Data Efficient Lithography Modeling with Residual Neural Networks and Transfer Learning

Yibo Lin, Yuki Watanabe[†], Taiki Kimura[†], Tetsuaki Matsunawa[†], Shigeki Nojima[†], Meng Li, David Z. Pan
ECE Department, The University of Texas at Austin
Memory Lithography Group, Toshiba Memory Corporation, Yokohama, Japan[†]
{yibolin, alfred, dpan}@cerc.utexas.edu
{yuki9.watanabe, taiki2.kimura, tetsuaki.matsunawa, shigeki.nojima}@toshiba.co.jp[†]

ABSTRACT

Lithography simulation is one of the key steps in physical verification, enabled by the substantial optical and resist models. A resist model bridges the aerial image simulation to printed patterns. While the effectiveness of learning-based solutions for resist modeling has been demonstrated, they are considerably data-demanding. Meanwhile, a set of manufactured data for a specific lithography configuration is only valid for the training of one single model, indicating low data efficiency. Due to the complexity of the manufacturing process, obtaining enough data for acceptable accuracy becomes very expensive in terms of both time and cost, especially during the evolution of technology generations when the design space is intensively explored. In this work, we propose a new resist modeling framework for contact layers that utilizes existing data from old technology nodes to reduce the amount of data required from a target lithography configuration. Our framework based on residual neural networks and transfer learning techniques is effective within a competitive range of accuracy, i.e., 2-10X reduction on the amount of training data with comparable accuracy to the state-of-the-art learning approach.

1. INTRODUCTION

Due to the continuous semiconductor scaling from $10nm$ technology node (N10) to $7nm$ node (N7) [1,2],

the prediction of printed pattern sizes is becoming increasingly difficult and complicated due to the complexity of manufacturing process and variations. However, complex designs demand accurate simulations to guarantee functionality and yield. Resist modeling, as a key component in lithography simulation, is critical to bridge the aerial image simulation to manufactured wafer data. Rigorous simulations that perform physics-level modeling suffer from large computational overhead, which are not suitable when used extensively. Thus compact resist models are widely used in practice.

Figure 1(a) shows the process of lithography simulations where the optical model computes the aerial image from the input mask patterns and the resist model determines the output patterns from this. As the aerial image contains the light intensity map, the resist model needs to determine the slicing thresholds for the output patterns as shown in Figure 1(b). With the thresholds, the critical dimensions (CDs) of printed patterns can be computed, which need to match CDs measured from manufactured patterns. In practice, various factors may impact a resist model such as the physical properties of photoresist, design rules of patterns, process variations.

Accurate lithography simulation like rigorous physics-based simulation is notorious for its long computational time, while simulation with compact models suffers from accuracy issues [3,4]. On the other hand, machine learning techniques are able to construct accurate models and then make efficient predictions. These approaches first take training data to calibrate a model and then use this model to make predictions on testing data for validation. The effectiveness of learning-based solutions has been studied in various lithography related areas including aerial image simulation [5], hotspot detection [6–11], optical proximity correction (OPC) [12–15], sub-resolution assist features (SRAF) [16,17], resist modeling [3,4], etc. In resist modeling, a convolutional neural network (CNN) that predicts slicing thresholds in aerial images is proposed [4]. The neural network consists of three convolution layers and two fully connected layers. Since the slicing threshold is a continuous value, learning a resist model is a regression task rather than a classification task. Around 70% improvement in accuracy is reported compared with calibrated compact models from Mentor Calibre [18]. Shim et al. [3] propose an artificial neural network (ANN) with five hidden layers to predict the height of resist after exposure. Significant speedup is reported with high accuracy compared with a rigorous simulation.

Although the learning-based approaches are able to achieve high accuracy, they are generally data-demanding in model training. In other words, big data is assumed to guarantee accuracy and generality. Furthermore, one data sample can only be used to train the corresponding model under the same lithography configuration, indicating a low data efficiency. Here data efficiency evaluates the accuracy a model can achieve given a specific amount of data, or the amount of data samples are required to achieve target accuracy. Nevertheless, obtaining a large amount of data is often expensive and time-consuming, especially when the technology node switches from one to another and the design space is under active exploration, e.g., from N10 to N7. The lithography configurations including optical sources, resist materials, etc., are frequently changed for experiments. Therefore, a fast preparation of models with high accuracy is urgently desired.

Different from previous approaches, in this work, we assume the availability of large amounts of data from the previous technology generation with old lithography configurations and small amounts of data from a target lithography configuration. We focus on increasing the data efficiency by reusing

ISPD'18, March 25–28, 2018, Monterey, CA
© 2018 ACM. ISBN 978-1-4503-5626-8/18/03…$15.00
DOI: http://dx.doi.org/10.1145/3177540.3178242

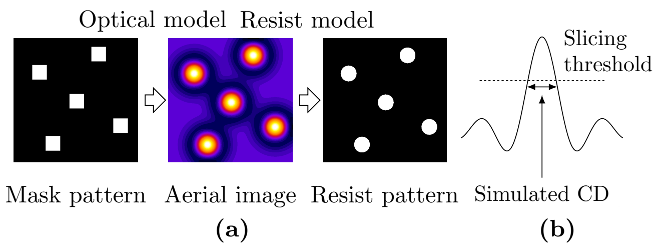

Figure 1: (a) Process of lithography simulation with optical and resist models. (b) Thresholds for aerial image determine simulated CD, which should match manufactured CD.

those from other lithography configurations and transfer the knowledge between different configurations. The objective is to achieve accurate resist models with significantly fewer data to a target configuration. The major contributions are summarized as follows.

- We propose a high performance resist modeling technique based on the residual neural network (ResNet).
- We propose a transfer learning scheme for ResNet that can reduce the amount of data with a target accuracy by utilizing the data from other configurations.
- We explore the impacts from various lithography configurations on the knowledge transfer.
- The experimental results demonstrate 2-10X reduction in the amount of training data to achieve accuracy comparable to the state-of-the-art learning approach [4].

