $\mathcal{Z} = \{Z_i^t, i = 1, 2, \ldots, N\}$ and the corresponding reference mask set $\mathcal{M} = \{M_i^*, i = 1, 2, \ldots, N\}$. Let $\mathbf{G}(z; W_g)$ represent a generator that generates z following a distribution p_g, where W_g is the parameters for the generator. Let $\mathbf{D}(x; W_d)$ represent the probability of x drawn from a distribution p_d, where W_d is the parameters for the discriminator. For simplicity, we sometimes drop W_g and W_d for $\mathbf{G}$ and $\mathbf{D}$, respectively. The objective of discriminator is to differentiate generated mask and reference mask,

$$\begin{aligned}\max_{\mathbf{D}} \quad & \mathbb{E}_{Z^t\sim\mathcal{Z}}[\log \mathbf{D}(Z^t, M^*)] \\ & +\mathbb{E}_{Z^t\sim\mathcal{Z}}[1 - \log \mathbf{D}(Z^t, \mathbf{G}(Z^t))].\end{aligned} \tag{6}$$

The objective of generator is to deceive the discriminator by maximizing the log-likelihood of the discriminator predicting the generated mask is the reference mask,

$$\max_{\mathbf{G}} \quad \mathbb{E}_{Z^t\sim\mathcal{Z}}[\log \mathbf{D}(Z^t, \mathbf{G}(Z^t))]. \tag{7}$$

We also want to minimize the difference between M^* and $\mathbf{G}(Z^t)$,

$$\min_{\mathbf{G}} \quad \mathbb{E}_{Z^t\sim\mathcal{Z}}\left\|M^* - \mathbf{G}(Z^t)\right\|_2. \tag{8}$$

Combining all the objectives, the training process can be formulated into a min-max game,

$$\begin{aligned}\min_{\mathbf{G}}\max_{\mathbf{D}} \quad & \mathbb{E}_{Z^t\sim\mathcal{Z}}[\log \mathbf{D}(Z^t, M^*)] \\ & +\mathbb{E}_{Z^t\sim\mathcal{Z}}[1 - \log \mathbf{D}(Z^t, \mathbf{G}(Z^t))] \\ & +\mathbb{E}_{Z^t\sim\mathcal{Z}}\left\|M^* - \mathbf{G}(Z^t)\right\|_2^2,\end{aligned} \tag{9}$$

where the last term of ℓ_2 norm for Eq. (8) is squared for optimization. Eq. (9) can be optimized with various stochastic gradient descent algorithms [1].

The output of GAN-OPC is used as the starting point for a conventional ILT engine [62], as shown in Fig. 18. Compared with conventional ILT, GAN-OPC flow is reported to achieve 9% reduction in EPE error, 1% reduction in PVBand, and over 2X reduction in the overall runtime. The runtime for the generator

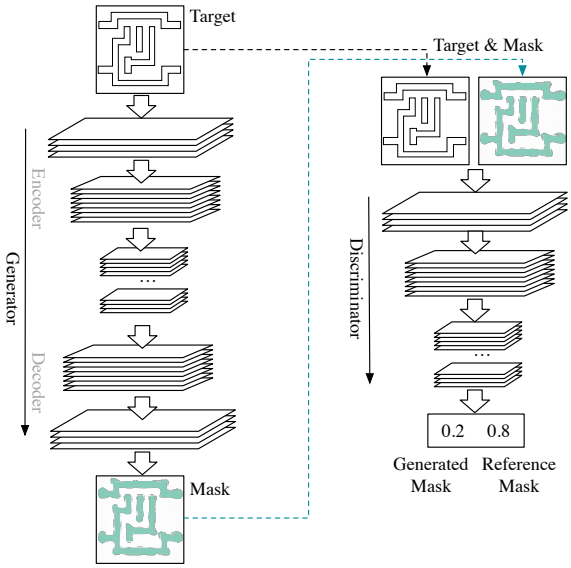

Fig. 17 Neural network architecture of GAN-OPC [59].

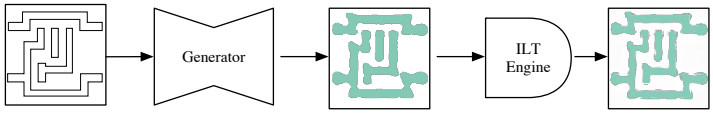

Fig. 18 GAN-OPC flow [59].

inference is only around 0.05% of the entire runtime. GAN-OPC not only speedups the convergence of ILT, but also improve the solution quality.

