Research Article | Vol. 28, No. 14 / 6 July 2020 / Optics Express 20404
Optics EXPRESS

Fast inverse lithography based on dual-channel model-driven deep learning

XU MA,[1,*] XIANQIANG ZHENG,[1] AND GONZALO R. ARCE[2]

[1]Key Laboratory of Photoelectronic Imaging Technology and System of Ministry of Education of China, School of Optics and Photonics, Beijing Institute of Technology, Beijing, 100081, China
[2]Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716, USA
[*]maxu@bit.edu.cn

Abstract: Inverse lithography technology (ILT) is extensively used to compensate image distortion in optical lithography systems by pre-warping the photomask at the pixel scale. However, computational complexity is always a central challenge of ILT due to the big throughput of data volume. This paper proposes a dual-channel model-driven deep learning (DMDL) method to overcome the computational burden, while break through the limit of image fidelity over traditional ILT algorithms. The architecture of DMDL network is not inherited from conventional deep learning, but derived from the inverse optimization model under a gradient-based ILT framework. A dual-channel structure is introduced to extend the capacity of the DMDL network, which allows to simultaneously modify the mask contour and insert sub-resolution assist features to further improve the lithography image fidelity. An unsupervised training strategy based on auto-decoder is developed to avoid the time-consuming labelling process. The superiority of DMDL over the state-of-the-art ILT method is verified in both of the computational efficiency and image fidelity obtained on the semiconductor wafer.

1. Introduction

Inverse lithography is an important technology used to improve the imaging performance of optical lithography system during the manufacturing process of integrated circuits. Figure 1(a) illustrates the diagram of a typical optical lithography system with deep ultraviolet (DUV) illumination [1,2]. The light rays uniformly illuminate the photomask that carries the layout pattern of integrated circuits, and then the graphic information of the layout is transferred to the wafer through the projection optics. The photochemical reaction occurs when the photoresist on the wafer is irradiated by the light rays. Finally, the layout pattern is printed on the wafer after the development of photoresist. Image fidelity in lithography must be strictly controlled to increase the yield of semiconductor devices [2,3].

As the critical dimensions (CD) of integrated circuits continue to shrink in pace with Moore's law [4], inverse lithography technology (ILT) has been widely utilized to offset lithographic image distortions induced by adverse effects such as optical proximity effect and so on [3,5,6]. ILT grids the mask pattern into an array of pixels, and then inversely optimizes the transmittances of all mask pixels to improve the image quality of lithography system [7,8]. As shown in Fig. 1(b), ILT is capable of simultaneously optimizing the freeform mask pattern that consists of the main features and the surrounding sub-resolution assist features (SRAF). Pixel-based ILT can effectively increase the degrees of optimization freedom, but on the other hand it poses a big challenge on the computational complexity, since a large amount of variables are involved in the optimization problem [9].

In the past, gradient-based algorithms became the predominant mathematical tools to solve for the ILT problem in electronic design automation (EDA) software [10,11]. Different variations of the gradient-based algorithms were developed to solve for the ILT problem [6,8,12–18]. In addition, machine learning techniques, such as neural networks [19,20], support vector machine

#396661 https://doi.org/10.1364/OE.396661
Journal © 2020 Received 1 May 2020; revised 12 Jun 2020; accepted 18 Jun 2020; published 24 Jun 2020

Optics EXPRESS

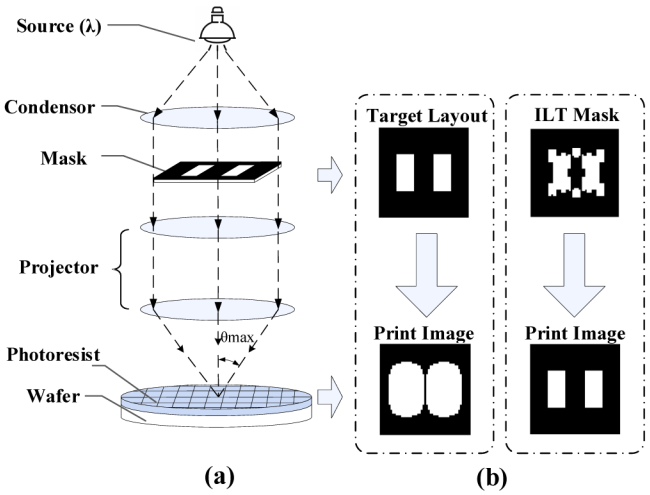

Fig. 1. (a) Diagram of the optical lithography system, where the photomask is illuminated by the light source, and the layout pattern is transferred to the wafer through the projection optics. (b) Mask optimization method based on ILT, where the transmittances of all mask pixels are optimized to improve the image quality of lithography system.

[21], and nonparametric kernel regression [22] have been introduced to improve the efficiency of ILT algorithms. The ILT problem is challenging since it is characterized with a high degree of nonlinearity attributed to the nonlinear imaging model and nonlinear photoresist effects [9,16]. However, most traditional machine learning techniques are either inadequate to accurately synthesize the nonlinear mapping between the target layouts and the ILT solutions, or they consume a large amount of computing resources in supervised training procedures.

In the latest decade, deep learning has become a driving force in machine learning and artificial intelligence [23–25]. Increasing the depth in deep neural networks effectively improves its capacity to fit any complex nonlinear function and make accurate decision or prediction. With its apparent advantages, deep learning has been introduced in the lithography realm. These works include but are not limited to the automatic hotspot correction based on the cycle-consistent generative adversarial network [26], mask defect characterization based on the convolutional neural network [27], mask design using the OPC-oriented generative adversarial network [28], and layout design using the variational autoencoder [29].

