Journal of
Micro/Nanolithography,
MEMS, and MOEMS

Nanolithography.SPIEDigitalLibrary.org

SoulNet: ultrafast optical source optimization utilizing generative neural networks for advanced lithography

Ying Chen
Yibo Lin
Lisong Dong
Tianyang Gai
Rui Chen
Yajuan Su
Yayi Wei
David Z. Pan

SPIE.

Ying Chen, Yibo Lin, Lisong Dong, Tianyang Gai, Rui Chen, Yajuan Su, Yayi Wei, David Z. Pan, "SoulNet: ultrafast optical source optimization utilizing generative neural networks for advanced lithography," *J. Micro/Nanolith. MEMS MOEMS* **18**(4), 043506 (2019), doi: 10.1117/1.JMM.18.4.043506.

SoulNet: ultrafast optical source optimization utilizing generative neural networks for advanced lithography

Ying Chen,[a,b,c] Yibo Lin,[c,d] Lisong Dong,[a,b] Tianyang Gai,[a,b] Rui Chen,[a] Yajuan Su,[a,*] Yayi Wei,[a,b,*] and David Z. Pan[c]
[a]Chinese Academy of Science, Institute of Microelectronics, Key Laboratory of Microelectronics Devices and Integrated Technology, Beijing, China
[b]University of Chinese Academy of Science, Beijing, China
[c]The University of Texas at Austin, Department of Electrical and Computer Engineering, Texas, United States
[d]Peking University, School of EECS, Center for Energy-Efficient Computing and Applications, Beijing, China

Abstract. An optimized source has the ability to improve the process window during lithography in semiconductor manufacturing. Source optimization is always a key technique to improve printing performance. Conventionally, source optimization relies on mathematical–physical model calibration, which is computationally expensive and extremely time-consuming. Machine learning could learn from existing data, construct a prediction model, and speed up the whole process. We propose the first source optimization process based on autoencoder neural networks. The goal of this autoencoder-based process is to increase the speed of the source optimization process with high-quality imaging results. We also make additional technical efforts to improve the performance of our work, including data augmentation and batch normalization. Experimental results demonstrate that our autoencoder-based source optimization achieves about $10^5\times$ speed up with 4.67% compromise on depth of focus (DOF), when compared to conventional model-based source optimization method. [DOI: 10.1117/1.JMM.18.4.043506]

Keywords: source optimization; machine learning; autoencoder neural networks.

Paper 19052 received Jun. 14, 2019; accepted for publication Oct. 25, 2019; published online Nov. 18, 2019.

1 Introduction

Optical lithography has been an important part in semiconductor manufacturing.[1] As the technology node continues to shrink, the feature sizes are getting smaller and smaller. The improvement of image quality relies on resolution enhancement techniques (RETs), such as phase-shift mask, optical proximity correction (OPC), and subresolution assist feature (SRAF) insertion. Source optimization, as one of the RETs, could help to improve the image quality by modifying the source intensity distribution and the incident light rays' directions.

Source optimization starts at the application of the off-axis illumination, then plenty of research on source optimization have been proposed.[2–7] Tian et al.[8] explored the relationship between the critical patterns and the optimized source and demonstrated the effectiveness of a pixelated source. Mulder et al.[9] described the FlexRay, which could generate free sources by manipulating an array of mirrors. With the benefit of faster convergence, particle swarm optimization is used by Wang et al.[10] to implement source optimization. In addition to obtaining the optimized source for some specific patterns, source optimization could also be conducted in the co-optimization of mask patterns and source.[11] There are various research efforts in this area. Peng et al.[12] utilized the gradients of the objective function to guide the optimization. Shen et al.[13] applied the level-set-based inverse lithography technology to co-optimize the source and the mask. Li et al.[14] used the augmented Lagrangian methods (ALMs) and quasi-Newton method to accelerate the optimization process. However, gradient-based method, level-set-based method, and ALMs are sensitive to local information and the optimization could be compromised. Therefore, Fuhner and Erdmann[15] applied a genetic algorithm for source mask optimization (SMO), due to its insensitivity to the local information. Sun et al.[16] formulated the SMO as a nonlinear compressive sensing reconstruction problem and applied the Newton-iteration hard-thresholding algorithm to accelerate convergence. Moreover, SMO for extreme ultraviolet lithography (EUVL) has also been explored. Ma et al.[17] developed the parametric SMO and the pixelated SMO with the gradient-based numerical algorithms for EUVL. There are also efforts toward fast methods for source optimization. For example, Rosenbluth and Seong[18] considered a simplified model in which the mask clips are already defined and where the exposing dose develops according to a simple threshold model. They formulated source optimization as a near-linear-programming problem and solved it quite quickly.

For source optimization, previous work mostly relied on mathematical–physical model construction and calibration, which demand huge computational resources and running time. Though this kind of model calibration and simulation aim to achieve optimal source optimization, it is not practical for real manufacturing due to the huge time cost. For each input layout, model simulation output the optimized source which could achieve the best imaging quality. In real integrated circuit manufacturing, in terms of the cost, different clips of layout patterns may share one source during lithography as long as the imaging quality meets the manufacture standard. The optimized sources obtained from simulation only provide guidance to the construction of the final source for manufacture. This means, source that could provide basic guidance is good enough for the manufacturing, and there is no need to spend too much time on getting the optimal source from the perspective of the calibrated model.

*Address all correspondence to Yayi Wei, E-mail: weiyayi@ime.ac.cn; Yajuan Su, E-mail: suyajuan@ime.ac.cn

1932-5150/2019/$28.00 © 2019 SPIE

The machine learning technique could learn from the training dataset and calibrate a mathematical model to conduct predictions. The model training and calibration are a one-time event, and predicting with the calibrated model is fast and comparably accurate. Therefore, the machine learning technique has been applied in computational lithography-related research, such as lithographic hotspot detection,[19] OPC,[20–22] and SRAF insertion.[23] The machine learning techniques take the feature of layout patterns as input, learn the correlation between layout features and the output object (e.g., hotspots or nonhotspots for the hotspot detection, the shifting distance of an edge segment for the OPC, and locations and shape for the SRAF insertion), and develop the prediction model. The machine learning techniques also fit for the optical source optimization domain. However, to the best of our knowledge, there is no prior research in applying the machine learning technique to the source optimization.