B. Sub-Resolution Assist Feature Generation

Conventional SRAF generation includes model-based and rule-based approaches. Model-based SRAF generation usually requires lithography simulation to determine the locations for SRAF insertion, which leads to high-quality and robust lithography printing [63]–[66], but it is not very scalable to large designs. Rule-based approaches leverage complicated look-up tables to achieve ultra fast turn-around time, while its performance heavily relies on the look-up tables and it requires huge amount of engineering effort to maintain them [67], [68]. Machine learning approaches can bridge the gap between fast rule-based approaches and high-quality model-based ones [54], [69].

Existing machine learning approaches divide a layout clip into small grids/pixels and formulate a classification problem, as shown in Fig. 19(a) [54]. A pixel with label 0 indicates no SRAF at the corresponding location and that with label 1 indicates that an SRAF should be inserted. Thus, the classification problem is to predict a label for each pixel given its feature vector. The feature vector for a pixel is sampled using CCAS, as shown in Fig. 19(b). In order to generate SRAFs for the entire layout clip, it is necessary to make predictions for all pixels. A post processing step is followed to actually insert SRAFs with the guide of predictions and following all the design rules. Experimental results demonstrate 10X speedup in $1 - 2\mu m^2$ layout windows, and 3X speedup for $100\mu m^2$ layout clips, with logistic regression or SVM as the classifiers. SVM provides

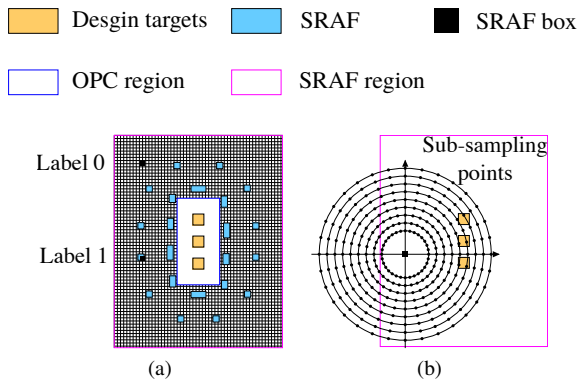

Fig. 19 (a) SRAF label extraction and sampling; (b) CCAS at one grid point [54].

better quality than does logistic regression, i.e., around 13.5% better EPE with 1.6% degradation in PVBand compared with model-based SRAF generation.

C. Challenges

Despite the success of applying machine learning techniques to OPC and SRAF, there are still challenges. For GAN-OPC, there are two major challenges. Firstly, it is very difficult to achieve good convergence for training, which is a common issue in GAN [1]. Although techniques like pre-training with auto-encoder may help, the performance differs significantly with different initialization. Secondly, after using the generative model to produce the starting point for the ILT engine, the runtime bottleneck is still dominated by the ILT engine. Therefore, to improve the runtime, the generative model needs to achieve even higher accuracy such that ILT iterations can be further reduced.

In SRAF generation, machine learning models are used to make predictions for one pixel at a time. For a layout clip with $M \times N$ pixels, MN predictions are required. Since a layout clip usually has width and height in the scale of micrometers and the precision should have the scale of nanometers, M and N are quite large. This eventually limits the usage of complicated models for higher prediction accuracy. Therefore, pixel-wise modeling approach suffers from the runtime bottleneck to further improve the solution quality. Generative models like GAN-OPC (Section V-A) are a promising approach to resolve this issue.

VI. Conclusion

This paper reviews the recent machine learning applications to various yield learning problems, such as performance modeling, lithography modeling, hotspot detection, and mask optimization. Due to the fact that yield learning problems are usually high dimensional, data intensive, and computationally expensive, machine learning is able to improve both performance and efficiency, eventually contributing to fast design closure and good manufacturability.

On the other hand, there are still challenges for the machine learning applications, as discussed in Section II-F, III-C, IV-B, and V-C. The major challenges can be summarized as,

- requirement of high resolution;
- requirement of large amount of training data;
- imbalanced dataset;
- requirement of high accuracy and low false alarm;
- requirement of high inference efficiency.

These are issues are remaining to be solved/improved with more powerful and efficient learning techniques.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[2] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[3] "Google DeepMind," https://deepmind.com/.

[4] X. Li and L. Pileggi, *Statistical Performance Modeling and Optimization*. Now Publishers, 2007.

[5] H. Zhang, T.-H. Chen, M.-Y. Ting, and X. Li, "Efficient design-specific worst-case corner extraction for integrated circuits," in *ACM/IEEE Design Automation Conference (DAC)*, 2009, pp. 386–389.

