LithoGAN: End-to-End Lithography Modeling with Generative Adversarial Networks

Wei Ye
ECE Department, UT Austin
weiye@utexas.edu

Mohamed Baker Alawieh
ECE Department, UT Austin
mohdbaker@utexas.edu

Yibo Lin
ECE Department, UT Austin
yibolin@utexas.edu

David Z. Pan
ECE Department, UT Austin
dpan@ece.utexas.edu

ABSTRACT

Lithography simulation is one of the most fundamental steps in process modeling and physical verification. Conventional simulation methods suffer from a tremendous computational cost for achieving high accuracy. Recently, machine learning was introduced to trade off between accuracy and runtime through speeding up the resist modeling stage of the simulation flow. In this work, we propose LithoGAN, an end-to-end lithography modeling framework based on a generative adversarial network (GAN), to map the input mask patterns directly to the output resist patterns. Our experimental results show that LithoGAN can predict resist patterns with high accuracy while achieving orders of magnitude speedup compared to conventional lithography simulation and previous machine learning based approach.

1 INTRODUCTION

Lithography holds a fundamental position in today's semiconductor manufacturing [1]. It transfers a designed mask pattern into a resist pattern on the top surface of a semiconductor wafer [2, 3]. In order to bypass the cost-intensive and time-consuming experimental verification, the semiconductor industry has relied on lithography simulation for process development and performance verification [4, 5]. However, the steady decrease of the feature sizes along with the growing complexity and variation of the manufacturing process have tremendously increased the lithography modeling complexity and prolonged the already-slow simulation procedure.

Lithography simulation mainly falls into two categories: physics-level rigorous simulation and compact model-based simulation. Rigorous simulation precisely simulates the physical effects of materials to obtain the printed patterns [6, 7]. In practice, the physical properties of photoresist (resist) and optical systems, the mask patterns, and the process variations are all correlated to the printing. As a rigorous model has to include these cross-related quantities, it is computationally expensive. Also, the calibration of lithography models can take several weeks at advanced technology nodes [8]. In VLSI manufacturing, modeling efficiency is crucial for fast design closure along with modeling accuracy. Therefore, compact models stand as a speedup alternative to rigorous computation with a small sacrifice in accuracy.

DAC '19, June 2–6, 2019, Las Vegas, NV, USA

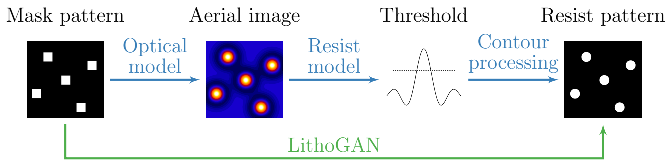

Figure 1: Conventional lithography simulation flow consisting of multiple stages and the proposed LithoGAN flow.

Figure 1 shows a typical flow of lithography simulation. First, an aerial image is generated from a mask pattern using an optical model which is characterized by the illumination type and projection lenses of an exposure tool. Then a resist model is used to determine the locally varying slicing thresholds [9]. Lastly, the thresholds are processed through extrapolation together with the corresponding aerial image to evaluate the critical dimension (CD) of the printed patterns or to generate the resist contours.

Although conventional variable threshold resist (VTR) models are highly efficient, they fail to keep up their accuracy at advanced technology nodes [10]. To improve simulation quality, machine learning based techniques have been proposed to construct accurate and efficient resist models [10–13]. These approaches first take a set of training data to train (calibrate) a model and then use this model to make predictions on test data. [11] proposes an artificial neural network (ANN) to predict the height of resist after exposure. However, efforts are spent on determining the appropriate set of features for model training. To overcome the explicit feature extraction, [10] proposes a convolutional neural network (CNN) model that predicts the slicing thresholds in aerial images accurately. Recently, [12] proposed a transfer learning scheme together with an active learning approach to cope with the deficiency in the manufacturing data at advanced technology nodes.

Nevertheless, several drawbacks exist in the mainstream compact models and machine learning approaches. The proposed resist models rely on optical simulation to generate aerial images, which are accompanied by a high computational cost. Additionally, only resist height or slicing threshold is predicted from the proposed models, which requires further processing to finalize the contour patterns. Hence, the state-of-the-art lithography modeling techniques still suffer from an exorbitant computational cost while providing partial modeling schemes that rely heavily on pre- and post-processing procedures.