Recently, a new kind of neural network, namely model-driven convolutional neural network (MCNN), was developed to efficiently solve for the ILT problem [30]. Different from conventional deep learning methods, MCNN uses a systematic manner to derive and initialize the network architecture and parameters from the inverse optimization model. It was shown that MCNN can greatly reduce the computational time of the mask optimization procedure. However, the MCNN method has its own limitations. First, MCNN treats the entire mask pattern as a single image. Thus, it can only predict the optimized main features on the mask, but falls short to insert SRAFs around the main features. Generation of SRAFs is known as an important characteristic of the ILT methods, since the SRAFs are indeed necessary to achieve preferable lithography image fidelity at the advanced semiconductor technology nodes [31]. Due to this limitation, the MCNN can only provide an approximate guess of the ILT solution, which needs to be further optimized by the subsequent approach of a gradient-based algorithm. The second drawback is the shallow structure of the MCNN. For the ILT problem, the prediction capacity of MCNN cannot be further improved as the network includes more than a couple of layers. Thus, it is inadequate to take full

Optics EXPRESS

advantages of the deep structure to extract the underlying features or latent relationship from the input data.

This paper proposes a new kind of dual-channel model-driven deep learning (DMDL) approach that achieves superior prediction capacity compared to the state-of-the-art MCNN. The DMDL method benefits from the explicit physical meaning of its network architecture, and the remarkable ability to predict the optimization results for both main features and SRAFs on the mask pattern. Basically, the mask pattern is represented as the superposition of the main feature pattern and the SRAF pattern. Accordingly, the data flow is separated into two channels, which are used to predict the main features and SRAFs, respectively. At the end of each convolution layer, a connection route is used to fuse the data from both channels to synthesize the entire mask. Then, the fused data is used as a part of the input to the next layer. The dual-channel data flow enables the DMDL network to directly predict ILT solutions which include SRAFs.

It is important to note that the DMDL architecture exploits both, the advantages of conventional deep learning and the inverse optimization model under the gradient-based ILT framework. This paper thus systematically develops a method to derive and initialize the architecture and parameters of DMDL network. In particular, the mathematical model of the ILT problem is first established based on the imaging model of the lithography system. Subsequently, the gradient-based ILT algorithm is unfolded and truncated to form the initial architecture of the DMDL network, where each iteration in the gradient-based algorithm is replaced by one convolution layer. The paper also develops an unsupervised training method for the DMDL network to avoid the time-consuming labelling operation on the training samples. In the training process, the DMDL network is considered an encoder that transfers the target layout to the corresponding ILT solution. As the counterpart, an auto-decoding module is established based on the lithography imaging model to invert the transformation of the encoder. Then, the network parameters are optimized to minimize the distance between the input of the DMDL network and the output of the decoder.

In addition, the proposed DMDL method successfully extends the depth of the network up to dozens of layers. It is found that the residual error of the traditional MCNN is propagated back through one path [30], so extending the network's depth will cause the vanishing gradient problem [32]. However, the proposed DMDL propagates the residual error back through multiple paths, thus effectively solves this problem. Benefiting from the deep structure of the network, the prediction capacity of DMDL is effectively improved. It shows that the DMDL can directly generate promising ILT solutions for simple layouts without any auxiliary refining process as it is needed in [30]. The resulting image fidelity is superior over those obtained by conventional gradient-based methods and by the MCNN method. For complex layouts, the DMDL can provide an approximate guess of the ILT solution. Aided by a few iterations of mask refining process, the DMDL method still outperforms other ILT methods. The simulations also show that the DMDL approach can improve the computational efficiency up to an order of magnitude compared to the conventional gradient-based algorithm. Due to the length limit, this paper is mainly focusing on the coherent lithography system. It is worth noting that the proposed method can be generalized to partial coherent lithography systems as will be explained later.

The rest of this paper is organized as follows. The inverse optimization model for ILT problem is introduced in Section 2. The formulation of DMDL network is described in Section 3. The unsupervised training method is developed in Section 4. The simulation results and analysis are provided in Section 5. The paper is summarized in Section 6.

2. Inverse optimization model for ILT problem

The imaging process of coherent lithography system is described in Fig. 2. As the input of the model, $\mathbf{M}$ is an $N \times N$ binary matrix representing the pixelated mask pattern, where the zero-valued pixels (grey area) and one-valued pixels (white area) represent the opaque and

Optics EXPRESS

transparent regions, respectively. The mask pattern can be represented as the superposition of the main feature pattern $\mathbf{M}_m$ and the SRAF pattern $\mathbf{M}_s$. The main features refer to the major geometries on the mask that are distorted from the target layout. The SRAFs refer to the minor assistant openings distributed around the main features. The SRAFs themselves do not print on the wafer, but they have an important impact on the image quality of main features. However, the main feature patterns and SRAF patterns may overlap each other during the optimization procedure. It is noted that most of lithography masks in real application have binary pixel values. In order to restrict the mask pattern to be binary, the entire mask is formulated as

$$\mathbf{M}_b = \Gamma\{\mathbf{M}_g - t_m\} = \Gamma\{(\mathbf{M}_m + \mathbf{M}_s) - t_m\}, \tag{1}$$

where $\mathbf{M}_g$ represents the grey-scale mask before thresholding, $\mathbf{M}_b$ represents the binary mask pattern, $\Gamma\{\cdot\}$ is the hard threshold function, and $t_{\mathrm{m}} = 0.5$ is the threshold value. Although this paper only considers the case of binary masks, the proposed method can be generalized to handle phase-shifting masks [33,34]. In order to use gradient-based algorithms to train the DMDL network, we use the differentiable sigmoid function to replace the hard threshold function, and modify Eq. (1) as following:

$$\mathbf{M}_b = \mathrm{sig}_m(\mathbf{M}_g, t_m) = \frac{1}{1 + \exp[-a_m(\mathbf{M}_m + \mathbf{M}_s - t_m)]}, \tag{2}$$

where a_m is the steepness index of the sigmoid function.