[6] X. Li, J. L. Le, P. Gopalakrishnan, and L. Pileggi, "Asymptotic probability extraction for nonnormal performance distributions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 26, no. 1, pp. 16–37, 2006.

[7] F. Gong, Y. Shi, H. Yu, and L. He, "Variability-aware parametric yield estimation for analog/mixed-signal circuits: Concepts, algorithms, and challenges," *IEEE Design & Test*, vol. 31, no. 4, pp. 6–15, 2014.

[8] 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, pp. 1–6.

[9] M. B. Alawieh, F. Wang, R. Kang, X. Li, and R. Joshi, "Efficient analog circuit optimization using sparse regression and error margining," in *IEEE International Symposium on Quality Electronic Design (ISQED)*, 2016, pp. 410–415.

[10] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[11] S. boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[12] X. Li, "Finding deterministic solution from underdetermined equation: large-scale performance variability modeling of analog/rf circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 29, no. 11, pp. 1661–1668, 2010.

[13] M. B. Alawieh, F. Wang, and X. Li, "Efficient hierarchical performance modeling for integrated circuits via bayesian co-learning," in *ACM/IEEE Design Automation Conference (DAC)*, 2017, pp. 1–6.

[14] F. Wang, M. Zaheer, X. Li, J.-O. Plouchart, and A. Valdes-Garcia, "Co-learning bayesian model fusion: Efficient performance modeling of analog and mixed-signal circuits using side information," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016, pp. 575–582.

[15] F. Wang, P. Cachecho, W. Zhang, S. Sun, X. Li, R. Kanj, and C. Gu, "Bayesian model fusion: large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 35, no. 8, pp. 1255–1268, 2015.

[16] X. Li, "Finding deterministic solution from underdetermined equation: large-scale performance modeling by least angle regression," in *ACM/IEEE Design Automation Conference (DAC)*, 2009, pp. 364–369.

[17] Q. Huang, F. Fang, Chenlei amd Yang, X. Zeng, and X. Li, "Efficient multivariate moment estimation via bayesian model fusion for analog and mixed-signal circuits," in *ACM/IEEE Design Automation Conference (DAC)*, 2015, pp. 1–6.

[18] M. B. Alawieh, F. Wang, and X. Li, "Efficient hierarchical performance modeling for analog and mixed-signal circuits via bayesian co-learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, pp. 1–13, 2018.

[19] M. B. Alawieh, X. Tang, and D. Z. Pan, "S^{2}PM : Semi-supervised learning for efcient performance modeling of analog and mixed signal circuits," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*.

[20] Y. Lin, M. Li, Y. Watanabe, T. Kimura, T. Matsunawa, S. Nojima, and D. Z. Pan, "Data efficient lithography modeling with transfer learning and active data selection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, pp. 1–1, 2018.

[21] S. Shim, S. Choi, and Y. Shin, "Machine learning-based resist 3d model," in *Proc. of SPIE Vol*, vol. 10147, pp. 101 471D–1.

[22] Y. Watanabe, T. Kimura, T. Matsunawa, and S. Nojima, "Accurate lithography simulation model based on convolutional neural networks," in *SPIE Advanced Lithography*. International Society for Optics and Photonics, 2017, pp. 101 470K–101 470K.

[23] Mentor Graphics, "Calibre verification user's manual," 2008.

[24] J.-R. Gao, B. Yu, and D. Z. Pan, "Accurate lithography hotspot detection based on PCA-SVM classifier with hierarchical data clustering," in *Proceedings of SPIE*, vol. 9053, 2014.

[25] J. Kim and M. Fan, "Hotspot detection on Post-OPC layout using full chip simulation based verification tool: A case study with aerial image simulation," in *Proceedings of SPIE*, vol. 5256, 2003.

[26] E. Roseboom, M. Rossman, F.-C. Chang, and P. Hurat, "Automated full-chip hotspot detection and removal flow for interconnect layers of cell-based designs," in *Proceedings of SPIE*, vol. 6521, 2007.

[27] J. Xu, S. Sinha, and C. C. Chiang, "Accurate detection for process-hotspots with vias and incomplete specification," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2007, pp. 839–846.

[28] Y.-T. Yu, Y.-C. Chan, S. Sinha, I. H.-R. Jiang, and C. Chiang, "Accurate process-hotspot detection using critical design rule extraction," in *ACM/IEEE Design Automation Conference (DAC)*, 2012, pp. 1167–1172.