In spite of various rigorous models and compact models at hand, it is extremely desirable to further improve lithography modeling efficiency without compromising much accuracy. Considering the fact that machine learning based approaches have demonstrated superior efficacy in a particular stage during lithography modeling, a natural question then arises: is it possible to build an end-to-end lithography model with machine learning techniques? Toward this goal, we propose *LithoGAN*, a novel lithography modeling framework based on conditional generative adversarial network (CGAN) that has demonstrated tremendous success in computer vision over the past few years [14–18]. CGAN manifests itself among numerous generative models with an inherent capability to perform image translation tasks such as image colorization and background masking, where an image in one domain is mapped to a corresponding image in another domain. In addition, CGAN has been adopted for optical proximity correction (OPC) enhancement in IC manufacturing [19].

Our proposed LithoGAN framework is the first complete end-to-end lithography modeling approach mapping the mask pattern at one end to the resist pattern at the other. This approach builds on a CGAN to *translate* an image from the layout to the resist shape. It turns out that this translation can achieve high accuracy in predicting the shape and size of the resist pattern. Moreover, to further boost the performance of the CGAN, LithoGAN integrates a CNN that can predict the pattern center to help with localization. The major contributions of this paper are highlighted as follows.

- The end-to-end lithography modeling problem is formulated as an image translation task, which maps mask patterns to resist patterns directly without running optical or resist simulation.
- The proposed framework is based on a conditional generative adversarial network, paired with a convolutional neural network to achieve both high accuracy and efficiency.
- Our framework can achieve ~1800× runtime reduction compared to rigorous simulation and ~190× compared to previous approaches with machine learning based threshold prediction [10, 12].
- Experimental results demonstrate our framework achieves comparable accuracy to the state-of-the-art work [12] which requires optical simulation and contour processing.

The rest of this paper is organized as follows. Section 2 reviews the basic concepts and gives the problem formulation. Section 3 provides a detailed explanation of the proposed LithoGAN framework. Section 4 demonstrates the effectiveness of our approaches with comprehensive results, followed by the conclusion in Section 5.

2 PRELIMINARIES

An accurate end-to-end lithography model should produce patterns consistent with the manufactured (golden) ones. In order to evaluate the accuracy of a model, evaluation metrics are required to quantify the critical mismatches. Edge placement error (EPE) is a commonly used metric in lithography to characterize pattern fidelity [2, 20]. Technically, EPE measures the Manhattan distances between the printed resist contours and the intended mask patterns at given measurement points. However, our focus is to measure the performance of the proposed LithoGAN framework where we expect a well-trained model to produce contours similar to the golden contours. In other words, the objective is not to optimize EPE, but rather to mimic the golden contours obtained from rigorous simulation. Hence, we propose a new measure, denoted as edge displacement error, which is tailored to our problem.

Definition 1 (Edge Displacement Error, EDE)**.** Given the bounding boxes of the golden and predicted contours respectively, the edge displacement error for a given edge in the bounding box is defined as the distance between the golden edge and the predicted one.

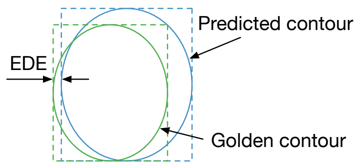

Figure 2: An illustration of the EDE evaluation metric.

The definition of EDE is very similar to EPE, except that EDE is defined between two contours, while EPE is defined between a contour and a design target. Figure 2 illustrates how EDE measures the edge distance between the model predicted contour and the golden lithography contour. However, this measure is not effective in capturing the details of the mismatch between the two contours. While evaluating the quality of the contours is still an open problem, we introduce additional metrics to provide a comprehensive evaluation. Considering that the essence of the LithoGAN task is to predict the color of each pixel in a monochrome image, we adopt the metrics commonly used in computer vision tasks such as semantic segmentation [21].

In this work, three metrics are used to evaluate the quality of the synthesized image besides the EDE metric. For the generality of the terminology, we use class i to represent color i of a pixel in the following discussions. Let $p_{i,j}$ be the number of pixels of class i predicted to belong to class j, where $i, j \in \{0, 1\}$. Let $t_i = \sum_j p_{i,j}$ be the total number of pixels of class i.

Definition 2 (Pixel Accuracy)**.** Pixel accuracy is defined as the percentage of pixels in the image which are correctly classified, $(\sum_i p_{i,i})/(\sum_i t_i)$.

Definition 3 (Class Accuracy)**.** Class accuracy is defined as the average percentage of pixels in the image which are correctly classified for each class, $\frac{1}{2}\sum_i (p_{i,i}/t_i)$.