The rest of the paper is organized as follows. Section 2 illustrates the problem formulation. Section 3 explains the details of our approach. The effectiveness of our approach is verified in Section 4 and the conclusion is drawn in Section 5.

2. PRELIMINARIES

In this section, we will briefly introduce the background knowledge on lithography simulation and resist modeling. Then the problem formulation is explained. We mainly focus on contact layers in this work, but our methodology shall be applicable to other layers.

2.1 Lithography Simulation

Lithography simulation is generally composed of two stages, i.e., optical simulation and resist simulation, where optical and resist models are required, respectively. In the optical simulation, an optical model, characterized by the illumination tool, takes mask patterns to compute aerial images, i.e., light intensity maps. Then in the resist simulation, a resist model finalizes the resist patterns with the aerial images from the optical simulation. Generally, there are two types of resist models. One is a variable threshold resist (VTR) model in which the thresholds vary according to aerial images, and the other is a constant threshold resist (CTR) model in which the light intensity is modulated in an aerial image. We adopt the former since it is suitable to learning-based approaches [4].

Figure 2 shows an example of lithography simulation for a clip with three contacts. We assume that proper resolution enhancement techniques (RETs) such as OPC and SRAF have been applied before the computation of the aerial image [19]. The optical simulation generates the aerial image, as shown in Figure 2(b). Resist simulation then computes the thresholds in the aerial image to predict printed patterns. If we consider the horizontal sizes of contacts along the dotted line in Figure 2(c), the light intensity profiling can be extracted from the aerial image along the line and calculates the CDs for each contact

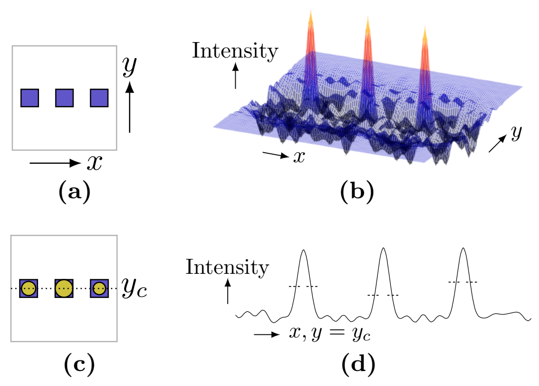

Figure 2: (a) Design target of 3 contacts and (b) the light intensity plot of aerial image. Assume that RETs such as SRAF and OPC have been already applied to the contacts before optical simulation. (c) A dotted line horizontally crosses the centers at $y = y_c$ and the circles denote the contours of printed patterns. (d) Light intensity profiling along the dotted line at $y = y_c$ extracted from the aerial image and different slicing thresholds for each contact.

with the thresholds.

2.2 Historical Data and Transfer Learning

Since the lithography configurations evolve from one generation to another with the advancement of technology nodes, there are plenty of historical data available for the old generation. As mentioned in Section 1, accurate models require a large amount of data for training or calibration, which are expensive to obtain during the exploration of a new generation. If the lithography configurations have no fundamental changes, the knowledge learned from the historical data may still be applicable to the new configuration, which can eventually help to reduce the amount of new data required.

Transfer learning represents a set of techniques to transfer the knowledge from one or multiple source domains to a target domain, utilizing the underlying similarity between the data from these domains. Various studies have explored the effectiveness of knowledge transfer in image recognition and robotics [20–22], while it is not clear whether the knowledge between different resist models is transferable or not.

In this work, we consider the evolution of the contact layer from the cutting edge technology node N10 to N7 [1,2]. A large amount of available N10 data are assumed. During the evolution to N7, different design rules for mask patterns, optical sources and resist materials for lithography are explored. Table 1 shows the lithography configurations considered for N10 and N7. Differences in letters A, B represent different configurations of design rules, optical sources, or resist materials. One configuration for N10 is considered, while two configurations are considered for N7, i.e., N7$_a$, N7$_b$, with two kinds of resist materials (about 20% difference in the slopes of dissolution curves). From N10 to N7, both the design rules and optical sources are changed. For N10, we consider a pitch of 64nm with double patterning lithography, while for N7, the pitch is set to 45nm with triple patterning lithography [1]. The width of each contact is set to half pitch. The lithography target of each contact is set to 60nm for both N10 and N7. Optical sources calibrated with industrial strength for N10 and N7 are shown in Figure 3, with the same type of illumination shapes.

Various combinations of knowledge transfer can be explored from Table 1, such as N10→N7, N7$_i$→N7$_j$, and N10+N7$_i$→N7$_j$, where $i \neq j, i, j \in \{a, b\}$.

Table 1: Lithography Configurations for N10 and N7

	N10	N7	
		N7$_a$	N7$_b$
Design Rule	A	B	B
Optical Source	A	B	B
Resist Material	A	A	B

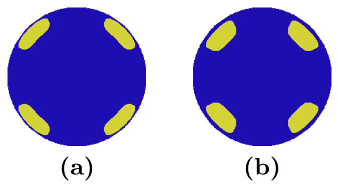

Figure 3: Optical sources (yellow) for (a) N10 and (b) N7.

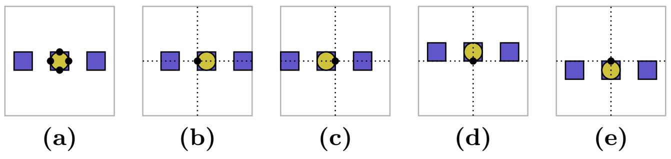

Figure 4: (a) The thresholds for the middle of the 4 edges of the center contact are predicted. (b) (c) (d) (e) The clip window is shifted such that the target position lies in the center of the clip.