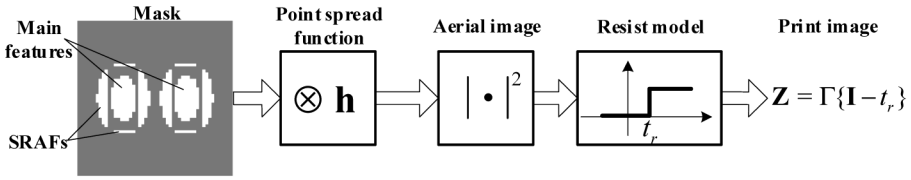

Fig. 2. The imaging process of coherent optical lithography system (Ref. [30], Fig. 2).

According to the Hopkins diffraction model, the aerial image on the wafer can be expressed as [3,35]

$$\mathbf{I} = |\mathbf{M}_b \otimes \mathbf{h}|^2. \tag{3}$$

In the above equation, $\mathbf{I}$ is the aerial image representing the distribution of light intensity on the top of wafer, $\otimes$ is the convolution operator, and $\mathbf{h}$ represents the point spread function of the lithography system given by

$$\mathbf{h} = \frac{J_1(2\pi rNA/\lambda)}{2\pi rNA/\lambda}, \tag{4}$$

where $J_1(\cdot)$ is the Bessel function of the first kind, r is the distance from a point of $\mathbf{h}$ to the center, NA is the numerical aperture of projector, and λ is the illumination wavelength. The point spread function $\mathbf{h}$ is normalized to have unit energy. The photoresist effect can be approximately depicted by a simple hard threshold function [3,6]. After the development of photoresist, the print image on the wafer can be calculated as

$$\mathbf{Z} = \Gamma\{\mathbf{I} - t_r\} = \Gamma\{|\mathbf{M}_b \otimes \mathbf{h}|^2 - t_r\}, \tag{5}$$

where t_r is the threshold of photoresist. Similar to Eq. (2), we use the sigmoid function to replace the hard threshold function, and Eq. (5) is adjusted as follows

$$\mathbf{Z} \approx \mathrm{sig}_r(\mathbf{I}, t_r) = \frac{1}{1 + \exp[-a_r(\mathbf{I} - t_r)]}, \tag{6}$$

where a_r is the steepness index.

Optics EXPRESS

In this paper, the goal of ILT is to minimize the residual error between the target layout $\tilde{\mathbf{Z}}$ and the actual print image $\mathbf{Z}$. Therefore, the cost function can be formulated as

$$F = ||\tilde{\mathbf{Z}} - \mathbf{Z}||_2^2 \approx ||\tilde{\mathbf{Z}} - \mathrm{sig}_r(\mathbf{I}, t_r)||_2^2, \tag{7}$$

where $\|\cdot\|_2$ represents the l_2-norm. The inverse optimization problem of ILT is modelled as

$$\{\hat{\mathbf{M}}_m, \hat{\mathbf{M}}_s\} = \arg\min_{\mathbf{M}_m, \mathbf{M}_s} F, \tag{8}$$

where $\hat{\mathbf{M}}_m$ and $\hat{\mathbf{M}}_s$ represent the optimized main feature pattern and SARF pattern, respectively.

3. Formulation of DMDL network

This section derives the architecture and formulation of the proposed DMDL network based on the inverse optimization model described in Eq. (8). Based on the conventional gradient-based algorithm, Eq. (8) can be solved by iteratively updating the mask patterns as follows [3,16]:

$$\mathbf{M}_m^{k+1} = \mathbf{M}_m^k - step \cdot \nabla F|_{\mathbf{M}_m}, \tag{9}$$

$$\mathbf{M}_s^{k+1} = \mathbf{M}_s^k - step \cdot \nabla F|_{\mathbf{M}_s}, \tag{10}$$

where *step* is the step length;$\nabla F|_{\mathbf{M}_m}$ and $\nabla F|_{\mathbf{M}_s}$ respectively represent the gradients of the cost function with respect to $\mathbf{M}_m$ and $\mathbf{M}_s$, where the superscripts indicate the iteration number. According to Eqs. (2)–(8), the gradients can be calculated as

$$\nabla F|_{\mathbf{M}_m} = \nabla F|_{\mathbf{M}_s} = -4a_r \cdot a_m \cdot \{[(\tilde{\mathbf{Z}} - \mathbf{Z}) \odot \mathbf{Z} \odot (\mathbf{1} - \mathbf{Z}) \odot (\mathbf{M}_b \otimes \mathbf{h})] \otimes \mathbf{h}^{\circ} \odot \mathbf{M}_b \odot (\mathbf{1} - \mathbf{M}_b)\}, \tag{11}$$

where $\odot$ is the Hadamard product, $\mathbf{1} \in \mathbb{R}^{N \times N}$ is the one-valued matrix, $\mathbf{h}^{\circ}$ means to rotate the matrix $\mathbf{h}$ by 180° in both horizontal and vertical directions. The flowchart of the dual-channel gradient-based ILT algorithm is shown in Fig. 3. According to Eq. (11), we can define the following variables:

$$\mathbf{S}_m = \mathbf{S}_s = \delta(x, y),\ \mathbf{D}_m = \mathbf{D}_s = \mathbf{h},\ \mathbf{T} = (\tilde{\mathbf{Z}} - \mathbf{Z}) \odot \mathbf{Z} \odot (\mathbf{1} - \mathbf{Z}),\ \mathbf{W}_m = \mathbf{W}_s = \mathbf{h}^{\circ}, \text{ and } Q = 4a_r \cdot a_m \cdot step, \tag{12}$$

where $\delta(x, y)$ is the Dirac delta function. Figure 3 gives the intuitive illustration of the iteration process. It shows that the data flow is separated into two channels, which are used to update the main feature pattern and SRAF pattern, respectively.