Definition 4 (Mean IoU)**.** Intersection over union (IoU) measures the number of pixels present in both the golden and predicted patterns (intersection) divided by the number of all pixels present in either of them (union). Mean IoU is an average of the IoU scores for all classes, $\frac{1}{2}\sum_i (p_{i,i}/(t_i - p_{i,i} + \sum_j p_{j,i}))$.

The proposed lithography modeling framework first builds a CGAN model using a set of layout clip pairs, where each pair includes a mask pattern and a resist pattern of the center contact as shown in Figure 3(a) and Figure 3(b) respectively. We define the CGAN-based end-to-end lithography modeling problem as follows.

Problem 1 (End-to-End Lithography Modeling)**.** Given a dataset containing the pairs of mask patterns and corresponding resist patterns of center contacts, the objective of end-to-end lithography modeling is to train a model that can accurately predict the resist pattern of the center contact based on a given mask pattern.

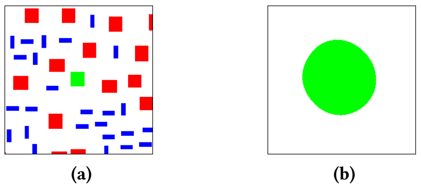
Figure 3: (a) Mask pattern and (b) resist pattern of target contact. Green rectangle denotes the center contact after OPC; red rectangles represent other contacts after OPC; blue rectangles denote the SRAFs.

3 LITHOGAN FRAMEWORK

3.1 Data Preparation

For training the proposed framework, a dataset consisting of paired images corresponding to mask patterns and resist patterns is needed. Proper resolution enhancement techniques (RETs) such as sub-resolution assist feature (SRAF) generation and OPC have been applied to the original input mask clips of size $2 \times 2\,\mu$m. Towards a better localization around the target contact, these clips are then cropped to $1 \times 1\,\mu$m such that, in each clip, the target contact is located exactly at the center of the clip.

The obtained clips are converted to RGB images of size 256×256 pixels where the target contact of interest is encoded into the green channel, neighboring contacts are encoded into the red channel, and SRAFs are encoded into the blue channel. This coloring scheme, demonstrated by the example in Figure 3(a), maps the different types of objects to different colors to help the model discriminate these objects during the learning and inference processes. On the other hand, the target contact is designed to be 60×60 nm; hence, we use the window size 128×128 nm to crop the golden resist pattern of the target contact. Although synthesizing a 128×128 image might be enough for generating the pattern, the cost of misprediction could be high. For example, mispredicting 1 pixel may result in 1 nm error to the contour, hence, imposing an extremely high requirement to the model. Therefore, we scale the 128×128 nm clip to a monochrome image of size 256×256 pixels as in Figure 3(b) such that error from mispredicting 1 pixel is around 0.5 nm. Further improvement to the accuracy is possible by scaling the clip to larger images, but it may cause additional overhead in the modeling effort.

3.2 CGAN Architecture Design

GANs are deep neural networks that use a training dataset to learn the distribution of the input, typically images, and generate new images from the learned distribution. At the highest level, GANs consist of two networks that compete with each other: a generator and a discriminator [14]. The generator G generates fake samples to fool the discriminator, while the adversarially trained discriminator D distinguishes between real images and fake images generated by the generator. The competition throughout the training process drives both to improve: the discriminator guides the generator on what images to create, while also improving itself by learning what distinguishes real images from the fake ones from the generator.

At the end of the training process, the generator learns the distribution of the training data and is eventually able to generate real-looking images. On the other hand, it is hard for the discriminator to distinguish between training set images and generated images. After the GAN model converges, the role of the discriminator is over, and the main interest is in the generator who is now able to generate high-quality images. In this way, a GAN learns a generative model that maps a random noise vector $\mathbf{z}$ to output image $\widehat{\mathbf{y}}$: $\widehat{\mathbf{y}} = G(\mathbf{z})$.

Unlike the aforementioned unconditional GAN, the goal of a CGAN is to learn how to generate fake samples with a specific condition or characteristics rather than a generic sample purely based on random noise [15]. Specifically, for image translation tasks, both the generator and discriminator observe another input image $\mathbf{y}$. CGAN requires the generated image $G(\mathbf{x}, \mathbf{z})$ to not only fool the discriminator but also to be close to the ground truth output corresponding to the particular input image. Hence, in this work, we adopt this image translation idea proposed in [16].