In this paper, we propose the first machine learning-based framework for source optimization. We apply an autoencoder neural network[24] in our approach. Usually training an autoencoder is an unsupervised learning, where the input is the same as the output. The autoencoder also could achieve dimensionality reduction and encode the input image data. There are many different researches on autoencoder. The autoencoder could be used for image reconstruction and anomaly detection. Zheng and Peng[25] applied an autoencoder for electrical capacitance tomography image reconstruction. Deng et al.[26] combined an unsupervised autoencoder training with a supervised classifier learning, trained a feature extractor along with a semisupervised classifier, and achieved satisfying results on speech emotion recognition. Zamini and Montazer[27] applied autoencoder-based clustering for fraud detection. Chen et al.[28] trained an autoencoder and used the reconstruction error as the anomalous scores to find the anomaly.

Our approach, which is based on autoencoder, could achieve fast source optimization with little imaging quality compromise. The main contributions are summarized as follows.

- An autoencoder-based framework is proposed for the source optimization, where the model is calibrated with the model-based optimized source as the training data.
- A pretraining and fine-tuning paradigm is developed to fit the real manufacturing, which trains the prediction model with the consideration of limited data.
- The experimental results show that the framework can achieve at least $10^5\times$ speed-up with about 4.67% compromise on depth of focus (DOF), compared to the professional software for source optimization.

The rest of the paper is organized as follows. Section 2 introduces basic concepts and provides the problem formulation. Section 3 presents the detailed algorithm for the autoencoder-based neural networks. Section 4 validates the proposed framework with experimental results. Section 5 concludes the paper.

2 Preliminaries and Problem Formulation

In this section, we will review the background of source optimization and provide the problem formulation in this work.

2.1 *Source Optimization*

In optical lithography, light emitting from the illumination source will pass through the lithography mask and transform the shape of the layout patterns from the mask to the photoresist. The intensity and incident angle of light could influence the quality of the printing contour. A bad imaging quality will cause a fail transform and will lead to a rework of lithography. To avoid this, source optimization is indispensable.

Source optimization starts from a simple set of parameters for annular and multiple source shapes. Then the freeform illumination sources[9] lead to superior tuning flexibility by describing the source with intensity points arrayed in two-dimensional (2-D) grids. Conventionally, the lithography imaging model is used to help in optimizing the source. Mainly, the imaging model includes the aerial imaging formation and the photoresist development. Pixel values of the source are updated iteratively with the objective to minimize the error difference between the photoresist image and the target pattern. Optimized source could be constructed easily through this process. Lithography imaging model-based optimization is extremely time-consuming and often slows down the fabrication.

Figure 1 presents the conventional flow which is used to generate the final source in real manufacturing. First, we choose the representative layout clips from the whole layout clips. These representative layout clips cover the critical and tight parts of the whole layout clips. This step typically is conducted by lithography engineers based on their experience. Then, we conduct source optimization on the selected representative layout clips to get the optimized global source. This optimized global source is suitable for printing all selected clips with good performance. Then, to check the global source's suitability for all layout clips, we apply the global source on all layout clips and check the imaging performance through lithography simulation. If the imaging performance is acceptable, then the optimized global source is set as the final source. Otherwise, we need to find the critical process window (PW)-limited layout clip and get the optimized local source for the clip through source optimization.

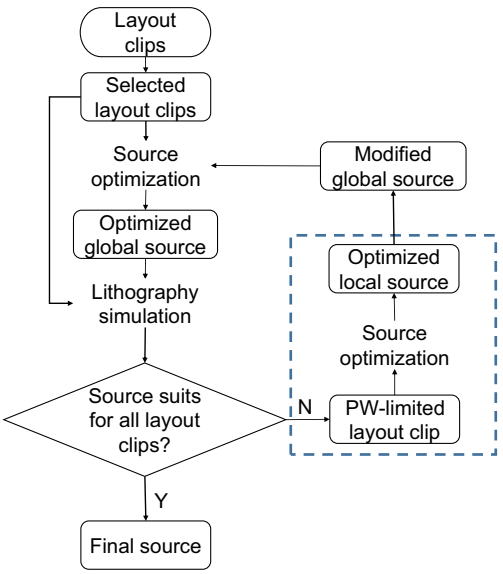

Fig. 1 The conventional final source generation flowchart in real manufacturing.

This process is marked with the blue dash rectangle in Fig. 1. Then, the lithography engineer could use the optimized local source as a guide to modify the optimized global source based on their experience. We use the modified global source as the starting point for source optimization on the selected layout clips and get the new optimized global source. The new optimized global source is used to check the imaging performance again. This iteration will stop until the optimized global source is suitable for all layout clips to get the acceptable imaging quality.

To evaluate the performance of an optimized source, the imaging contour is simulated through the lithography model with the optimized source as the illuminator. The better the imaging quality, the better the performance of the optimized source. To quantify the lithographic variations, we define the following metrics.

Definition 1 (PW). To guarantee the contour on photoresist meets the process standard, lithography parameters are all under control of a small region, which is called the PW.[29] PW is defined by focus–energy matrix and is evaluated by DOF.[29]

Definition 2 (DOF). DOF is used to evaluate PW. It represents the relationship between the imaging quality and the position of wafer surface. The printing quality is guaranteed to be good within DOF.

Definition 3 (exposure latitude). Exposure latitude (EL)[29] is a measure of a lithographic system's insensitivity to the dose variation. It relates to the aerial image contrast. High EL permits a greater variance of exposure and still achieves an acceptable result.

Definition 4 (mask error enhancement factor). Mask error enhancement factor (MEEF)[29,30] is defined as the ratio of the change in resist feature width to the change in mask feature width, supposing that everything else in the process remains constant.

The objective of source optimization is to minimize the difference between the photoresist contour and the target layout patterns. To do this, the aim is to maximize DOF and EL, while minimizing MEEF. DOF is a more important concern than EL or MEEF in the production applications of interest to us. For this reason, our procedure gives paramount importance to maximizing DOF.

2.2 *Problem Formulation*

As shown in Fig. 1, when facing a PW-limited layout clip, one source optimization needs to be conducted. In practice, obtaining the optimized local source for the layout clip through source optimization is very time-consuming, Moreover, the industry only needs the optimized local source to be a guide for the lithography engineer to determine the final source in real manufacturing. It is impractical to cost too much time to get the optimized source through model-based simulation. Therefore, it is desired to build an optimized source predictor with the ability to predict sources that can lead to a good imaging quality in a short time.

As shown in Fig. 2, we could use the machine learning-based source optimization to replace the local source optimization process as marked in Fig. 1. This method takes the PW-limited layout clip as input and gives the optimized local source as output. The goal of this machine learning-based method is to predict the optimized local source faster with satisfying imaging quality.