[29] S.-Y. Lin, J.-Y. Chen, J.-C. Li, W.-Y. Wen, and S.-C. Chang, "A novel fuzzy matching model for lithography hotspot detection," in *ACM/IEEE Design Automation Conference (DAC)*, 2013, pp. 68:1–68:6.

[30] W.-Y. Wen, J.-C. Li, S.-Y. Lin, J.-Y. Chen, and S.-C. Chang, "A fuzzy-matching model with grid reduction for lithography hotspot detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 33, no. 11, pp. 1671–1680, 2014.

[31] D. G. Drmanac, F. Liu, and L.-C. Wang, "Predicting variability in nanoscale lithography processes," in *ACM/IEEE Design Automation Conference (DAC)*, 2009, pp. 545–550.

[32] D. Ding, J. A. Torres, and D. Z. Pan, "High performance lithography hotspot detection with successively refined pattern identifications and machine learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 30, no. 11, pp. 1621–1634, 2011.

[33] D. Ding, B. Yu, J. Ghosh, and D. Z. Pan, "EPIC: Efficient prediction of IC manufacturing hotspots with a unified meta-classification formulation," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2012, pp. 263–270.

[34] Y.-T. Yu, G.-H. Lin, I. H.-R. Jiang, and C. Chiang, "Machine-learning-based hotspot detection using topological classification and critical feature extraction," in *ACM/IEEE Design Automation Conference (DAC)*, 2013, pp. 671–676.

[35] T. Matsunawa, J.-R. Gao, B. Yu, and D. Z. Pan, "A new lithography hotspot detection framework based on AdaBoost classifier and simplified feature extraction," in *Proceedings of SPIE*, vol. 9427, 2015.

[36] Y.-T. Yu, G.-H. Lin, I. H.-R. Jiang, and C. Chiang, "Machine-learning-based hotspot detection using topological classification and critical feature extraction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 34, no. 3, pp. 460–470, 2015.

[37] H. Zhang, B. Yu, and E. F. Y. Young, "Enabling online learning in lithography hotspot detection with information-theoretic feature optimization," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016, pp. 47:1–47:8.

[38] Y. Tomioka, T. Matsunawa, C. Kodama, and S. Nojima, "Lithography hotspot detection by two-stage cascade classifier using histogram of oriented light propagation," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2017, pp. 81–86.

[39] H. Zhang, F. Zhu, H. Li, E. F. Y. Young, and B. Yu, "Bilinear lithography hotspot detection," in *ACM International Symposium on Physical Design (ISPD)*, 2017, pp. 7–14.

[40] D. Ding, B. Yu, J. Ghosh, and D. Z. Pan, "Epic: Efficient predition of ic manufacturing hotspots with a unified meta-classfification formulation," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2012.

[41] Y.-T. Yu, G.-H. Lin, I. H.-R. Jiang, and C. Chiang, "Machine learning based hotspot detection using topological classification and critical feature extraction," in *ACM/IEEE Design Automation Conference (DAC)*, 2013.

[42] T. Matsunawa, J.-R. Gao, B. Yu, and D. Z. Pan, "A new lithography hotspot detection framework based on adaboost classifier and simplified feature extraction," in *Proceedings of SPIE*, vol. 9427, 2015.

[43] H. Zhang, B. Yu, and Y. F. Evangeline, "Enabling online learning in lithography hotspot detection with information-theoretic feature optimization," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016.

[44] M. Shin and J.-H. Lee, "Accurate lithography hotspot detection using deep convolutional neural networks," in *Journal of Micro/Nanolithography, MEMS, and MOEMS (JM3)*, 2016.

[45] T. Matsunawa, S. Nojima, and T. Kotani, "Automatic layout feature extraction for lithography hotspot detection based on deep neural network," in *Proceedings of SPIE*, 2016.

[46] H. Yang, Y. Lin, B. Yu, and F. E. Young, "Lithography hotspot detection: From shallow to deep learning," in *IEEE International System-on-Chip Conference (SOCC)*, 2017.

[47] H. Yang, L. Luo, J. Su, C. Lin, and B. Yu, "Imbalance aware lithography hotspot detection: A deep learning approach," in *Proceedings of SPIE*, 2017.

[48] H. Yang, J. Su, Y. Zou, B. Yu, and F. E. Young, "Layout hotspot detection with feature tensor generation and deep biased learning," in *ACM/IEEE Design Automation Conference (DAC)*, 2017.