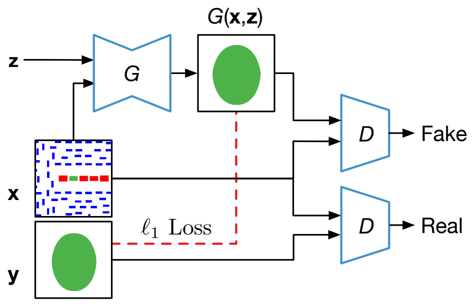

Figure 4: CGAN for lithography modeling.

Figure 4 shows the training process of our proposed lithography modeling CGAN. $\mathbf{x}$ represents the mask pattern image after SRAF insertion and OPC, and $\mathbf{y}$ represents the golden resist pattern of the target contact given by lithography simulation. The generator generates a fake resist pattern $G(\mathbf{x}, \mathbf{z})$ when fed with the input mask pattern $\mathbf{x}$. The discriminator is responsible for classifying this image pair $(\mathbf{x}, G(\mathbf{x}, \mathbf{z}))$ as fake, and meanwhile, it needs to predict the image pair $(\mathbf{x}, \mathbf{y})$ as real. Here "real" means that $\mathbf{y}$ is the output image corresponding to input $\mathbf{x}$. In other words, the target contact in $\mathbf{x}$ after resist development will become $\mathbf{y}$.

The discriminator outputs a value $D(\mathbf{x}, \mathbf{y})$ indicating the chance that $(\mathbf{x}, \mathbf{y})$ is a real pair. As demonstrated in Figure 4, the objective of the discriminator is to maximize the chance of recognizing the image pair $(\mathbf{x}, \mathbf{y})$ as real and the image pair $(\mathbf{x}, G(\mathbf{x}, \mathbf{z}))$ as fake. Mathematically, the objective function of D is given by [14]

$$\max_{D} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log D(\mathbf{x}, \mathbf{y})] + \mathbb{E}_{\mathbf{x},\mathbf{z}}[\log(1 - D(\mathbf{x}, G(\mathbf{x}, \mathbf{z})))]. \tag{1}$$

On the generator side, the objective is to generate images with the highest possible value of $D(\mathbf{x}, G(\mathbf{x}, \mathbf{z}))$ to fool the discriminator. Besides, the generator wishes that the generated image $G(\mathbf{x}, \mathbf{z})$ is close to the ground truth $\mathbf{y}$. The objective of G is defined as [15, 16]

$$\min_{G} \mathbb{E}_{\mathbf{x},\mathbf{z}}[\log(1 - D(\mathbf{x}, G(\mathbf{x}, \mathbf{z})))] + \lambda \cdot \mathbb{E}_{\mathbf{x},\mathbf{y},\mathbf{z}}[\|\mathbf{y} - G(\mathbf{x}, \mathbf{z})\|_1], \tag{2}$$

where ℓ_1 norm is used to quantify the pixel-wise difference between the generated image and the ground truth. In practice, it has been shown that ℓ_1 norm encourages less blurring when compared to ℓ_2 norm [16]. Combining Equation (1) and Equation (2), we have the following objective function for CGAN,

$$\min_G \max_D \ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log D(\mathbf{x},\mathbf{y})] + \mathbb{E}_{\mathbf{x},\mathbf{z}}[\log(1 - D(\mathbf{x}, G(\mathbf{x},\mathbf{z})))] \\ + \lambda \cdot \mathbb{E}_{\mathbf{x},\mathbf{y},\mathbf{z}}[\|\mathbf{y} - G(\mathbf{x},\mathbf{z})\|_1]. \tag{3}$$

The details of the CGAN architecture are summarized in Table 1. The problem that we consider maps a high-resolution input (256 × 256) to a high-resolution output (256 × 256), and a common approach to design such a generator is the use of an encoder-decoder network [14–16, 22]. The encoder passes the input through a series of layers that progressively downsample the input until a bottleneck layer; then the decoder reverses the process by progressively upsampling. In Table 1, column "Filter" gives the size and stride of the filter. All convolutional (Conv) and deconvolutional (Deconv) layers have 5 × 5 filters with a stride of 2. Batch normalization (BN) [23] is selectively applied on certain convolutional layers. The encoder uses leaky ReLU (LReLU) as the activation function, whereas the decoder uses ReLU. The discriminator is a convolutional neural network that performs classification to distinguish between the real image pairs and fake image pairs.