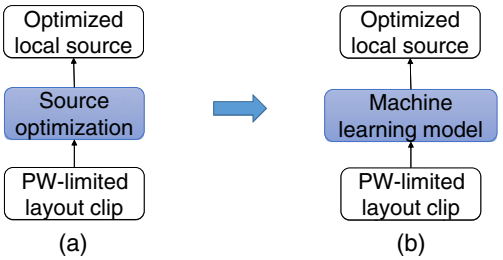

Fig. 2 The comparison of (a) the conventional local source optimization and (b) the autoencoder based source optimization.

For source optimization based on machine learning approach, the training dataset is formed by a set of target layout pattern clips and the model-based sources. The machine learning technique could take an input layout clip as an image. The optimized source leading to the best imaging quality could be seen as its label. Therefore, source optimization based on machine learning can be formulated as an image reconstruction problem in which the lithography imaging process information is stored in the correlation between input samples and their labels. Related terminologies are shown as follows.

Definition 5 (source label). Source is represented by the pixel source values in 2-D grid plane. For simplicity, the source label for each grid is a number between 0 and 1, which represents the light intensity.

We then formulate the machine learning-based source optimization problem as follows.

Problem 1 (machine learning-based source optimization). Given the training layout clips with model-based sources, feature vectors and source labels are extracted and a regression model is trained to predict the pixelated light intensity for good imaging quality.

3 Algorithms

In this section, we will explain how our autoencoder-based learning model works. First, the basic concepts of autoencoder are introduced, and then the autoencoder architecture in our approach is shown. Finally, the whole framework for source optimization is presented.

3.1 *Conventional Autoencoder*

Normally, a basic autoencoder consists of an encoder and a decoder.[31] The encoder compresses the input vector $\mathbf{x}$ into hidden representation $\mathbf{h}$, and then $\mathbf{h}$ is mapped back to reconstruct $\hat{\mathbf{x}}$ through the decoder. The mapping functions of encoder and decoder are as follows:

$$h = s_f(Wx + b), \tag{1}$$

$$\hat{x} = s_g(W'h + b'), \tag{2}$$

where $\mathbf{x}$ is the input vector, $\mathbf{W}$ and $\mathbf{W}'$ represent the encoder and the decoder weight matrices, respectively, and $\mathbf{b}$ and $\mathbf{b}'$ are the bias vectors. Here $\hat{\mathbf{x}}$ is the output vector. The

autoencoder intends to minimize the difference between $\mathbf{x}$ and $\hat{\mathbf{x}}$ by finding the optimal parameter values for $\boldsymbol{\theta} = (\mathbf{W}, \mathbf{b}, \mathbf{W}', \mathbf{b}')$. The reconstruction error is evaluated by squared error function as in Eq. (3).

$$L(x;\theta) = \sum_{i=1}^{n} \|x_i - \hat{x}_i\| \tag{3}$$

3.2 *Convolutional Autoencoder*

Convolutional autoencoder (CAE) is a special kind of autoencoder.[32] The CAE is similar to the conventional autoencoder, except that there is no fully connected neural layers in CAE. Also, the CAE model applies convolutional layer in the encoder and deconvolutional layer in the decoder. Owing to the utilization of convolutional and deconvolutional layers, the CAE only accept 2-D data as the input vector. The latent representation of k'th feature map is given as

$$h^k = \sigma(x \otimes W^k + b^k), \tag{4}$$

where $\mathbf{x}$ is the input 2-D vector and $\mathbf{W^k}$ and $\mathbf{b^k}$ represent the k'th filter weight matrix and the bias matrix, respectively. Here σ is the activation function and $\otimes$ denotes the the convolution operation. For example, if the size of $\mathbf{x}$ is $I \times I \times d$, the size of $\mathbf{W^k}$ is $F \times F \times d \times o$, and the size of the $\mathbf{b^k}$ is o, then we could utilize stride s with zero-padding for the filter and get the output $\mathbf{h^k}$ with the size of $I/s \times I/s \times o$. For the deconvolutional part, the reconstruction $\hat{x}$ is obtained by the Eq. (5).

$$\hat{x}^k = \sigma(h'^k \hat{\otimes} W'^k + b'^k), \tag{5}$$

where $\mathbf{h'^k}$ is the input vector and $\mathbf{W'^k}$ and $\mathbf{b'^k}$ represent the k'th deconvolutional filter weight matrix and the bias matrix, respectively. The only difference between Eqs. (4) and (5) is $\hat{\otimes}$, which denotes the deconvolutional operation. We present an example to explain the details of the deconvolutional operation as follows:[33] the size of $\mathbf{h'^k}$ is $h \times h \times o$, the size of $\mathbf{W'^k}$ is $F \times F \times d \times I$, the size of the b^k is d, the stride is s with zero-padding, and the size of $\hat{\mathbf{x}}^\mathbf{k}$ is $I \times I \times d$. The goal is to increase the feature number from h to I. To do that, we first insert $(s-1)$ zeros between two numbers in h'^k and expend its dimension as the output size (I) with zero-padding, then we apply $\mathbf{W'^k}$ on the new $\mathbf{h'^k}$ with stride size as 1 and zero-padding to conduct convolutional operations. After that, the size of $h'^k \hat{\otimes} W'^k$ is $I \times I \times d$, and the feature number is increased from h to I.

Figure 3 shows an illustration of the convolutional and deconvolutional layers. In CAE, the encoder applies the convolutional layers to decrease the feature number, while the decoder applies the deconvolutional layers to increase the feature number. Similar to the conventional autoencoder, the CAE training intends to minimize the reconstruction error.

U-Net[34,35] is a special CAE architecture. For better precise locations, U-Net includes skip connections to concatenate all channels at the encoder layers with the mirrored decoder layers, as shown in Fig. 4. U-Net is mostly suitable for situations when the input and the output share the location of prominent edges. For source optimization, though there is no similar parts between the input layout clips and the output

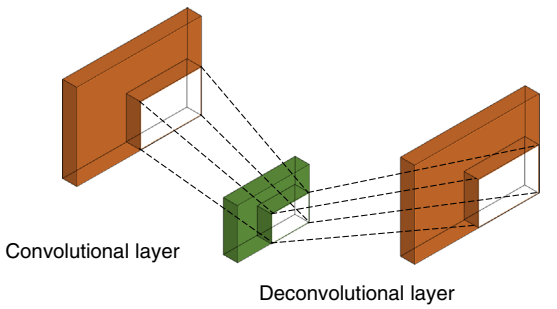

Fig. 3 The illustration of convolutional and deconvolutional layers.