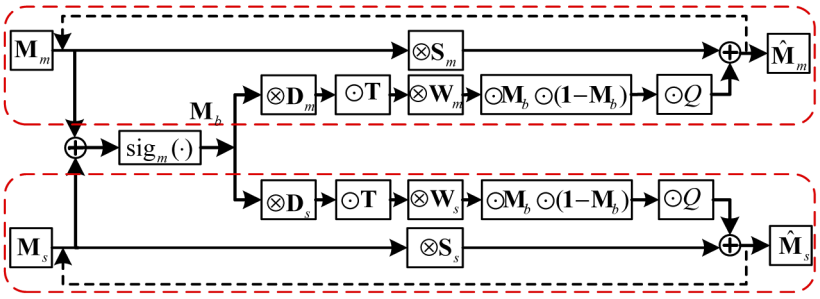

Fig. 3. The flowchart of the dual-channel gradient-based ILT algorithm, where the two channels are respectively used to update the main features and SRAFs.

The principle of DMDL is to construct the network architecture by unfolding and truncating the dual-channel gradient-based iterations. If we only retain the first K updating steps and ignore

Optics EXPRESS

the following iterations, then the architecture of DMDL network is given by Fig. 4(a). Each iteration is transferred to a layer in the network, and the convolution kernels and parameters can be initialized according to Eq. (12). The original input of the network is the target layout $\tilde{\mathbf{Z}}$, which is the expected print image to be obtained on the wafer. The final output is the corresponding ILT solution $\hat{\mathbf{M}}$. Thus, the entire DMDL network can be considered an encoder to transform the layout pattern into the ILT solution. Similar to Fig. 3, the data flow of the DMDL is separated into two channels that are used to predict the solutions of main feature pattern and SRAF pattern, respectively. At the end of each layer, a connection route is used to fuse the data from both channels to synthesize the entire mask. Then, the output main feature pattern, SRAF pattern and the synthesized mask pattern are used as the input data to the next layer. It is worth noting that one of the differences between Fig. 3 and Fig. 4(a) is that DMDL adds the sigmoid functions as the activation functions at the end of each layer. These activation functions transfer the main feature, the SRAF and the synthesized mask from grey-scale patterns to approximate binary patterns. It is noted that the activation functions in DMDL network is inherited from the sigmoid function in Eq. (2). The sigmoid function is used because its output range is (0,1), and the sigmoid function with larger the steepness factor is close to the hard threshold function. In the future, we will study the influence of other nonlinear activation functions such as tanh function, rectified linear unit (ReLU) function and so on. In addition, we will also study the strategy to use different activation functions for separate network layers. Next, we formulate the DMDL network in detail. Since the structure is the same for every layer, we only take the kth (k=1, 2, . . . , K) layer as an example. The three input matrices $\mathbf{M}_b^k$, $\mathbf{M}_{mb}^k$ and $\mathbf{M}_{sb}^k$ represent the entire mask pattern, main feature pattern and SRAF pattern, respectively. The output grey-scale matrices from the kth layer are $\mathbf{M}_{mg}^k$, $\mathbf{M}_{sg}^k$ and $\mathbf{M}_g^k$, which can be calculated as

$$\mathbf{M}_{mg}^k = \mathbf{M}_{mb}^k \otimes \mathbf{S}_m^k + [\mathbf{T}^k \odot (\mathbf{M}_b^k \otimes \mathbf{D}_m^k)] \otimes \mathbf{W}_m^k \odot \mathbf{M}_b^k \odot (\mathbf{1} - \mathbf{M}_b^k) \odot Q, \tag{13}$$

$$\mathbf{M}_{sg}^k = \mathbf{M}_{sb}^k \otimes \mathbf{S}_s^k + [\mathbf{T}^k \odot (\mathbf{M}_b^k \otimes \mathbf{D}_s^k)] \otimes \mathbf{W}_s^k \odot \mathbf{M}_b^k \odot (\mathbf{1} - \mathbf{M}_b^k) \odot Q, \tag{14}$$

$$\mathbf{M}_g^k = \mathbf{M}_{sg}^k + \mathbf{M}_{mg}^k, \tag{15}$$

where the superscript "k" represents the kth iteration. The subscript "b" indicates the output matrices from the sigmoid functions, whose elements are close to 0 or 1. The subscript "g" indicates the grey-scale matrices before the sigmoid functions. $\mathbf{S}_m^k \in \mathbb{R}^{N_f \times N_f}$, $\mathbf{S}_s^k \in \mathbb{R}^{N_f \times N_f}$, $\mathbf{D}_m^k \in \mathbb{R}^{N_f \times N_f}$, $\mathbf{D}_s^k \in \mathbb{R}^{N_f \times N_f}$, $\mathbf{W}_m^k \in \mathbb{R}^{N_f \times N_f}$ and $\mathbf{W}_s^k \in \mathbb{R}^{N_f \times N_f}$ are the convolution kernels that need to be optimized through the training process. N_f is the size of the convolution kernels, and $\mathbf{T}^k \in \mathbb{R}^{N \times N}$ is the transmission matrix defined as

$$\mathbf{T}^k = \mathbf{T} = (\tilde{\mathbf{Z}} - \mathbf{Z}) \odot \mathbf{Z} \odot (\mathbf{1} - \mathbf{Z}). \tag{16}$$

At the end of each layer we convert the grey-scale matrices to the approximate binary matrices using the sigmoid functions:

$$\mathbf{M}_{mb}^{k+1} = \mathrm{sig}_m(\mathbf{M}_{mg}^k, t_m) = \frac{1}{1 + \exp[-a_m(\mathbf{M}_{mg}^k - t_m)]}, \tag{17}$$

$$\mathbf{M}_{sb}^{k+1} = \mathrm{sig}_m(\mathbf{M}_{sg}^k, t_m) = \frac{1}{1 + \exp[-a_m(\mathbf{M}_{sg}^k - t_m)]}, \tag{18}$$

$$\mathbf{M}_b^{k+1} = \mathrm{sig}_m(\mathbf{M}_g^k, t_m) = \frac{1}{1 + \exp[-a_m(\mathbf{M}_g^k - t_m)]}. \tag{19}$$

It is noted that the sigmoid functions in Eqs. (17)–(19) can be regarded as the nonlinear activation functions, which enable the network to construct the complex nonlinear mapping