The standard approach to train GANs alternates between one step of optimizing D and one step of optimizing G [14]. In this way, we train both the generator and the discriminator to improve simultaneously, thus avoiding the case where one network is significantly more mature than the other. Here we use mini-batch stochastic gradient descent (SGD) for gradient update and apply the Adam solver [24] during the training stage.

3.3 LithoGAN

CGAN has demonstrated proven success in image generation tasks [15, 16] where generated images follow the distribution of the training data conditioned on the input images. However, for traditional computer vision tasks, locations of the objects in the generated image are not a major concern. For example, when trained to generate car images, the output of the GAN is judged upon based on the quality of an image as seen by a human while neglecting the exact location of the car in the image. However, for the lithography modeling task, the center of the generated resist pattern is as important as the shape of the pattern. Here the center refers to the center of the bounding box enclosing the resist pattern. In fact, we are interested in predicting a resist pattern which is accurate in both the shape and center.

With these two objectives in mind, and based on our experiments shown in Section 4, it is evident that CGAN falls short of predicting the correct center location of the resist pattern while demonstrating excellent results predicting the shape of the pattern. Hence, we propose a dual learning framework, referred to as *LithoGAN*, which splits the modeling task into two objectives:

- Resist shape modeling: a CGAN model is used to predict the shape of the resist pattern while neglecting the center;
- Resist center prediction: a CNN model is used to predict the center location of the resist pattern.

The application of the proposed LithoGAN framework is illustrated in Figure 5 where two data paths are shown. In the first path, a trained CGAN model is utilized to predict the shape of the resist pattern. During training, the golden pattern is re-centered at the center of the image, and the coordinates of the original center are saved for CNN training. In other words, the model is trained to

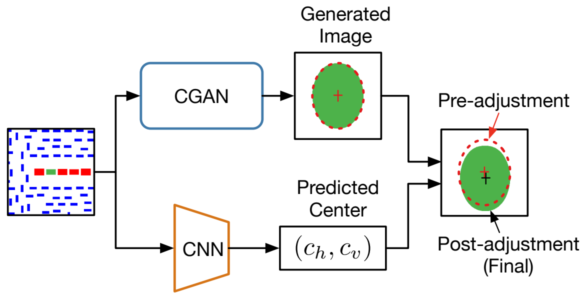

Figure 5: The proposed LithoGAN Framework.

predict resist patterns that are always centered at the center of the images. On the other hand, the second path is composed of a CNN trained to predict the center of the resist pattern based on the mask image. The CNN architecture for the resist center prediction task is shown in Table 2, where max-pooling (P) with filter size 2 × 2 and stride 2 is applied after each convolutional layer.

In such a way, the shape and the center of the resist pattern are predicted separately. They are combined in the last step before output. As shown in Figure 5, the image generated by CGAN is adjusted by recentering the resist shape based on center the coordinates predicted from the CNN. The resulting adjusted image is the final output of the LithoGAN framework.

4 EXPERIMENTAL RESULTS

The proposed framework for lithography modeling is implemented in Python with the TensorFlow library [25] and validated on a Linux server with 3.3GHz Intel i9 CPU and Nvidia TITAN Xp GPU. The experiments are performed on two benchmarks obtained from [12], where 982 and 979 mask clips are generated at 10nm technology node (N10) and 7nm node (N7) respectively. [12] performed SRAF insertion and OPC using Mentor Calibre [26], and then ran rigorous simulation to generate resist patterns using Synopsys Sentaurus Lithography [27] calibrated from manufactured data. In this work, the resist patterns generated by rigorous simulation are considered as the golden results. To guarantee highly accurate resist patterns, the pattern corresponding to the center contact in a clip is the only one adopted after each simulation. In other words, obtaining the golden resist pattern for each contact in a mask layout requires one rigorous simulation [28], and similarly, predicting this pattern using LithoGAN requires one model evaluation.

Each data sample for model training is a pair of the mask pattern image and the resist pattern image created using the color encoding scheme presented in Section 3.1. We randomly sample 75% of the data for training different models for N10 and N7 respectively, and the remaining 25% clips are for testing. In our experiments, we set the batch size to 4 and the number of maximum training epochs to 80. The weight parameter λ in Equation (3) is set to 100. The learning rate and the momentum parameters in the Adam optimizer are set to 0.0002 and (0.5, 0.999). The training time for each of CGAN and LithoGAN is around 2 hours. Note that we train the CGAN and LithoGAN models five times each with different random seeds to eliminate random performance variation. The results reported in this section are the average of the five runs.

Table 1: The CGAN architecture.