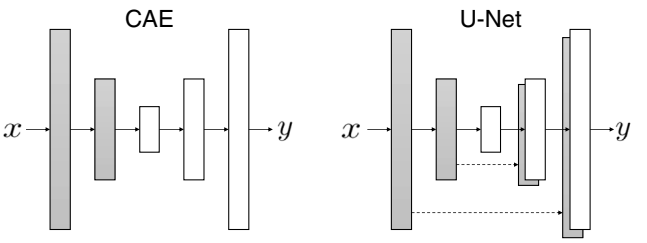

Fig. 4 Comparison of CAE and U-Net.

source images, inner correlations are contained in the lithography printing process. Our task for source optimization has unique characteristics such as limited layout–source training data and complex layout reconstruction. U-Net is able to deal with limited training data with good generality and also propagates more data information with additional feature channels (such as the skip connections) for reconstruction. Therefore, in this paper, we utilize U-Net to do the model training.

The architecture of the U-Net is shown in Fig. 5. The layers used in the encoder and the decoder are mirrored, except that we use skip connections by concatenating the output of each deconvolutional layers with the feature maps from the encoder at the same level. The mathematical formulation of skip connections is given in Eq. (6).

$$z_k = \begin{cases} x_k, & \text{if } k < q_1 \\ y_{k-q_1}, & \text{others} \end{cases}, \tag{6}$$

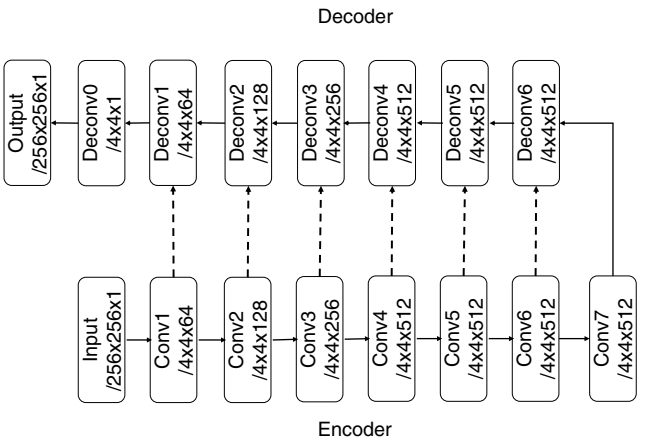

Fig. 5 The architecture of the U-Net.

where $\mathbf{x} \in R^{m\times n\times q_1}$ represents the output from the convolutional layer, $\mathbf{y} \in R^{m\times n\times q_2}$ represents the output from the deconvolutional layer, and $\mathbf{z} \in R^{m\times n\times(q_1+q_2)}$ represents the output after skip connections between x and y. Here k denotes the index of their third dimension. For example, in Fig. 5, the size of both Conv1 output (x) and Deconv1 output (y) is $128 \times 128 \times 64$ ($m = n = 128$, $q_1 = q_2 = 64$). The skip connection concatenates the channel path of x and y, and then the size of the input vector (z) for Deconv0 becomes $128 \times 128 \times 128$.

Maxpooling/unmaxpooling may lead to information loss; therefore, there is no maxpooling or unmaxpooling layers. We apply only convolutions for downsampling. Owing to its benefit to fast convergence and nonlinearity to the network, a rectified linear unit (ReLU) layer is applied for activation following the convolution layer. Since we use 16 hidden layers, the training network becomes a complicated deep neural network and face a problem called internal covariate shift. Internal covariate shift occurs since the distribution of each layer's input changes with the parameters of the previous layers during model training. This leads to a slow and hard training process due to the lower learning rates requirement and the saturating nonlinearities. To deal with this problem, we perform the batch normalization for each layer input. The kernel sizes and the number of kernels are annotated in the Fig. 5. Each convolutional part consists of one 4×4 convolution with stride 2 for downsampling and a following ReLU layer. Detailed configurations are shown in Table 1.

Moreover, different from the basic autoencoder, which is usually trained for reconstruction with the objective of making the input and the output as similar as possible, our work intends to get the optimized source for an input layout pattern. With the source label, the autoencoder model training becomes a supervised learning instead of unsupervised learning. Therefore, the new cost function for U-Net-based source optimization is defined as in Eq. (7).

$$L_{\mathrm{SO}} = \sum_{i=1}^{n} \|y_i - \hat{y}_i\|, \tag{7}$$

where n is the number of input samples and vectors $\mathbf{y_i}$ and $\hat{\mathbf{y}}_\mathbf{i}$ are the pixelated model-based source and autoencoder-generated source for i'_{th} sample, respectively.

3.3 *Framework of the Autoencoder-Based Network for Source Optimization*

Typically, machine learning-related techniques, especially supervised learning, rely on abundant training data for good prediction performance. In real manufactoring, collecting a large amount of layout–source pairs is not practical. Therefore, we propose an autoencoder-based framework for source optimization with limited number of layout–source pairs. The flow of our framework is shown in Fig. 6. It consists of the following parts.

3.3.1 *Data preparation*

For autoencoder model training, we employ the original target layout clips as the input and the model-based sources as the output. An example of the layout clip and the model-based source is shown in Fig. 7. The light intensity of the source increases as the color changes from blue to red. During the training, the layout clips and the model-based sources are taken as images. To fit the training requirement, we need to preprocess the collected layouts and the sources. The original layout clip size is 1200×1200 nm^2, which is expensive for neural networks to process. We first generate images with 1200×1200 pixels and downscale to 256×256 with the nearest-neighbor algorithm. For the preprocess of the sources, the original model-based source is represented with 201×201 pixels. The pixel values range from 0 to 1, which represent the light intensity. We convert the source pixel value array to the grayscale map with the size of 201×201. The grayscale map is scaled up to 256×256 and transformed to the pixel value array. Then, we get the source images with 256×256 pixels. We take the grayscale map of the source for training. After training, the model outputs grayscale map, and we convert it to light intensity pixel by pixel with the constraint ranging from 0 to 1. In addition, we apply data augmentation to increase the training sample size due to the big network structure and the limited training

Table 1 Neural network configuration.