Optics EXPRESS

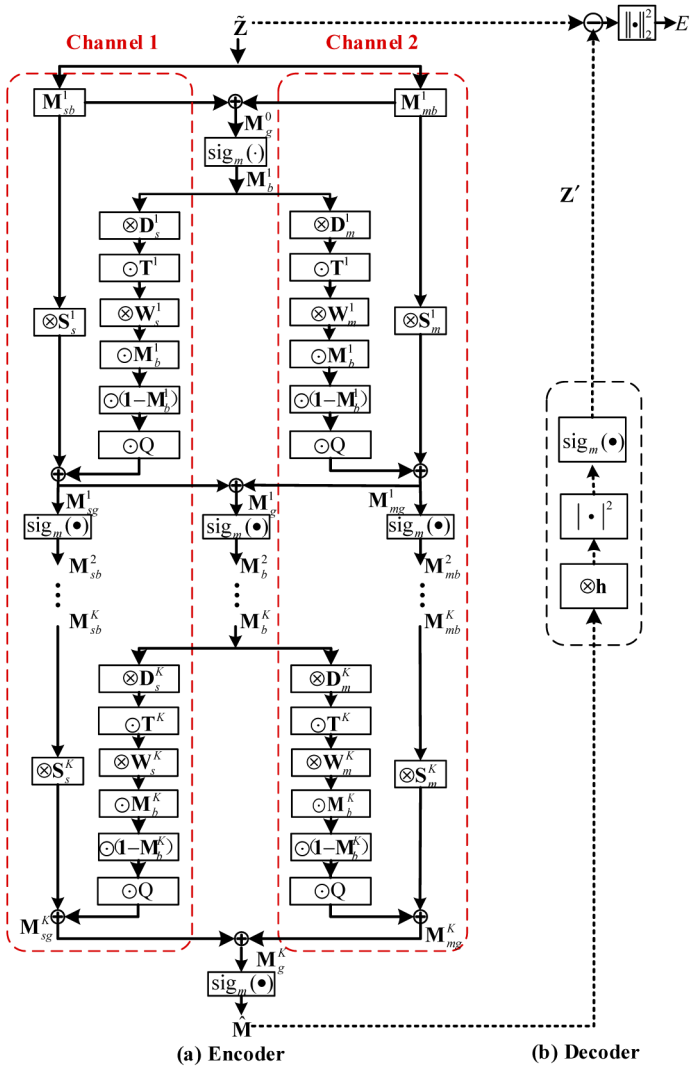

Fig. 4. (a) The architecture of the DMDL network (the encoder), and (b) the decoder used to train the DMDL network. Channel 1 and Channel 2 are used to predict the solutions of SRAF pattern and main feature pattern, respectively.

between the target layout and the ILT solution. Then, $\mathbf{M}_{mb}^{k+1}$, $\mathbf{M}_{sb}^{k+1}$ and $\mathbf{M}_{b}^{k+1}$ are used as the input matrices for the $k+1$th layer. The final output of the DMDL network is $\hat{\mathbf{M}} = \mathrm{sig}_m(\mathbf{M}_g^K, t_m)$, which is an approximate binary mask pattern. When the training process is completed, we should use a hard threshold function to replace the sigmoid function at the end of the last layer, such that the predicted ILT solution is an absolute binary mask.

It is worth noting that the formulation above focuses on coherent lithography systems, but the proposed DMDL method can be extended to partially coherent lithography systems with scalar imaging model and vector imaging model. According to the scalar Abbe's imaging model, the aerial image of the partially coherent lithography system can be formulated as [3,16]

$$\mathbf{I} = \sum_{x_s, y_s} \mathbf{J}(x_s, y_s) |\mathbf{M}_b \otimes \mathbf{h}^{x_s y_s}|^2, \tag{20}$$

Optics EXPRESS

where $\mathbf{J} \in \mathbb{R}^{N_s \times N_s}$ represents the source pattern, $\mathbf{J}(x_s, y_s)$ is the light intensity of the source point at the coordinate (x_s, y_s), and $\mathbf{h}^{x_s y_s}$ is the point spread function corresponding to the source point (x_s, y_s). The aerial image of the partially coherent lithography system is formulated as the summation of the coherent aerial images contributed by all source points. Substituting Eq. (20) into Eqs. (6) and (7), we can derive the gradients of cost function as following

$$\begin{aligned}\nabla F|_{\mathbf{M}_m} = \nabla F|_{\mathbf{M}_s} \quad = -4a_r \cdot a_m \cdot \sum_{x_s, y_s} \mathrm{Real}\{\mathbf{J}(x_s, y_s)(\tilde{\mathbf{Z}} - \mathbf{Z}) \odot \mathbf{Z} \odot (\mathbf{1} - \mathbf{Z}) \\ \odot (\mathbf{M}_b \otimes \mathbf{h}^{x_s y_s *}) \otimes (\mathbf{h}^{x_s y_s})^{*\circ} \odot \mathbf{M}_b \odot (\mathbf{1} - \mathbf{M}_b)\}\end{aligned}, \tag{21}$$

where $\mathrm{Real}\{\cdot\}$ indicates the real part of the argument, and $*$ is the conjugate operation. According to Eq. (21), we can define the following variables:

$$\begin{aligned}\mathbf{S}_m = \mathbf{S}_s = \delta(x, y),\ \mathbf{D}_m^{x_s y_s} = \mathbf{D}_s^{x_s y_s} = (\mathbf{h}^{x_s y_s})^{*\circ},\ \mathbf{T} = (\tilde{\mathbf{Z}} - \mathbf{Z}) \odot \mathbf{Z} \odot (\mathbf{1} - \mathbf{Z}),\ \mathbf{W}_m^{x_s y_s} = \mathbf{W}_s^{x_s y_s} = \mathbf{h}^{x_s y_s *}, \\ \text{and } Q = 4a_r \cdot a_m \cdot step,\end{aligned} \tag{22}$$

where *step* is the step length; the defined notations represent the convolution kernels and weighting parameters in the DMDL network.