Generator Encoder			Generator Decoder			Discriminator		
Layer	Filter	Output Size	Layer	Filter	Output Size	Layer	Filter	Output Size
Input	-	256×256×3	Deconv-BN-LReLU	5×5,2	2×2×512	Input	-	256×256×6
Conv-ReLU	5×5,2	128×128×64	Dropout	-	2×2×512	Conv-LReLU	5×5,2	128×128×64
Conv-BN-ReLU	5×5,2	64×64×128	Deconv-BN-LReLU	5×5,2	4×4×512	Conv-BN-LReLU	5×5,2	64×64×128
Conv-BN-ReLU	5×5,2	32×32×256	Dropout	-	4×4×512	Conv-BN-LReLU	5×5,2	32×32×256
Conv-BN-ReLU	5×5,2	16×16×512	Deconv-BN-LReLU	5×5,2	8×8×512	Conv-BN-LReLU	5×5,1	16×16×512
Conv-BN-ReLU	5×5,2	8×8×512	Deconv-BN-LReLU	5×5,2	16×16×512	FC	-	1
Conv-BN-ReLU	5×5,2	4×4×512	Deconv-BN-LReLU	5×5,2	32×32×256			
Conv-BN-ReLU	5×5,2	2×2×512	Deconv-BN-LReLU	5×5,2	64×64×128			
Conv-BN-ReLU	5×5,2	1×1×512	Deconv-BN-LReLU	5×5,2	128×128×64			
			Deconv-LReLU	5×5,2	256×256×3			

Table 2: The CNN architecture.

Layer	Filter	Output Size
Input	-	256×256×3
Conv-ReLU-BN-P	7×7,1	128×128×32
Conv-ReLU-BN-P	3×3,1	64×64×64
Conv-ReLU-BN-P	3×3,1	32×32×64
Conv-ReLU-BN-P	3×3,1	16×16×64
Conv-ReLU-BN-P	3×3,1	8×8×64
FC	-	64
ReLU+Dropout	-	64
FC	-	2

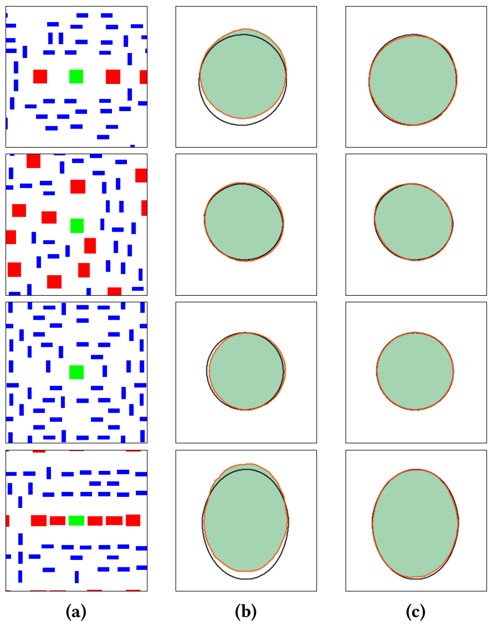

Figure 6: (a) Mask pattern input (b) CGAN output and (c) LithoGAN output. Each row represents one clip example. The golden contour is outlined in black. The prediction pattern is filled with green and outlined in red.

4.1 CGAN vs. LithoGAN

To demonstrate the performance of both frameworks discussed in this work: (i) the proposed lithography modeling CGAN and (ii) the improved LithoGAN, we visualize their performance in Figure 6. The top two rows are for samples from the N10 dataset, and the bottom two rows are for samples from the N7 dataset. According to [12], there are three types of contact arrays in the dataset, and Figure 6 includes at least one sample from each type. One can clearly see that CGAN outputs a shape very close to the golden resist pattern but the resist center can be quite far from the golden center; whereas, LithoGAN predicts both the shape and the center accurately. By examining the histogram showing the distribution of EDE in Figure 7, one can notice that LithoGAN can achieve lower EDE values when compared to CGAN; hence, making it closer to the golden solution.

LithoGAN achieves better accuracy compared to CGAN with the assistance of the CNN which predicts the location of the resist shape center. The average Euclidean distance between the golden

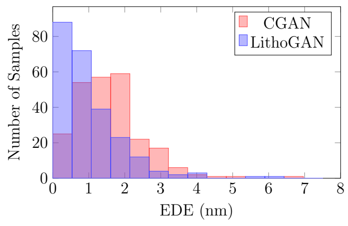

Figure 7: EDE distributions for CGAN and LithoGAN.

location of the center and the predicted location on the test set is used to measure the CNN prediction error. The error values for N10 and N7 datasets are 0.43 nm and 0.37 nm respectively.