Encoder				Decoder			
Layer	Kernel size	Stride	Output node #	Layer	Kernel size	Stride	Output node #
Conv1	4	2	$128 \times 128 \times 64$	Deconv6	4	2	$4 \times 4 \times 512$
Conv2	4	2	$64 \times 64 \times 128$	Deconv5	4	2	$8 \times 8 \times 512$
Conv3	4	2	$32 \times 32 \times 256$	Deconv4	4	2	$16 \times 16 \times 512$
Conv4	4	2	$16 \times 16 \times 512$	Deconv3	4	2	$32 \times 32 \times 256$
Conv5	4	2	$8 \times 8 \times 512$	Deconv2	4	2	$64 \times 64 \times 128$
Conv6	4	2	$4 \times 4 \times 512$	Deconv1	4	2	$128 \times 128 \times 64$
Conv7	4	2	$2 \times 2 \times 512$	Deconv0	4	2	$256 \times 256 \times 1$

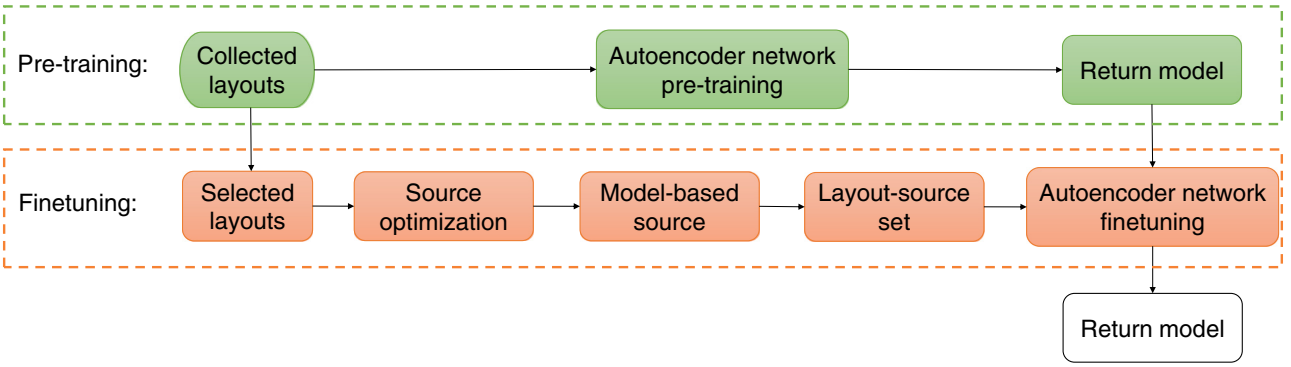

Fig. 6 Overall flowchart of autoencoder-based source optimization.

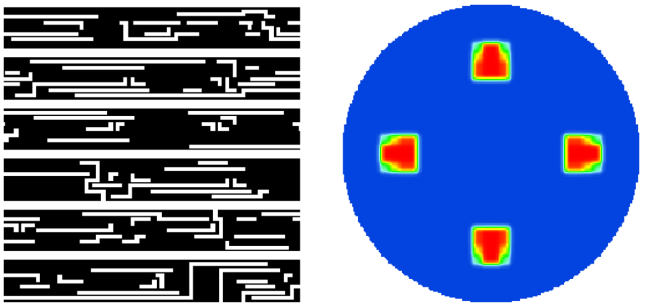

Fig. 7 The illustrations of (a) a layout and (b) a model-based source.

data. For a layout, the optimized source is fourfold symmetrical. Therefore, we use one quadrant of the source for model training; we also generate similar training layout data with the same source labels. For each training layout clip, we apply the image rotate and flip transform. The rotated/flipped layout image shares identical source label with the original layout image. In addition to the original layout image, we create five more images: 90-deg rotate, 180-deg rotate, 270-deg rotate, X-flip, an Y-flip. After the training, we get one quadrant of the source for one input layout clip, and then we unfold it to get the final optimized source.

3.3.2 *Autoencoder pretraining*

In this step, we use the collected layout clips to train the autoencoder. Since we only get limited number of layout–source pairs, this step intends to pretrain the autoencoder network. Both the input and the output are the layout clips. The autoencoder is trained for layout clips reconstruction. The autoencoder is optimized by minimizing the cost function in Eq. (3). After the pretraining, the autoencoder model is saved for fine-tuning.

3.3.3 *Autoencoder fine-tuning*

In the fine-tuning phase, we select representative layout clips from the collected layouts, with the number of the selected layout clips being S. Usually S is less than 30, considering the time cost in real manufacture. We conduct conventional source optimization to get the optimized sources for those selected layout clips. And then those layout–source pairs are fed into the pretrained autoencoder model for another round of training. we use the layout clip as input and the source as output for training. During the training, the parameters of the encoder are fixed and are obtained from the pretraining phase. We only retrain the decoder in fine-tuning. Through this step, the autoencoder model is fine-tuned for source optimization by minimizing the cost function in Eq. (7).

4 Experimental Results

4.1 *Experimental Setup*

This autoencoder neural network is implemented in Python with Tensorflow 1.12.0[36] on a Linux server with three 8-core 2.5GHz CPUs, a Nvidia Tesla P100 GPU, and 32 GB memory. The framework is validated on 14-nm technology node. The collected layouts include two parts: one is the regular and simple test pattern (such as head-to-head, head-to-tip) and the other is scaling down from the 28-nm industrial benchmarks from ICCAD2012 CAD contest. To be noted, this contest is for hotspot detection. We only select the nonhotspot patterns for our experiments. The clips are randomly split into training and testing samples, so we can consider that both datasets follow the same distribution. As a general methodology, the proposed approach will work as long as the training and testing datasets come from the same source designs and follow the same distribution. The number of training samples and testing samples are 441 and 99, respectively. Tachyon of ASML Brion[37] on server with four Intel Xeon2 GHz octuple-core CPUs, 256 GB memory and 5 tachyon license is used to conduct lithography simulation. Considering imaging quality and real manufacturing requirement, we apply SMO instead of single source optimization to get the model-based source. The cost function for SMO is as follows:

$$f = \sum \|\mathrm{EPE}\| + P, \tag{8}$$

where p is the penalty term. Edge placement error (EPE) is the difference between lithography contour and design target. Here $\sum \|\mathrm{EPE}\|$ sums EPEs under different conditions, varying focus, dose, and mask bias. Minimizing the EPE with different focus, dose, and mask bias will help improve DOF, EL, and MEEF, respectively. In our experiment, for SMO, the ranges of focus, dose, and mask bias are ± 40 nm, $\pm 3\%$and ± 0.5 nm, respectively.

Table 2 shows the details of the training configurations for autoencoder. We apply Adam[38] as the gradient descent

Table 2 Training configurations of autoencoder.