[49] X. Zhu, "Semi-supervised learning literature survey," *Computer Science, University of Wisconsin-Madison*, vol. 2, no. 3, p. 4, 2006.

[50] H. Yang, L. Luo, J. Su, C. Lin, and B. Yu, "Imbalance aware lithography hotspot detection: a deep learning approach," *Journal of Micro/Nanolithography, MEMS, and MOEMS (JM3)*, vol. 16, no. 3, p. 033504, 2017.

[51] B. Settles, "Active learning literature survey," Tech. Rep., 2010.

[52] H. Yang, S. Li, C. Tabery, B. Lin, and B. Yu, "Bridging the gap between layout pattern sampling and hotspot detection via batch active sampling," *arXiv preprint arXiv:1807.06446*, 2018.

[53] Y. Lin, X. Xu, J. Ou, and D. Z. Pan, "Machine learning for mask/wafer hotspot detection and mask synthesis," in *Photomask Technology*, vol. 10451. International Society for Optics and Photonics, 2017, p. 104510A.

[54] X. Xu, Y. Lin, M. Li, T. Matsunawa, S. Nojima, C. Kodama, T. Kotani, and D. Z. Pan, "Sub-Resolution Assist Feature Generation with Supervised Data Learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. PP, no. 99, 2017.

[55] S. Miyama, K. Yamamoto, and K. Koyama, "Large-area optical proximity correction with a combination of rule-based and simulation-based methods," *Japanese Journal of Applied Physics*, vol. 35, no. 12S, p. 6370, 1996.

[56] N. Jia and E. Y. Lam, "Machine learning for inverse lithography: using stochastic gradient descent for robust photomask synthesis," *Journal of Optics*, vol. 12, no. 4, pp. 045 601:1–045 601:9, 2010.

[57] K.-S. Luo, Z. Shi, X.-L. Yan, and Z. Geng, "SVM based layout retargeting for fast and regularized inverse lithography," *Journal of Zhejiang University SCIENCE C*, vol. 15, no. 5, pp. 390–400, 2014.

[58] T. Matsunawa, B. Yu, and D. Z. Pan, "Optical proximity correction with hierarchical bayes model," *Journal of Micro/Nanolithography, MEMS, and MOEMS*, vol. 15, no. 2, pp. 021 009–021 009, 2016.

[59] H. Yang, S. Li, Y. Ma, B. Yu, and E. F. Young, "Gan-opc: mask optimization with lithography-guided generative adversarial nets," in *Proceedings of the 55th Annual Design Automation Conference*. ACM, 2018, p. 131.

[60] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[61] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[62] J.-R. Gao, X. Xu, B. Yu, and D. Z. Pan, "MOSAIC: Mask optimizing solution with process window aware inverse correction," in *ACM/IEEE Design Automation Conference (DAC)*, 2014, pp. 52:1–52:6.

[63] K. Sakajiri, A. Tritchkov, and Y. Granik, "Model-based sraf insertion through pixel-based mask optimization at 32nm and beyond," in *Proceedings of SPIE*, 2008, pp. 702 811–702 811.

[64] R. Viswanathan, J. T. Azpiroz, and P. Selvam, "Process optimization through model based sraf printing prediction," in *Proceedings of SPIE*, 2012, pp. 83 261A–83 261A.

[65] J. Ye, Y. Cao, and H. Feng, "System and method for model-based sub-resolution assist feature generation," Feb. 1 2011, US Patent 7,882,480.

[66] S. D. Shang, L. Swallow, and Y. Granik, "Model-based sraf insertion," Oct. 11 2011, US Patent 8,037,429.

[67] J.-H. Jun, M. Park, C. Park, H. Yang, D. Yim, M. Do, D. Lee, T. Kim, J. Choi, G. Luk-Pat *et al.*, "Layout optimization with assist features placement by model based rule tables for 2x node random contact," in *Proceedings of SPIE*, 2015, pp. 94 270D–94 270D.

[68] C. Kodama, T. Kotani, S. Nojima, and S. Mimotogi, "Sub-resolution assist feature arranging method and computer program product and manufacturing method of semiconductor device," Aug. 19 2014, US Patent 8,809,072.

[69] C. B. Tan, K. K. Koh, D. Zhang, and Y. M. Foong, "Sub-resolution assist feature (sraf) printing prediction using logistic regression," in *Proceedings of SPIE*, 2015, pp. 94 261Y–94 261Y.