2.3 Learning-based Resist Modeling

The thresholds of positions near the contacts are of significant importance since they usually determine the boundaries of printed contacts. Hence we consider the middle of the left, right, bottom and top edges for each contact, as shown in Figure 4(a), where the positions for prediction are highlighted with black dots. In addition, the threshold is mainly influenced by the surrounding mask patterns. Therefore, resist models typically compute the threshold using a clip of mask patterns centered by a target position. To measure the thresholds in Figure 4(a), we select a clip where the target position lies in its center, as shown in Figure 4(b) to Figure 4(e). The task of a resist model is to compute the thresholds for these positions of each contact [4].

Learning-based resist modeling consists of two phases, training and testing. In the training phase, training dataset with both aerial images and thresholds are used to calibrate the model, while in the testing phase, the model predicts thresholds for the aerial images from the testing dataset and compares with the golden thresholds to validate the model.

2.4 Problem Formulation

The accuracy[1] of a model is evaluated with root mean square (RMS) error defined as follows,

$$\epsilon = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}-y)^2}, \tag{1}$$

where N denotes the amount of samples, y denotes the golden values and $\hat{y}$ denotes the predicted values. We further define relative RMS error,

$$\epsilon_r = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\frac{\hat{y}-y}{y})^2}, \tag{2}$$

where a relative ratio of error from the golden values can be represented. Both metrics can refer to errors in either CD or threshold. Although during model training, the RMS error of threshold is generally minimized due to easier computation, the eventual model is often evaluated with the RMS error of CD for its physical meaning to the patterns. The RMS errors in threshold and CD essentially have almost the same fidelity, and usually yield consistent comparison. For convenience, we report relative RMS error in threshold (ϵ_r^{th}) for comparison of different models since it removes the dependency to the scale of thresholds, and use RMS error in CD (ϵ^{CD}) for data efficiency related comparison.

Definition 1 (Data Efficiency). *The amount of target domain data required to learn a model with a given accuracy.*

Given a specific amount of data from a target domain, if one can learn a model with a higher accuracy than another, it also indicates higher data efficiency. Thus improving model accuracy benefits data efficiency as well.

The resist modeling problem is defined as follows.

Problem 1 (Learning-based Resist Modeling). *Given a dataset containing information of aerial images and thresholds at their centers, train a resist model that can maximize the accuracy for the prediction of thresholds.*

In practice, accuracy is not the only objective. The amount of training data should be minimized as well due to the high cost of data preparation. Therefore, we propose the problem of data efficient resist modeling as follows.

Problem 2 (Data Efficient Resist Modeling). *Given datasets from N10 and N7 containing information of aerial images and thresholds, train a resist model for target dataset N7$_i$ that can achieve high accuracy and meanwhile minimize the amount of data required for N7$_i$, where $i \in \{a, b\}$.*

3. ALGORITHMS

In this section, we will explain the structure of our models and then the details regarding the transfer learning scheme.

3.1 Data Preparation

Figure 5 gives the flow of data preparation. We first generate clips and perform SRAF insertion and OPC. The aerial images are then computed from the optical simulation, and at the same time, the golden thresholds need to be computed from either the rigorous simulation or the manufactured data. Each data sample consists of an aerial image and the threshold at its center.

3.1.1 Clip Generation

Following the design rules such as minimum pitch of contacts, we generate three types of $2 \times 2\mu m$ clips. It is necessary to ensure that there is a contact in the center of each clip since that is the target contact for threshold computation.

Contact Array. All possible $m \times n$ arrays of contacts within the dimensions of clips are enumerated. The steps of the arrays can be multiple times of the minimum pitch p, i.e., $p, 2p, 3p, \ldots$, in horizontal or vertical directions. An example of 3×3 contact array with a certain pitch is shown in Figure 6(a). It needs to mention that the same 3×3 contact

[1]Note that the accuracy we talk about in this paper refers to the accuracy at end of lithography flow including all RETs.

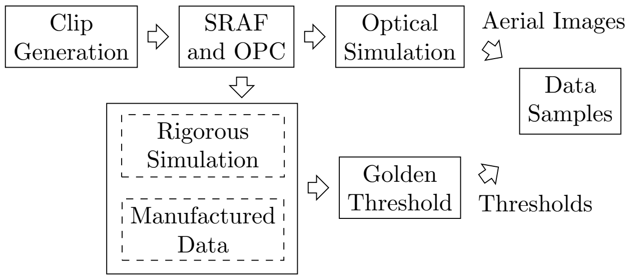

Figure 5: Flow of data preparation.

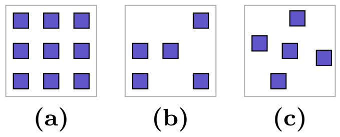

Figure 6: (a) A clip of 3×3 contact array. (b) A clip of 3×3 randomized contact array. (c) A clip of contacts with random positions.

array with different steps should be regarded as different clips due to discrepant spacing.

Randomized Contact Array. The aforementioned contact arrays essentially distribute contacts on grids and fill all the slots in the grid maps. The randomization of contact arrays is implemented by a random distribution of contacts in those grid maps. Fig 6(b) shows an example of randomized contact array from the 3×3 contact array in Figure 6(a). Various distribution of contacts can be generated even from the same grid maps.

Contacts with Random Positions. Contacts in this type of clips do not necessarily align to any grid map, as their positions are randomly generated, while the design rules are still guaranteed. An example is shown in Figure 6(c). No matter how the surrounding contacts change, the contact in the center of the clip should remain the same.

3.1.2 Data Augmentation

Due to the symmetry of optical sources in Figure 3, data can be augmented with rotation and flipping, improving the data efficiency [23]. Eight combinations of rotation and flipping are shown in Figure 7, where new data samples are obtained without new thresholds. Data augmentation inflates datasets to obtain models with better generalization.