Configurations	Value
Optimizer	Adam[38]
Initial learning rate	0.0002
$\beta 1$ (Adam)	0.5
Batch size	4
Dropout rate	0.5
Total training steps for pretraining	35,000
Total training steps for fine-tuning	2000
Number of selected layout clips (S)	20

optimizer for model training; the initial learning rate is 0.0002 and the exponential decay rate (also known as $\beta 1$) is 0.5. The batch size is 4, and the total training steps for pretraining and fine-tuning are 35,000 and 2000, respectively. We also apply a dropout ratio of 0.5 after the deconvolutional part to avoid overfitting. The number of selected layout clips for fine-tuning is 20. These selected layout clips are the representative layout clips, which are selected by lithography engineers with their experience.

4.2 *Performance Evaluation*

Optimizing the source with tachyon usually takes 1 to 4 h. The time it takes depends on the complexity of the layout patterns. New simulation needs to be conducted when facing a new layout clip. Training the autoencoder model needs about 2 h.

Normally, for a basic autoencoder work, the reconstruction error in Eq. (3) could be used as a score to evaluate the trained model performance. However, difference between the model-based source and autoencoder-based source is not suitable for evaluating the generated source, due to the characteristics of lithography process. To validate our framework, we compare the imaging performance between the autoencoder-based source and the model-based source, with the latter as the baseline (ratio = 1). The baseline quality metrics are calculated using SMO-optimized masks (co-optimized masks). As for SoulNet, we apply the autoencoder-based source, optimize the mask (doing OPC and inserting SRAF) under the source, and then get the imaging performance through simulation. For each testing samples, we consider its overlapped DOF[29] and the worst-case MEEF. The result analyses presented in this section are based on the 99 testing samples.

Table 3 Performance comparison on DOF, EL, MEEF and optimization time.

		DOF	EL	MEEF	Optimization time
Model-based	Average	**141.17**	**0.1158**	**3.25**	3.5 h
	Ratio	**1**	**1**	**1**	1
Autoencoder-based	Average	134.59	0.1110	4.34	**0.1237 s**
	Ratio	0.9533	0.9590	1.3388	**9.82E-06**

Figure 8 shows the illustrations of the model-based source and the autoencoder-based source for a layout clip. We utilize the metrics in Sec. 2 to evaluate the imaging performance. For a clear comparison, we lists the average DOF, EL, and MEEF on the testing set in Table 3. The calculation of the average metrics among all testing samples is performed to evaluate the general performace of SoulNet. Moreover, for a fair comparison, we run all the optimization processes with CPU to compare the optimization time. Since the training of the autoencoder is a one time thing, we only consider the running time of prediction for the autoencoder-related process. The average source optimization time on the testing set of model-based source optimization and autoencoder-based source optimization is also shown in Table 3. The metrics with better performance are shown in boldface. In our experiment, we focus on the imaging results on DOF. Compared to model-based source optimization approach, autoencoder-based source optimization achieves about $10^5\times$ speed-up with about 4.67% compromise on DOF. It enables huge time-saving compared to the conventional source optimization.

4.3 *Evaluation on the Effectiveness of U-Net*

For autoencoder network training, we adopt the U-Net. To evaluate the effectiveness of the U-Net, we conduct a side

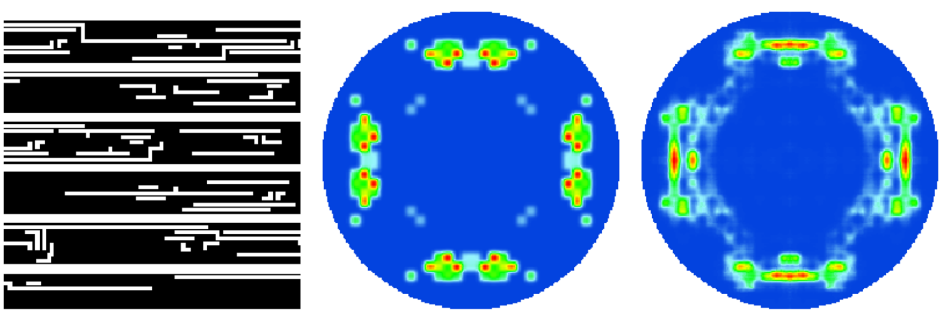

Fig. 8 Illustrations of (a) a layout clip, (b) a model-based source, and (c) an autoencoder-based source.

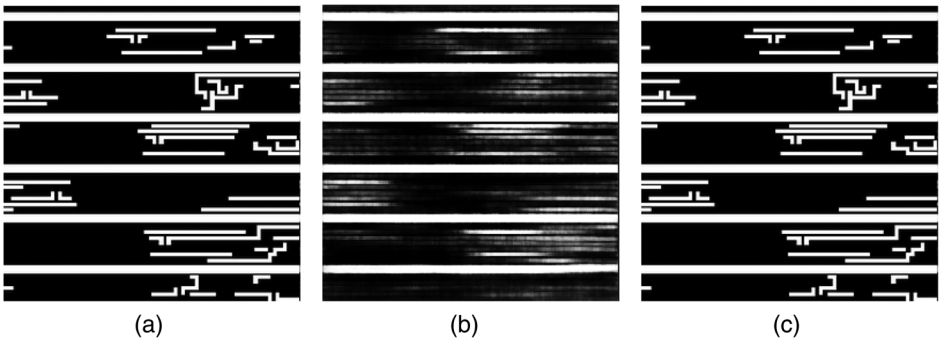

Fig. 9 Illustrations of (a) a layout clip, (b) the reconstruction image of CAE model, and (c) the reconstruction image of U-Net model.

experiment by canceling all skip connections of the U-Net and training a CAE model. During the pretraining, the CAE model only could output the critical main features of the layout but without the details. The illustrations of the input layout clip and its CAE output are shown in Fig. 9. It happens because the 256×256 pixelated layout image is too complex for the CAE architecture to construct a good model. Therefore, the U-Net performs better than the CAE.