Figure 8 gives a visualization example of how resist pattern images generated by LithoGAN progressively become more real and closer to the golden results along the training process. Besides, the loss changes of the generator and discriminator are depicted in Figure 9. It shows that the model converges after 50 epochs and produces resist patterns of high quality.

4.2 Framework Validation

We first compare the accuracy of our proposed LithoGAN with the state-of-the-art work on lithography modeling [12]. The work [12] first runs the optical simulation with Mentor Calibre [26] on the mask pattern clips. Then it uses the trained CNN model to predict four thresholds for each clip and performs threshold processing to generate the final contours. Instead, the proposed CGAN and LithoGAN for direct lithography modeling only need the mask pattern clips as input and directly output the resist shapes.

Table 3 gives a detailed comparison among the three methods using the proposed metrics in Section 2, where the average results over all the test samples are reported. In this work, the goal is to mimic the results of the rigorous simulation; hence, these results are considered a reference and all metrics are computed with reference to them. In addition to the mean EDE error over all the test samples, we also report the standard deviation for their EDE values. By examining the results in Table 3, one can easily find that LithoGAN outperforms CGAN in all the metrics, and the detailed comparison has been shown in Section 4.1. Besides, although [12] achieves slightly better results, LithoGAN is still competent for lithography usage at advanced technology nodes. That is because the average error of the critical dimension obtained from LithoGAN, 1.99 nm and 1.65 nm for N10 and N7 respectively, fall within the acceptable range (10% of the half pitch for contacts) [10, 12].

Next, we demonstrate the runtime comparison in Table 4. It is reported in [12] that the rigorous simulation for both of the two

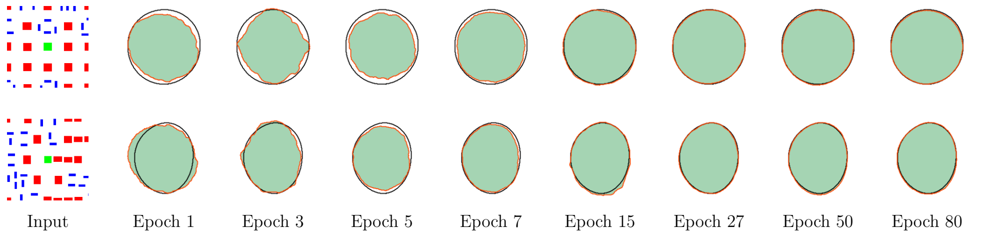

Figure 8: Visualization of the model advancement process. The prediction results for two testing samples using the LithoGAN model trained at different numbers of epochs are shown. Each row represents one clip example.

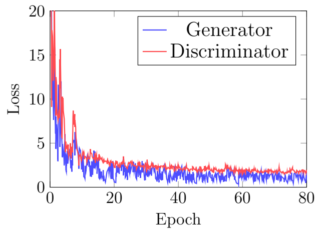

Figure 9: Loss curves of the generator and discriminator in LithoGAN.

Table 3: Comparison of evaluation metrics among different lithography modeling methods.

Dataset	Method	EDE (nm)		Pixel Acc.	Class Acc.	Mean IoU
		Mean	Std. dev.			
N10	Ref. [12]	0.67	0.55	0.98	0.99	0.98
	CGAN	1.52	0.95	0.96	0.97	0.94
	LithoGAN	1.08	0.88	0.97	0.98	0.96
N7	Ref. [12]	0.55	0.53	0.99	0.99	0.98
	CGAN	1.21	0.77	0.98	0.98	0.96
	LithoGAN	0.88	0.67	0.99	0.99	0.97

datasets takes more than 15 hours. For a fair comparison, we rerun the proposed lithography modeling flow in [12] on our platform. The first step in [12], optical simulation, takes around 80 minutes. We use the same training dataset as that of CGAN and LithoGAN to train their proposed CNN model. Prediction of the four thresholds for each sample in the entire dataset using the CNN model takes 8 seconds. Contour processing is performed on 6 cores in parallel and takes 15 minutes. On the other hand, prediction for an entire N10 or N7 dataset using our CGAN or LithoGAN model takes less than 30 seconds. By comparing the runtime of generating resist patterns for all clips reported in Table 4, one can notice that CGAN/LithoGAN can achieve ~1800× runtime reduction when compared to rigorous simulation and ~190× when compared to the flow with machine learning based threshold prediction approach [12]. Hence, the proposed LithoGAN framework achieves significant runtime reduction while obtaining evaluation results that fall within the accepted lithography range.