3.2 Convolutional Neural Networks

Convolutional neural networks (CNN) have demonstrated impressive performance on mask related applications in lithography such as hotspot detection, and resist modeling [4, 9]. The structure of CNN mainly includes convolution layers and fully connected layers. Features are extracted from convolution layers and then classification or regression is performed by fully connected layers. Figure 10(a) illustrates a CNN structure with three convolution layers and two fully connected layers [4]. The first convolution layer has 64 filters with dimensions of 7×7. Although not explicitly shown most of the time, a rectified linear unit (ReLU) layer for activation is applied immediately after the convolution layer, where the ReLU function is defined as,

$$x^l = \begin{cases} x^{l-1}, & \text{if } x^{l-1} \geq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Then the max-pooling layer performs down-sampling with a factor of 2 to reduce the feature dimensions and improve the invariance to translation [23]. After three convolution layers, two fully connected layers are applied where the first one has 256 hidden units followed with a ReLU layer and a 50% dropout layer, and second one connects to the output.

3.3 Residual Neural Networks

One way to improve the performance of CNN is to increase the depth for a larger capacity of the neural networks. However, the counterintuitive degradation of training accuracy in CNN is observed when stacking more layers, preventing the neural networks from better performance [24]. An example of CNNs with 5 and 10 layers is shown in Figure 8, where the deeper CNN fails to converge to a smaller training error than the shallow one due to gradient vanishing [25, 26], eventually resulting in the failure to achieve a better testing error either. The study from He et al. [24] reveals that the underlying reason comes from the difficulty of identity mapping. In other words, fitting a hypothesis $\mathcal{H}(x) = x$ is considerably difficult for solvers to find optimal solutions. To overcome this issue, residual neural networks (ResNet), which utilizes shortcut connections, are adopted to assist the convergence of training accuracy.

The building block of ResNet is illustrated in Figure 9, where a shortcut connection is inserted between the input and output of two convolution layers. Let the function $\mathcal{F}(x)$ be the mapping defined by the two convolution layers. Then the entire function for the building block becomes $\mathcal{F}(x) + x$. Suppose the building block targets to fit the hypothesis $\mathcal{H}(x)$. The residual networks train $\mathcal{F}(x) = \mathcal{H}(x) - x$, while the convolution layers without shortcut connections like that in CNN try to directly fit $\mathcal{F}(x) = \mathcal{H}(x)$. Theoretically, if $\mathcal{H}(x)$ can be approximated with $\mathcal{F}(x)$, then it can also be approximated with $\mathcal{F}(x)+x$. Despite the same nature, comprehensive experiments have demonstrated a better convergence of ResNet than that of CNN for deep neural networks [24]. We also observe a better performance of ResNet with the transfer learning schemes than that of CNN in our problem, which has never been explored before.

The ResNet is shown in Figure 10(b) with 8 convolution layers and 2 fully connected layers. Different from the original setting [24], we add a shortcut connection to the first convolution layer by broadcasting the input tensor of $64 \times 64 \times 1$ to $64 \times 64 \times 64$. This minor change enables better empirical results in our problem. For the rest of the networks, 3 building blocks for ResNet are utilized.

3.4 Transfer Learning

Transfer learning aims at adapting the knowledge learned from data in source domains to a target domain. The transferred knowledge will benefit the learning in the target domain with a faster convergence and better generalization [23]. Suppose the data in the source domain has a distribution P_s and that in the target domain has a distribution P_t. The un-

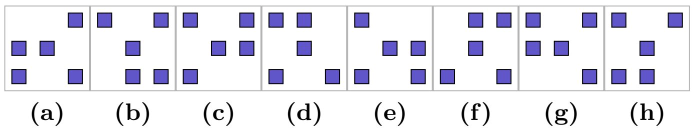

Figure 7: Combinations of rotation and flipping. (a) Original. (b) Rotate 90°. (c) Rotate 180°. (d) Rotate 270°. (a) Flip. (b) Flip and rotate 90°. (c) Flip and rotate 180°. (d) Flip and rotate 270°.

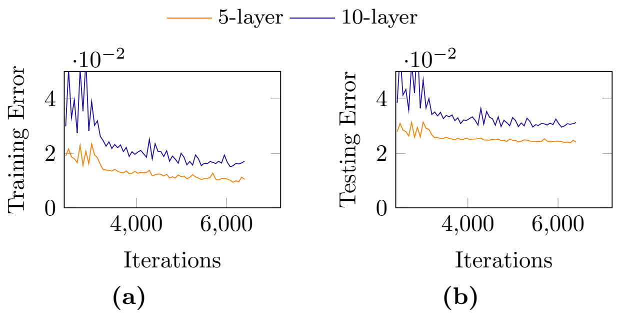

Figure 8: Counterintuitive (a) training and (b) testing errors for different depth of CNN with epochs.

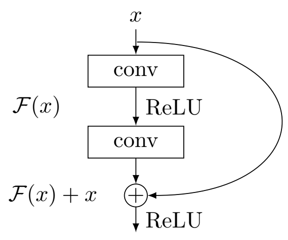

Figure 9: Building block of ResNet.

derlying assumption of transfer learning lies in the common factors that need to be captured for learning the variations of P_s and P_t, so that the knowledge for P_s is also useful for P_t. An intuitive example is that learning to recognize cats and dogs in the source task helps the recognition of ants and wasps in the target task, especially when the source task has significantly larger dataset than that of the target task. The reason comes from the low-level notions of edges, shapes, etc., shared by many visual categories [23]. In resist modeling, different lithography configurations can be viewed as separate tasks with different distributions.