4.4 *Imaging Performance Analysis on Different Layouts*

Table 3 gives the average performance evaluation on all testing samples. To get more specific analysis on each testing clips, Fig. 10 is plotted to show the DOF, EL, and MEEF variations of every testing clip with autoencoder-based source, compared to model-based source. The model-based source is set as the baseline (ratio = 1). The testing layout clips are sorted with the ascending order of the DOF's ratio. In Fig. 10, EL and DOF fluctuate evenly, while MEEF of most testing layout clips tend to increase. During imaging, the light intensity and distribution at the diagonal position are optimized to improve MEEF. Usually there are clear poles at the diagonal positions, as shown in Fig. 8(b). However, for most sources generated by the autoencoder, there are blurry parts at the diagonal positions. It leads to the increase of MEEF. Moreover, for some test layout clips,

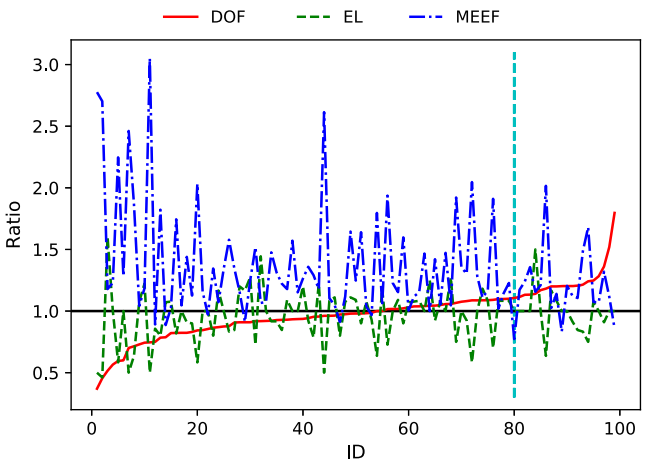

Fig. 10 Performance variation on DOF, EL, and MEEF for each testing samples.

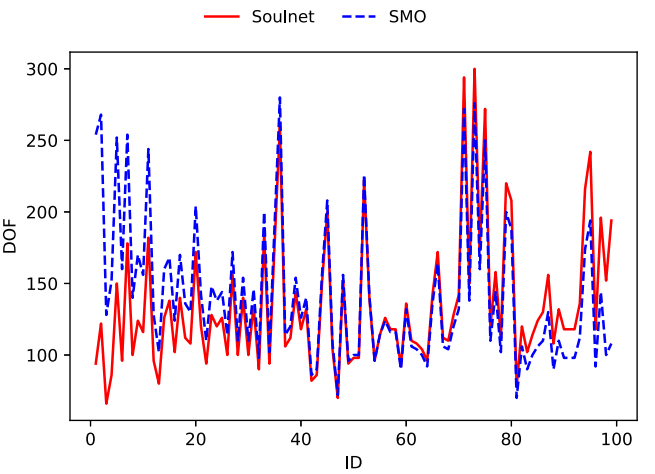

Fig. 11 The DOF comparison between SoulNet source and SMO source.

the autoencoder-based source even outperforms the model-based source. One example is marked with the blue vertical dash line in Fig. 10. For layout ID 80, the DOF increases 15%, the MEEF decreases 23%, the EL stays the same. With the autoencoder source, its imaging performance becomes better, which proves the potential of the autoencoder-based source optimization framework.

More specifically, Fig. 11 exhibits specific DOF values for every testing samples with SoulNet source (also known as autoencoder-based source) and SMO source (also known as model-based source). In Fig. 11, the last 20 samples show tigher DOF than the first 20 samples with SMO sources, and their DOFs are imporved by SoulNet. This indicates the potential of SoulNet on achieving advantages in DOF for samples with tight DOF under model-based source.

5 Conclusion

In this paper, we present the first autoencoder-based source optimization framework. In addition to utilizing the nature of autoencoder networks, we also make additional technical efforts to improve the performance of our work, including data augmentation and batch normalization. Compared to conventional model-based source optimization approaches, our method achieves a fast optimization time with acceptable compromise on imaging quality. It fits the industry requirements, which demands to get the optimized source to guide the source set in real manufacturing within short time.

Experimental results demonstrate the effectiveness and efficiency of our framework. Moreover, SoulNet compromises MEEF, as mentioned in Sec. 4.4, and we will try to improve MEEF in future works.

Acknowledgments

This work is supported by the National Science and Technology Major Project of China (Grant Nos. 2017ZX02315001 and 2017ZX02101004), the Opening Project of Key Laboratory of Microelectronic Devices and Integrated Technology, Institute of Microelectronics, Chinese Academy of Sciences (No. Y9YS02X001), and China Scholarship Council (No. 201704910587).