In addition, the proposed DMDL method can also be applied to vector lithography imaging model. The aerial image under vector imaging model is [36]

$$\mathbf{I} = \sum_{x_s, y_s} \left[\mathbf{J}(x_s, y_s) \sum_{p=x,y,z} |(\mathbf{M}_b \odot \mathbf{B}^{x_s, y_s}) \otimes \mathbf{H}_p^{x_s, y_s}|^2 \right], \tag{23}$$

where $\mathbf{B}^{x_s, y_s}$ denotes the mask diffraction matrix associated with the point source (x_s, y_s), and $\mathbf{H}_p^{x_s, y_s} (p = x, y, z)$ denotes the equivalent point spread functions of the x, y, and z components. Substituting Eq. (23) into Eqs. (6) and (7), we can derive the gradients of cost function as follows

$$\begin{aligned}\nabla F|_{\mathbf{M}_m} = \nabla F|_{\mathbf{M}_s} \quad = -4a_r \cdot a_m \cdot \sum_{x_s, y_s} \mathrm{Real} \Bigg\{ \mathbf{J}(x_s, y_s) \sum_{p=x,y,z} (\tilde{\mathbf{Z}} - \mathbf{Z}) \odot \mathbf{Z} \odot (\mathbf{1} - \mathbf{Z}) \odot (\mathbf{M}_b \odot \mathbf{B}^{x_s, y_s *}) \\ \otimes \mathbf{H}_p^{x_s, y_s *} \otimes (\mathbf{H}_p^{x_s, y_s})^{*\circ} \odot \mathbf{B}^{x_s, y_s *} \odot \mathbf{M}_b \odot (\mathbf{1} - \mathbf{M}_b) \Bigg\}\end{aligned}. \tag{24}$$

According to Eq. (24), we can define the following variables:

$$\begin{aligned}&\mathbf{S}_m = \mathbf{S}_s = \delta(x, y),\ \mathbf{D}_m^{x_s y_s} = \mathbf{D}_s^{x_s y_s} = (\mathbf{H}_p^{x_s, y_s})^{*\circ},\ \mathbf{T} = (\tilde{\mathbf{Z}} - \mathbf{Z}) \odot \mathbf{Z} \odot (\mathbf{1} - \mathbf{Z}), \\ &\mathbf{W}_m^{x_s y_s} = \mathbf{W}_s^{x_s y_s} = \mathbf{H}_p^{x_s, y_s *},\ \mathbf{G}_m^{x_s, y_s} = \mathbf{G}_s^{x_s, y_s} = \mathbf{B}^{x_s, y_s *},\ \text{and } Q = 4a_r \cdot a_m \cdot step\end{aligned}, \tag{25}$$

where the defined notations represent the convolution kernels and weighting parameters within the DMDL network.

Similar to the coherent lithography system, we can derive the DMDL network for the partially coherent lithography system with scalar and vector imaging models by unfolding and truncating the iteration process based on the above formulae. Compared to the coherent lithography system, the DMDL network of partially coherent lithography system has more parameters to be trained, but the principle to derive the network framework is similar to the coherent system. Due to the length limit of this paper, the DMDL approach for the partially coherent lithography systems will be described and studied in the future.

4. Unsupervised training method

Traditional supervised training methods require time and lots of computing resources to label the large number of training samples. To circumvent this problem, this paper proposes an

Optics EXPRESS

unsupervised training method to optimize the parameters of the DMDL network. The goal of ILT is to pursue the optimized mask pattern that reduces the pattern error between the print image and target layout as small as possible. As mentioned above, the DMDL network is considered an encoder to transform the target layout to the ILT solution. On the other hand, the imaging model of lithography system can be considered an auto-decoder module to transform the mask pattern to the corresponding print image on the wafer. As shown in Fig. 4(b), we concatenate the encoder with the decoder module, and the output of encoder is used as the input of decoder. Based on Eqs. (3) and (6), the formulation of decoder is given by

$$\mathbf{Z}' \approx \mathrm{sig}_r(|\hat{\mathbf{M}} \otimes \mathbf{h}|^2, t_r) = \frac{1}{1 + \exp[-a_r(|\hat{\mathbf{M}} \otimes \mathbf{h}|^2 - t_r)]}. \tag{26}$$

Ideally, the print image of the ILT solution should be very close to the target layout. Thus, the cost function of the training process can be formulated as

$$E = ||\tilde{\mathbf{Z}} - \mathbf{Z}'||_2^2 \approx ||\tilde{\mathbf{Z}} - \mathrm{sig}_r(|\hat{\mathbf{M}} \otimes \mathbf{h}|^2, t_r)||_2^2, \tag{27}$$

where $\tilde{\mathbf{Z}}$ is the target layout, and $\mathbf{Z}'$ is described in Eq. (26).

The purpose of the training process is to find the optimal combination of the convolution kernels $\mathbf{S}_m^k, \mathbf{D}_m^k, \mathbf{W}_m^k, \mathbf{S}_s^k, \mathbf{D}_s^k$ and $\mathbf{W}_s^k$ to minimize the residual error between $\tilde{\mathbf{Z}}$ and $\mathbf{Z}$. Thus, the training problem of DMDL is formulated as

$$\{\hat{\mathbf{S}}_m^k, \hat{\mathbf{D}}_m^k, \hat{\mathbf{W}}_m^k, \hat{\mathbf{S}}_s^k, \hat{\mathbf{D}}_s^k, \hat{\mathbf{W}}_s^k\} = arg \min_{\hat{\mathbf{S}}_m^k, \hat{\mathbf{D}}_m^k, \hat{\mathbf{W}}_m^k, \hat{\mathbf{S}}_s^k, \hat{\mathbf{D}}_s^k, \hat{\mathbf{W}}_s^k} E = arg \min_{\hat{\mathbf{S}}_m^k, \hat{\mathbf{D}}_m^k, \hat{\mathbf{W}}_m^k, \hat{\mathbf{S}}_s^k, \hat{\mathbf{D}}_s^k, \hat{\mathbf{W}}_s^k} ||\tilde{\mathbf{Z}} - \mathbf{Z}'||_2^2. \tag{28}$$