Typical transfer learning scheme for neural networks fixes the first several layers of the model trained for another domain and finetune the successive layers with data from the target domain. The first several layers usually extract general features, which are considered to be similar between the source and the target domains, while the successive layers are classifiers or regressors that need to be adjusted. Figure 11 shows an example of the transfer learning scheme. We first train a model with source domain data and then use the source domain model as the starting point for the training of the target domain. During the training for the target domain, the first k layers are fixed, while the rest layers are finetuned. We denote this scheme as TF_k, shortened from "Transfer and Fix", where k is the parameter for the number of fixed layers.

4. EXPERIMENTAL RESULTS

Our framework is implemented with Tensorflow [27] and validated on a Linux server with 3.4GHz Intel i7 CPU and Nvidia GTX 1080 GPU. Around 980 mask clips are generated according to Section 3.1 for N10 and N7 separately following the design rules in Section 2.2, respectively. N7$_a$ and N7$_b$ use the same set of clips, but different lithography configurations. SRAF, OPC and aerial image simulation are performed with Mentor Calibre [18]. The golden CD values are obtained from rigorous simulation using Synopsys Sentaurus Lithography models [28] calibrated from manufactured data for N10, N7$_a$, and N7$_b$ according to Table 1. Then golden thresholds are extracted. Each clip has four thresholds as shown in Figure 4. Hence the N10 dataset contains 3928 samples and each N7 dataset contains 3916 samples, respectively. The data aug-

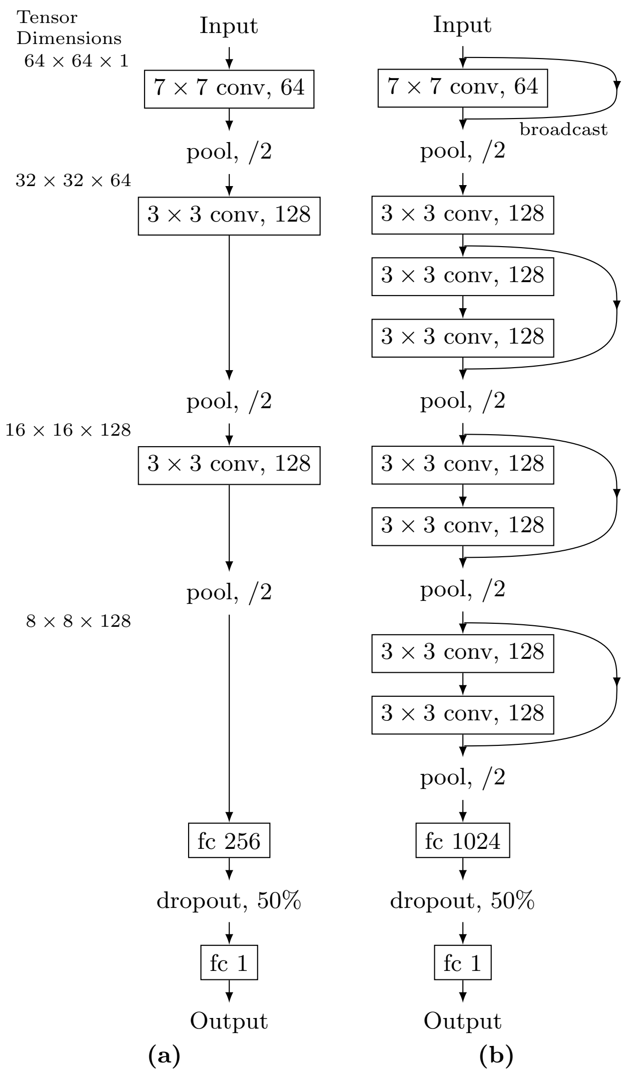

Figure 10: (a) CNN and (b) ResNet structure.

mentation technique in Section 3.1.2 is applied, so the training set and the testing set will be augmented by a factor of 8 independently. For example, if 50% of the data for N10 are used for training, then there are $3928 \times 50\% \times 8 = 15712$ samples. It needs to mention that always the same 50% portions are used during the validation of a dataset for fair comparison of different techniques. The batch size is set to 32 for training accommodating to the large variability in the sizes of training datasets. Adam [29] is used as the stochastic optimizer and maximum epoch is set to 200 for training.

The training time for one model takes 10 to 40 minutes according to the portions of a dataset used for training, and prediction time for an entire N10 or N7 dataset takes less than 10 seconds, while the rigorous simulation takes more than 15 hours for each N10 or N7 dataset. Thus we no longer report runtime since the prediction time is negligible compared with that of the rigorous simulation. Each experiment runs 10 different random seeds and report the average numbers.

4.1 CNN and ResNet

We first compare CNN and ResNet in Figure 12(a). Column "CNN-5" denotes the network with 5 layers shown in Figure 10(a). Column "CNN-10" denotes the one with 10 layers that has the same structure as that in Figure 10(b) but with-

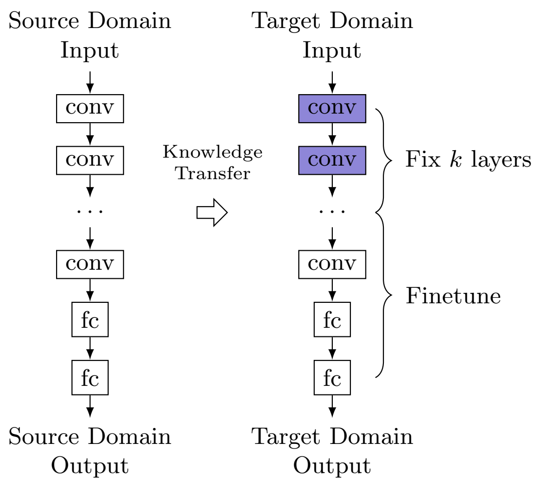

Figure 11: Transfer learning scheme with the first k layers fixed when training for target domain, denoted as TF_k.