References

1. X. Ma and G. R. Arce, *Computational Lithography*, Vol. **77**, John Wiley & Sons, Inc., Hoboken, New Jersey (2011).
2. M. Burkhardt et al., "Illuminator design for the printing of regular contact patterns," *Microelectron. Eng.* **41**, 91–95 (1998).
3. J.-C. Yu, P. Yu, and H.-Y. Chao, "Fast source optimization involving quadratic line-contour objectives for the resist image," *Opt. Express* **20**(7), 8161–8174 (2012).
4. R. Socha et al., "Illumination optimization of periodic patterns for maximum process window," *Microelectron. Eng.* **61**, 57–64 (2002).
5. J. Li, Y. Shen, and E. Y. Lam, "Hotspot-aware fast source and mask optimization," *Opt. Express* **20**(19), 21792–21804 (2012).
6. X. Ma and G. R. Arce, "Pixel-based simultaneous source and mask optimization for resolution enhancement in optical lithography," *Opt. Express* **17**(7), 5783–5793 (2009).
7. T.-S. Gau et al., "Customized illumination aperture filter for low k_1 photolithography process," *Proc. SPIE* **4000**, 271–283 (2000).
8. K. Tian et al., "Benefits and trade-offs of global source optimization in optical lithography," *Proc. SPIE* **7274**, 72740C (2009).
9. M. Mulder et al., "Performance of FlexRay: a fully programmable illumination system for generation of freeform sources on high NA immersion systems," *Proc. SPIE* **7640**, 76401P (2010).
10. L. Wang et al., "Pixelated source optimization for optical lithography via particle swarm optimization," *J. Micro/Nanolithogr. MEMS MOEMS* **15**(1), 013506 (2016).
11. A. E. Rosenbluth et al., "Optimum mask and source patterns to print a given shape," *J. Micro/Nanolithogr. MEMS MOEMS* **1**(1), 13–31 (2002).
12. Y. Peng et al., "Gradient-based source and mask optimization in optical lithography," *IEEE Trans. Image Process.* **20**(10), 2856–2864 (2011).
13. Y. Shen, N. Wong, and E. Y. Lam, "Level-set-based inverse lithography for photomask synthesis," *Opt. Express* **17**(26), 23690–23701 (2009).
14. J. Li, S. Liu, and E. Y. Lam, "Efficient source and mask optimization with augmented Lagrangian methods in optical lithography," *Opt. Express* **21**(7), 8076–8090 (2013).
15. T. Fuhner and A. Erdmann, "Improved mask and source representations for automatic optimization of lithographic process conditions using a genetic algorithm," *Proc. SPIE* **5754**, 415–427 (2005).
16. Y. Sun et al., "Fast nonlinear compressive sensing lithographic source and mask optimization method using Newton-IHTS algorithm," *Opt. Express* **27**(3), 2754–2770 (2019).
17. X. Ma et al., "Gradient-based source mask optimization for extreme ultraviolet lithography," *IEEE Trans. Comput. Imaging* **5**(1), 120–135 (2019).
18. A. E. Rosenbluth and N. Seong, "Global optimization of the illumination distribution to maximize integrated process window," *Proc. SPIE* **6154**, 61540H (2006).
19. Y. Chen et al., "Semi-supervised hotspot detection with self-paced multi-task learning," in *Proc. 24th Asia and South Pac. Des. Autom. Conf.*, ACM, pp. 420–425 (2019).
20. X. Ma et al., "Fast pixel-based optical proximity correction based on nonparametric kernel regression," *J. Micro/Nanolithogr. MEMS MOEMS* **13**(4), 043007 (2014).
21. X. Ma et al., "A fast and manufacture-friendly optical proximity correction based on machine learning," *Microelectron. Eng.* **168**, 15–26 (2017).
22. H. Yang et al., "GAN-OPC: mask optimization with lithography-guided generative adversarial nets," in *55th ACM/ESDA/IEEE Des. Autom. Conf.*, IEEE, pp. 1–6 (2018).
23. X. Xu et al., "A machine learning based framework for sub-resolution assist feature generation," in *Proc. Int. Symp. Phys. Des.*, ACM, pp. 161–2016) 168).
24. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science* **313**(5786), 504–507 (2006).
25. J. Zheng and L. Peng, "An autoencoder-based image reconstruction for electrical capacitance tomography," *IEEE Sens. J.* **18**(13), 5464–5474 (2018).
26. J. Deng et al., "Semisupervised autoencoders for speech emotion recognition," *IEEE/ACM Trans. Audio Speech Language Process.* **26**(1), 31–43 (2018).
27. M. Zamini and G. Montazer, "Credit card fraud detection using autoencoder based clustering," in *9th Int. Symp. Telecommun.*, IEEE, pp. 486–491 (2018).
28. Z. Chen et al., "Autoencoder-based network anomaly detection," in *Wireless Telecommun. Symp.*, IEEE, pp. 1–5 (2018).
29. C. Mack, *Fundamental Principles of Optical Lithography: The Science of Microfabrication*, John Wiley & Sons Ltd., Chichester, England (2008).
30. C. A. Mack, "More on the mask error enhancement factor," *Microlithogr. World* **8**(4), 2 (1999).
31. A. Ng et al., "Sparse autoencoder," CS294A Lecture Notes 72, pp. 1–19 (2011).
32. D. Holden et al., "Learning motion manifolds with convolutional autoencoders," in *SIGGRAPH Asia Technical Briefs*, ACM, p. 18 (2015).
33. S. Pattanayak, "Advanced neural networks," in *Pro Deep Learning with TensorFlow*, pp. 345–392, Springer, Berkeley, California (2017).
34. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," *Lect. Notes Comput. Sci.* **9351**, 234–241 (2015).
35. T. Falk et al., "U-Net: deep learning for cell counting, detection, and morphometry," *Nat. Methods* **16**(1), 67–70 (2019).
36. S. S. Girija, "TensorFlow: large-scale machine learning on heterogeneous distributed systems," tensorflow.org (2016).
37. "ASML," https://www.asml.com
38. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *3rd Int. Conf. Learning Representations (ICLR)*, Conference Track Proceedings, San Diego, California (2015).

Ying Chen received her BS degree in microelectronics and finance from Xi'an Jiaotong University, Xi'an, China, in 2013. She is currently pursuing her PhD at the Department of Microelectronics and Solid State Electronics, Institute of Microelectronics of the Chinese Academy of Sciences. Her research interests include physical design and design for manufacturability.

Yibo Lin received his BS degree in microelectronics from Shanghai Jiaotong University in 2013, and his PhD from the Electrical and Computer Engineering Department of the University of Texas at Austin in 2018. He was a postdoctoral researcher at the same university from 2018 to 2019. He is currently an assistant professor in Center for Energy-Efficient Computing and Applications, School of EECS at Peking University. His research interests include physical design and machine learning applications.

Lisong Dong received his BS and PhD degrees from Hefei University of Technology and Beijing Institute of Technology in 2008 and 2014, respectively. Currently, he works with R&D Center of Computational Lithography as an associate professor at IMECAS. His research interests include computational lithography, lithography modeling, and optimization.

Tianyang Gai received his BS degree in physics and microelectronics from Shandong University, Qingdao, China, in 2016. He is currently pursuing his MS degree at the Department of Microelectronics and Solid State Electronics, Institute of Microelectronics of the Chinese Academy of Sciences. His research interests include physical design and design for manufacturability.

Rui Chen received his PhD in electrical engineering from University at Buffalo in 2015. He then joined the Advanced Technology Development Department of GlobalFoundries as a senior photolithography engineer. His work at GlobalFoundries was focused on the 7nm backend-of-line (BEOL) patterning development. He is currently an associate professor at IMECAS, and his research is focused on the advanced node computational lithography modeling, etch, and ALD/ALE modeling.

Yajuan Su received her BS and MS degrees in microelectronics from University of Electronic Science and Technology of China in 1995 and 1998 respectively, and her PhD in microelectronics from Tsinghua University, in 2005. She is currently an associate professor in the Institute of Microelectronics of the Chinese Academy of

Sciences. Her research interests include graphene devices and MEMS/NEMS switching devices.

Yayi Wei received his MS degrees in electrics from the Institute of Electrics, Chinese Academy of Sciences, in 1992 and his PhD from Max Planck Institute for Solid State Research/University Stuttgart in 1998. Currently, he is a professor in the Institute of Microelectronics of the Chinese Academy of Sciences. His research interests include immersion lithography process and computational lithography, lithography materials, and equipment.

David Z. Pan received his BS degree from Peking University and his MS and PhD degrees from University of California, Los Angeles (UCLA). From 2000 to 2003, he was a research staff member with IBM T. J. Watson Research Center. He is currently the Engineering Foundation Professor at the Department of Electrical and Computer Engineering, The University of Texas at Austin. His research interests include cross-layer nanometer integrated circuits design for manufacturability, reliability, security, and physical design.