In order to constrain the output of DMDL as close to the binary mask as possible, we add a quadratic penalty term in the cost function [3,6]. The quadratic penalty term is defined as

$$R_Q = \mathbf{1}_{N\times 1}^T \cdot 4\hat{\mathbf{M}} \odot (\mathbf{1} - \hat{\mathbf{M}}) \cdot \mathbf{1}_{N\times 1}, \tag{29}$$

where $\mathbf{1}_{N\times 1}$ is the one-valued vector with dimension of $N \times 1$. Thus, Eq. (28) is modified as

$$\{\hat{\mathbf{S}}_m^k, \hat{\mathbf{D}}_m^k, \hat{\mathbf{W}}_m^k, \hat{\mathbf{S}}_s^k, \hat{\mathbf{D}}_s^k, \hat{\mathbf{W}}_s^k\} = arg \min_{\hat{\mathbf{S}}_m^k, \hat{\mathbf{D}}_m^k, \hat{\mathbf{W}}_m^k, \hat{\mathbf{S}}_s^k, \hat{\mathbf{D}}_s^k, \hat{\mathbf{W}}_s^k} E' = arg \min_{\hat{\mathbf{S}}_m^k, \hat{\mathbf{D}}_m^k, \hat{\mathbf{W}}_m^k, \hat{\mathbf{S}}_s^k, \hat{\mathbf{D}}_s^k, \hat{\mathbf{W}}_s^k} \{||\tilde{\mathbf{Z}} - \mathbf{Z}'||_2^2 + \gamma_Q \cdot R_Q\}, \tag{30}$$

where γ_Q is the weight of quadratic penalty. It is remarkable that we only need to input the training layout patterns into the network in turn, and then the DMDL will automatically perform unsupervised training without any labelling process.

This paper applies the back-propagation algorithm to optimize the convolution kernels [37]. The initial values of the convolution kernels are set according to Eq. (12). Then, we need to calculate the gradients of the cost function E' with respect to each convolution kernel. Let $\mathbf{A}_m^k$ represent the convolution kernels $\mathbf{S}_m^k$, $\mathbf{D}_m^k$ or $\mathbf{W}_m^k$, and let $\mathbf{A}_s^k$ represent the convolution kernels $\mathbf{S}_s^k$, $\mathbf{D}_s^k$ or $\mathbf{W}_s^k$. According to the Chain rule, the partial derivatives of the cost function E' to $\mathbf{A}_m^k$ and $\mathbf{A}_s^k$ can be calculated as

$$\frac{\partial E'}{\partial \mathbf{A}_m^k} = \frac{\partial E'}{\partial \hat{\mathbf{M}}} \cdot \frac{\partial \hat{\mathbf{M}}}{\partial \mathbf{M}_g^K} \cdot \left(\frac{\partial \mathbf{M}_{mg}^K}{\partial \mathbf{M}_{mb}^K} \cdot \frac{\partial \mathbf{M}_{mb}^K}{\partial \mathbf{M}_{mg}^{K-1}} + \frac{\partial \mathbf{M}_{mg}^K}{\partial \mathbf{M}_b^K} \cdot \frac{\partial \mathbf{M}_b^K}{\partial \mathbf{M}_{mg}^{K-1}} + \frac{\partial \mathbf{M}_{sg}^K}{\partial \mathbf{M}_b^K} \cdot \frac{\partial \mathbf{M}_b^K}{\partial \mathbf{M}_{mg}^{K-1}} \right) \cdots \frac{\partial \mathbf{M}_{mg}^k}{\partial \mathbf{A}_m^k}, \tag{31}$$

$$\frac{\partial E'}{\partial \mathbf{A}_s^k} = \frac{\partial E'}{\partial \hat{\mathbf{M}}} \cdot \frac{\partial \hat{\mathbf{M}}}{\partial \mathbf{M}_g^K} \cdot \left(\frac{\partial \mathbf{M}_{sg}^K}{\partial \mathbf{M}_{sb}^K} \cdot \frac{\partial \mathbf{M}_{sb}^K}{\partial \mathbf{M}_{sg}^{K-1}} + \frac{\partial \mathbf{M}_{sg}^K}{\partial \mathbf{M}_b^K} \cdot \frac{\partial \mathbf{M}_b^K}{\partial \mathbf{M}_{sg}^{K-1}} + \frac{\partial \mathbf{M}_{mg}^K}{\partial \mathbf{M}_b^K} \cdot \frac{\partial \mathbf{M}_b^K}{\partial \mathbf{M}_{sg}^{K-1}} \right) \cdots \frac{\partial \mathbf{M}_{sg}^k}{\partial \mathbf{A}_s^k}. \tag{32}$$

As mentioned in the Introduction, one of the important merits of the DMDL network is to propagate the residual error back through multiple paths so as to alleviate the vanishing gradient

Optics EXPRESS

problem. The innermost parenthesis in Eq. (31) includes three terms, such as $\partial \mathbf{M}_{sg}^K/\partial \mathbf{M}_b^K \cdot \partial \mathbf{M}_b^K/\partial \mathbf{M}_{mg}^{K-1}$, $\partial \mathbf{M}_{mg}^K/\partial \mathbf{M}_b^K \cdot \partial \mathbf{M}_b^K/\partial \mathbf{M}_{mg}^{K-1}$ and $\partial \mathbf{M}_{mg}^K/\partial \mathbf{M}_{mb}^K \cdot \partial \mathbf{M}_{mb}^K/\partial \mathbf{M}_{mg}^{K-1}$. That means that the residual error is propagated back through three paths. This can be also interpreted from the network architecture in Fig. 4(a). It shows that each layer of the DMDL network consists of three parallel routes. The first path and the second path correspond to Channel 1 and Channel 2, respectively. The third path corresponds to the data fusion route.