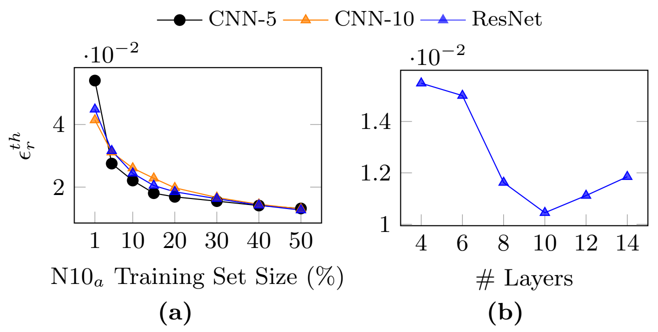

Figure 12: (a) Comparison on testing accuracy of CNN-5, CNN-10, and ResNet on N10. (b) Impact of depth on the testing accuracy of ResNet.

out shortcut connections. Column "ResNet" denotes the one with 10 layers shown in Figure 10(b). When using 1% to 20% training data, ResNet shows better average relative RMS error ϵ_r^{th} than CNN-10, but CNN-5 provides the best error. We will show later that ResNet on the contrary outperforms CNN-5 when transfer learning is incorporated.

The impacts of depth on the performance of ResNet are further explored in Figure 12(b), where we gradually stack more building blocks in Figure 9 before fully connected layers. The x-axis denotes total number of convolution and fully connected layers corresponding to different numbers of building blocks. For instance, 0 building block leads to 4 layers and 3 building blocks result in 10 layers (Figure 10(b)). The testing error decreases to lowest value at 10 layers and then starts to increase, indicating potential overfitting afterwards [23]. Therefore, we use 10 layers for the ResNet in the experiment.

4.2 Knowledge Transfer From N10 to N7

We then compare the testing accuracy between knowledge transfer from N10 to N7 and directly training from N7 datasets in Figure 13(a). In this example, the x-axis represents the percentage of training dataset for the target domain N7$_a$, while the percentage of data from the source domain N10 is always 50%. Similar trends are also observed for N7$_b$. Curve "CNN" denotes training the CNN of 5 layers in Figure 10(a) with data from target domain only, i.e., no transfer learning involved. Curve "CNN TF$_0$" denotes the transfer learning scheme in Section 3.4 for the same CNN with zero layer fixed. Curve

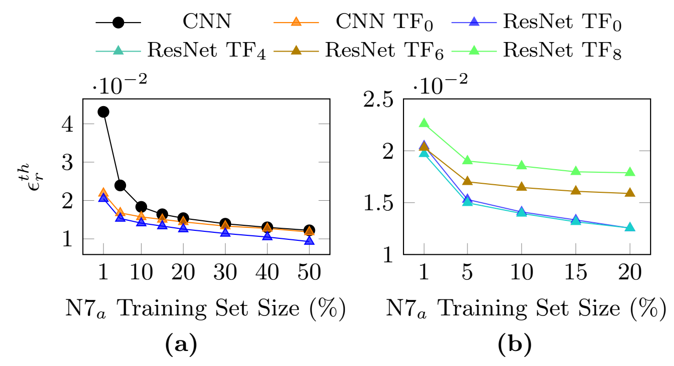

Figure 13: Testing accuracy of transfer learning from N10 to N7$_a$. (a) Comparison between CNN and transfer learning. (b) Comparison between transfer learning schemes where different numbers of layers are fixed.

"ResNet TF$_0$" denotes applying the same scheme to ResNet. The most significant benefit of transfer learning comes from small training dataset with a range of 1% to 20%, where there are around 52% to 18% improvement in the accuracy from CNN. Meanwhile, ResNet TF$_0$ can achieve an overall average of 13% smaller error than CNN TF$_0$.

Figure 13(b) further compares the results of fixing different numbers of layers during transfer learning. In this case, ResNet TF$_0$ and ResNet TF$_4$ have the best accuracy, while the error increases with more layers fixed. It is indicated that the tasks N10 and N7 are quite different and both feature extraction layers and regression layers need finetuning.

4.3 Knowledge Transfer within N7

The transfer learning between different N7 datasets, e.g., from N7$_a$ to N7$_b$, is also explored in Figure 14. The x-axis represents the percentage of training dataset for the target domain N7$_b$, while the percentage of data from the source domain N7$_a$ is always 50%. Compared with the knowledge transfer from N10 to N7, we achieve even higher accuracy between 1% and 20% training datasets in Figure 14(a). For example, with 1% training dataset, there is around 65% improvement in accuracy from CNN, and with 20% training dataset, the improvement is around 23%. ResNet TF$_0$ keeps having lower errors than that of CNN TF$_0$ as well, where the average benefit is around 15%.

The curves in Figure 14(b) show different insights from that of the knowledge transfer from N10 to N7. The accuracy of ResNet TF$_0$ can be further improved with more layers fixed, e.g., ResNet TF$_8$, by around 28% to 14%. This is reasonable since N7$_a$ and N7$_b$ have the same design rules and illumination shapes, and the only difference lies in the resist materials. Therefore, the feature extraction layers are supposed to remain almost the same. With the sizes of the training dataset increasing to 15% and 20%, the differences in the accuracy become smaller, because there are enough data to find good configurations for the networks.