Table 4: Runtime comparison among different methods.

Method	Rigorous Sim	Ref. [12]			Ours (CGAN/LithoGAN)
		Optical Sim	ML	Contour	
Time	> 15h	80m	8s	15m	30s
Ratio	> 1800	190			1

Therefore, given its compelling speedup, LithoGAN paves the way for a new lithography modeling paradigm that can address the ever-increasing challenge of lithography simulation. This new paradigm can provide an accelerated framework which can perform within the adequate accuracy range for lithography.

5 CONCLUSION

In this work, we have presented the LithoGAN framework for end-to-end lithography modeling. LithoGAN is a dual learning network that predicts the resist shape using a CGAN model and predicts resist center using a CNN model. Experimental results show that the proposed framework predicts resist patterns of high quality while obtaining orders of magnitude speedup compared to conventional lithography simulation and previous machine learning based approach.

ACKNOWLEDGMENT

This work is supported in part by NSF under Award No. 1718570 and Toshiba Memory Corporation.

REFERENCES

[1] A. K.-K. Wong, *Resolution Enhancement Techniques in Optical Lithography.* SPIE press, 2001, vol. 47.
[2] C. A. Mack, *Fundamental Principles of Optical Lithography: The Science of Microfabrication.* John Wiley & Sons, 2008.
[3] H. Levinson, *Principles of Lithography.* SPIE press, 2011.
[4] C. A. Mack, *Field Guide to Optical Lithography.* SPIE Press Bellingham, 2006, vol. 6.
[5] X. Ma and G. R. Arce, *Computational lithography.* John Wiley & Sons, 2011, vol. 77.
[6] A. Taflove and S. C. Hagness, *Computational electrodynamics: the finite-difference time-domain method.* Artech house, 2005.
[7] K. D. Lucas, H. Tanabe, and A. J. Strojwas, "Efficient and rigorous three-dimensional model for optical lithography simulation," *Journal of the Optical Society of America A*, vol. 13, no. 11, pp. 2187–2199, 1996.
[8] "A review of model development for 10nm lithography," http://www.techdesignforums.com/practice/technique/model-development-10nm-lithography, 2015.
[9] T. M. A.-M. G. M. E. John Randall, Kurt G. Ronse, "Variable-threshold resist models for lithography simulation," in *Proc. SPIE*, vol. 3679, 1999.
[10] T. M. S. N. Yuki Watanabe, Taiki Kimura, "Accurate lithography simulation model based on convolutional neural networks," in *Proc. SPIE*, vol. 10147, 2017.
[11] Y. S. Seongbo Shim, Suhyeong Choi, "Machine learning-based 3d resist model," in *Proc. SPIE*, vol. 10147, 2017.
[12] 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 TCAD*, 2018.
[13] Y. Lin, M. B. Alawieh, W. Ye, and D. Z. Pan, "Machine learning for yield learning and optimization," in *Proc. ITC*, 2018, pp. 1–10.
[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.
[15] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. CVPR*, 2017, pp. 5967–5976.
[17] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. CVPR*, vol. 2, no. 3, 2017, p. 4.
[18] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.
[19] H. Yang, S. Li, Y. Ma, B. Yu, and E. F. Young, "GAN-OPC: Mask optimization with lithography-guided generative adversarial nets," in *Proc. DAC*, 2018, pp. 131:1–131:6.
[20] X. Xu, T. Matsunawa, S. Nojima, C. Kodama, T. Kotani, and D. Z. Pan, "A machine learning based framework for sub-resolution assist feature generation," in *Proc. ISPD*, 2016, pp. 161–168.
[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, 2015, pp. 3431–3440.
[22] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.
[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, 2014.
[25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *Proc. OSDI*, vol. 16, 2016, pp. 265–283.
[26] Mentor Graphics, "Calibre verification user's manual," 2008.
[27] Synopsys, "Sentaurus Lithography," https://www.synopsys.com/silicon/mask-synthesis/sentaurus-lithography.html, 2016.
[28] T. Kimura, T. Matsunawa, S. Nojima, and D. Z. Pan, "Hybrid hotspot detection using regression model and lithography simulation," in *Proc. SPIE*, vol. 9781, 2016.