4.4 Impact of Various Source Domains

In transfer learning, the correlation between the datasets of source and target domains is critical to the effectiveness of knowledge transfer. Thus, we explore the impacts of source domain datasets on the accuracy of modeling for the target domain. Figure 15 plots the testing errors of learning N7$_b$ using ResNet TF$_0$ with various source domain datasets. Curves "N10$^{50\%}$" and "N7$_a^{50\%}$" indicate that 50% of the N10 or the N7$_a$ dataset is used to train source domain models, respec-

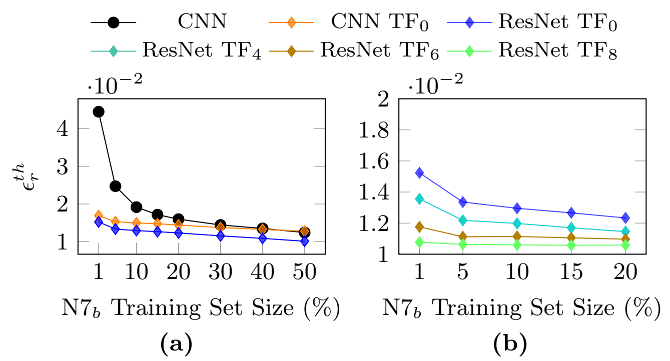

Figure 14: Testing accuracy of transfer learning from N7$_a$ to N7$_b$. (a) Comparison between CNN and transfer learning. (b) Comparison between transfer learning schemes where different numbers of layers are fixed.

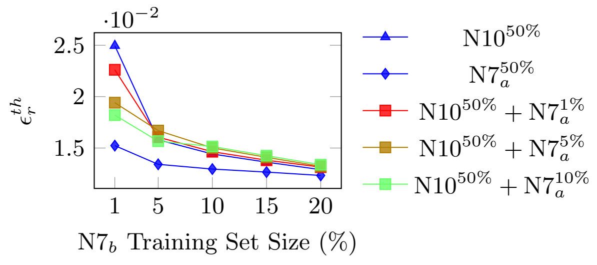

Figure 15: Testing accuracy of ResNet TF$_0$ for N7$_b$ from different source domain datasets.

tively. Curve "N10$^{50\%}$ + N7$_a^{1\%}$" describes the situation where we have 50% of the N10 dataset and 1% of the N7$_a$ dataset for training. In this case, as shown in Figure 16, we first use the 50% N10 data to train the first source domain model; then train the second source domain model using the first model as the starting point with the 1% N7$_a$ data; in the end, the target domain model for N7$_b$ is trained using the second model as the starting point with N7$_b$ data. Curves "N10$^{50\%}$ + N7$_a^{5\%}$" and "N10$^{50\%}$ + N7$_a^{10\%}$" are similar, simply with different amounts of N7$_a$ data for training.

The knowledge from N7$_a^{50\%}$ is the most effective for N7$_b$ due to the minor difference in resist materials between two datasets. For the rest curves, the accuracy of N10$^{50\%}$ + N7$_a^{5\%}$ and N10$^{50\%}$ + N7$_a^{10\%}$ is in general better than or at least comparable to that of N10$^{50\%}$. This indicates that having more data from closer datasets to the target dataset, e.g., N7$_a$, is still helpful.

4.5 Improvement in Data Efficiency

Table 2 presents the accuracy metrics, i.e., relative threshold RMS error (ϵ_r^{th}) and CD RMS error (ϵ^{CD}), for learning N7$_b$ from various source domain datasets. Since we consider the data efficiency of different learning schemes, we focus on

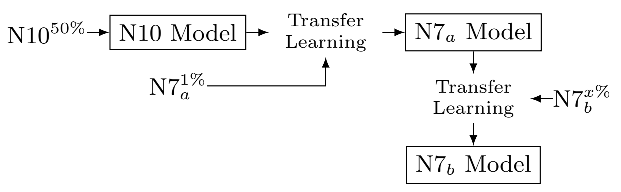

Figure 16: Transfer learning from 50% of N10 dataset and 1% of N7$_a$ dataset (i.e., N10$^{50\%}$ + N7$_a^{1\%}$) to N7$_b$ with x% of N7$_b$ dataset.

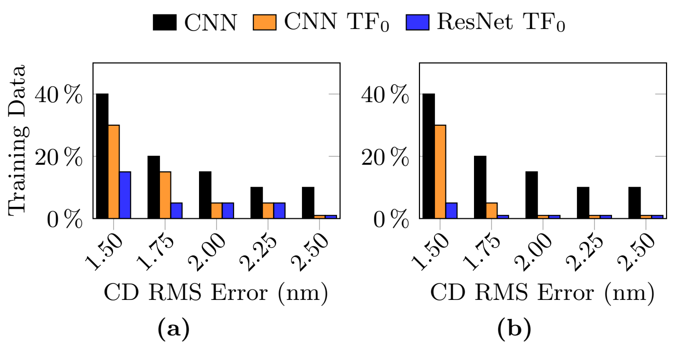

Figure 17: Amount of training data required for N7$_b$ given target CD RMS errors when (a) 50% N10 dataset is available or (b) 50% N7$_a$ dataset is available.

the small training dataset for N7$_b$, from 1% to 20%. Situations such as no source domain data ($\emptyset$), only source domain data from N10 (N10$^{50\%}$), only source domain data from N7$_a$ (N7$_a^{50\%}$), and combined source domain datasets, are examined. As mentioned in Section 2, the fidelity between relative threshold RMS error and CD RMS error is very consistent, so they share almost the same trends. Transfer learning with any source domain dataset enables an average improvement of 23% to 40% from that without knowledge transfer. In small training datasets of N7$_b$, ResNet also achieves around 8% better performance on average than CNN in the transfer learning scheme. At 1% of N7$_b$, combined source domain datasets have better performance compared with that of N10$^{50\%}$ only, but the benefits vanish with the increase of the N7$_b$ dataset.

In real manufacturing, models are usually calibrated to satisfy a target accuracy or target CD RMS error. Figure 17 demonstrates the amount of training data required in the target domain for learning the N7$_b$ model. Curve "CNN" does not involve any knowledge transfer, while curves "CNN TF$_0$" and "ResNet TF$_0$" utilize transfer learning in CNN and ResNet, respectively. The curves in Fig 17(a) assume the availability of N10 data. Consider the CD RMS error from 1.5nm to 2.5nm, which is around 10% of the half pitch for N7 contacts. This range of accuracy is also comparable to that of the state-of-the-art CNN [4]. ResNet TF$_0$ requires significantly fewer data than both CNN and CNN TF$_0$. For instance, when the target CD error is 1.75nm, ResNet TF$_0$ demands 5% training data from N7$_b$, while CNN requires 20% and CNN TF$_0$ requires 15%. Figure 17(b) considers the transfer from N7$_a$ to N7$_b$. Both ResNet TF$_0$ and CNN TF$_0$ only require 1% training data from N7$_b$ for most target CD RMS errors, where CNN TF$_0$ cannot achieve the accuracy unless given 30% data. Overall, ResNet TF$_0$ can achieve 2-10X reduction of training data within this range compared with CNN. It needs to mention that 1% of dataset only correspond to fewer than 40 samples owing to the data augmentation, indicating only thresholds of 40 clips are required.

5. CONCLUSION

A transfer learning framework based on residual neural networks is proposed for resist modeling. The combination of ResNet and transfer learning is able to achieve high accuracy with very few data from the target domains, under various situations for knowledge transfer, indicating high data efficiency. Extensive experiments demonstrate that the proposed techniques can achieve 2-10X reduction according to various requirements of accuracy comparable to the state-of-the-art learning approach. It is also shown that the performance of

Table 2: Relative Threshold RMS Error and CD RMS Error for $N7_b$ with Different Source Domain Datasets

Source Datasets		$\emptyset$		$N10^{50\%}$				$N7_a^{50\%}$				$N10^{50\%} + N7_a^{5\%}$		$N10^{50\%} + N7_a^{10\%}$	
Neural Networks		CNN		CNN TF_0		ResNet TF_0		CNN TF_0		ResNet TF_0		ResNet TF_0		ResNet TF_0	
		ϵ_r^{th} (10^{-2})	ϵ^{CD}	ϵ_r^{th} (10^{-2})	ϵ^{CD}	ϵ_r^{th} (10^{-2})	ϵ^{CD}	ϵ_r^{th} (10^{-2})	ϵ^{CD}	ϵ_r^{th} (10^{-2})	ϵ^{CD}	ϵ_r^{th} (10^{-2})	ϵ^{CD}	ϵ_r^{th} (10^{-2})	ϵ^{CD}
$N7_b$	1%	4.44	4.76	2.34	2.48	2.29	2.39	1.69	1.79	1.52	1.60	1.94	2.03	1.82	1.91
	5%	2.78	2.96	1.73	1.86	1.60	1.70	1.53	1.64	1.34	1.43	1.67	1.78	1.57	1.67
	10%	1.92	2.04	1.63	1.76	1.47	1.57	1.50	1.60	1.30	1.38	1.50	1.60	1.51	1.61
	15%	1.72	1.84	1.56	1.68	1.39	1.47	1.48	1.55	1.27	1.35	1.41	1.50	1.43	1.52
	20%	1.60	1.71	1.50	1.61	1.31	1.39	1.44	1.55	1.23	1.31	1.32	1.41	1.34	1.43
ratio		1.00	1.00	0.77	0.77	0.70	0.69	0.69	0.69	0.60	0.60	0.69	0.69	0.69	0.68

transfer learning differs from dataset to dataset and is worth exploring to see the correlation between datasets. Examining the quantitative relation between the correlation of datasets and performance of transfer learning is valuable in the future.

Acknowledge

This project is supported in part by Toshiba Memory Corporation, NSF, and the University Graduate Continuing Fellowship from the University of Texas at Austin. The authors would like to thank Memory Lithography Group from Toshiba Memory Corporation for helpful discussions and feedback.

6. REFERENCES

[1] L. Liebmann, A. Chu, and P. Gutwin, "The daunting complexity of scaling to 7nm without EUV: Pushing DTCO to the extreme," in *Proceedings of SPIE*, vol. 9427, 2015.

[2] L. Liebmann, J. Zeng, X. Zhu, L. Yuan, G. Bouche, and J. Kye, "Overcoming scaling barriers through design technology cooptimization," in *VLSI Technology, 2016 IEEE Symposium on*. IEEE, 2016, pp. 1–2.

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

[4] 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.

[5] X. Ma, X. Zhao, Z. Wang, Y. Li, S. Zhao, and L. Zhang, "Fast lithography aerial image calculation method based on machine learning," *Applied Optics*, vol. 56, no. 23, pp. 6485–6495, 2017.

[6] J.-Y. Wuu, F. G. Pikus, and M. Marek-Sadowska, "Efficient approach to early detection of lithographic hotspots using machine learning systems and pattern matching," in *SPIE Advanced Lithography*. International Society for Optics and Photonics, 2011, pp. 79 740U–79 740U.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] A. Gu and A. Zakhor, "Optical proximity correction with linear regression," *IEEE Transactions on Semiconductor Manufacturing (TSM)*, vol. 21, no. 2, pp. 263–271, 2008.

[13] 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, p. 045601, 2010.

[14] R. Luo, "Optical proximity correction using a multilayer perceptron neural network," *Journal of Optics*, vol. 15, no. 7, p. 075708, 2013.

[15] 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.

[16] 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.

[17] 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.

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

[19] L. W. Liebmann, S. M. Mansfield, A. K. Wong, M. A. Lavin, W. C. Leipold, and T. G. Dunham, "Tcad development for lithography resolution enhancement," *IBM Journal of Research and Development*, vol. 45, no. 5, pp. 651–665, 2001.

[20] J. P. Hanna and P. Stone, "Grounded action transformation for robot learning in simulation." in *AAAI*, 2017, pp. 3834–3840.

[21] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[22] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

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

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: https://www.tensorflow.org

[28] "Synopsys Sentaurus Lithography," https://www.synopsys.com/silicon/mask-synthesis/sentaurus-lithography.html.

[29] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.