Appendix A provides the details on how to calculate the partial derivatives in Eqs. (31) and (32). Then, the convolution kernels are updated iteratively according to the following formula:

$$\mathbf{A}_m^{k(n+1)} = \mathbf{A}_m^{k(n)} - step_A \cdot \nabla E'|_{\mathbf{A}_m^k}, \tag{33}$$

$$\mathbf{A}_s^{k(n+1)} = \mathbf{A}_s^{k(n)} - step_A \cdot \nabla E'|_{\mathbf{A}_s^k}, \tag{34}$$

where $step_A$ is the step size;$\nabla E'|_{\mathbf{A}_m^k}$ and $\nabla E'|_{\mathbf{A}_s^k}$ are the gradients of E' to $\mathbf{A}_m^k$ and $\mathbf{A}_s^k$, respectively. After each iteration, we enforce the convolution kernels to be symmetric along the x-axis and y-axis. Then, we normalize the convolution kernels by their l_2-norm such that $\mathbf{A}_m^{k(n+1)} = \mathbf{A}_m^{k(n+1)}/\|\mathbf{A}_m^{k(n+1)}\|_2$. After the training process is completed, the decoder is removed from the encoder, and the DMDL network can be used to predict the ILT solution for other testing layout patterns.

5. Simulation and analysis

This section provides the simulation results to verify the advantages of the proposed DMDL method. In Section 5.1, we will train and test the DMDL network based on a set of simple layout patterns. In addition, Section 5.2 provides the simulations based on a complex layout pattern.

5.1. Simulations based on simple layout patterns

In the following simulations, we use a DUV coherent lithography system with the illumination wavelength of $\lambda = 193$ nm. The CD of the target layout is 45 nm on the wafer scale. The numerical aperture of the projection optics is 1.35. The lateral size of the matrix $\mathbf{M}$ is $N = 51$. The lateral size of the point spread function $\mathbf{h}$ and the convolution kernels $\mathbf{S}_m^k, \mathbf{D}_m^k, \mathbf{W}_m^k, \mathbf{S}_s^k, \mathbf{D}_s^k$ and $\mathbf{W}_s^k$ is $N_f = 11$. The size of each pixel is 4.5 nm on the wafer scale. The steepness indexes

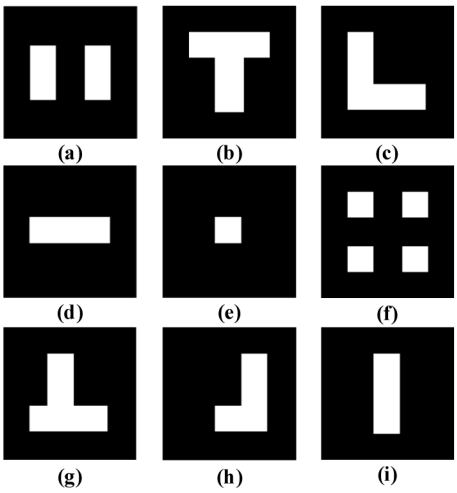

Fig. 5. The training layouts for the DMDL network.

Optics EXPRESS

in Eqs. (2) and (6) are $a_r = a_m = 10$. The threshold of photoresist in Eq. (5) is $t_r = 0.25$. The weight of the quadratic penalty in Eq. (30) is $\gamma_Q = 0.05$. The DMDL network includes 30 layers.

Figure 5 shows the training layouts used in this paper. It is worth noting that most layout patterns of integrated circuits are composed by the Manhattan geometries after normalization. The regular Manhattan geometries can be divided into three basic features, i.e., the concave corners, convex corners and straight edges [22]. The selected training data set in Fig. 5 includes a set of typical basic features that frequently appear on the layout patterns. Figure 6 shows the testing layouts used in this paper. From left to right, the testing layouts are called "Target 1", "Target 2", "Target 3" and "Target 4", respectively.

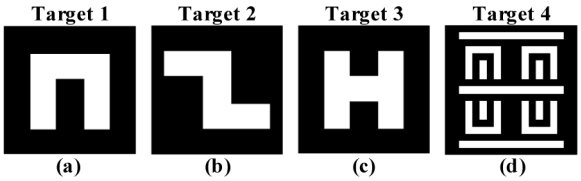

Fig. 6. The testing layouts for the DMDL network.

At the beginning of the training process, we need to initialize the main feature pattern and the SRAF pattern. Taking the two-bar shaped mask as an example, the initial main feature pattern is set to be the same as the target layout, which is represented by the white regions in Fig. 7(a). On the other hand, the initial SRAF pattern is defined as the narrow boundaries surrounding the target layout, which are shown by the shaded regions in Fig. 7(b). The width of the initial SRAF regions is set as 2 pixels in this paper. In particular, we dilate the target pattern by 2 pixels. Then, the initial SRAF regions are defined as the areas between the contours of the dilated pattern and original target pattern. We input the training layouts into the network in turn, and the step size to update the convolution kernels is $step_A = 4 \times 10^{-7}$.

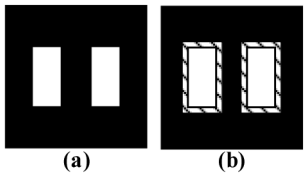

Fig. 7. (a) The initial main feature pattern (white regions), and (b) the initial SRAF pattern (shadow regions).

First, the proposed DMDL method is compared with other three methods, including the steepest descent (SD) ILT algorithm [3,6] and the MCNN methods proposed in [30]. In the SD algorithm, the step size is set to be 0.5, the maximum iteration number is 1000, and other parameters are the same as those used in DMDL. The second method in comparison is the MCNN approach without auxiliary refining process. In this approach, the output of the MCNN network is directly used as the ILT solution. The third method in comparison is the MCNN with auxiliary refining process, which is named as MCNN + SD for short [30]. In the MCNN + SD method, the output of MCNN is used as the initial guess of the ILT solution. Then, the SD algorithm is run for 300 iterations to further refine the mask patterns.

The simulation results for the three simple testing layouts are illustrated in Figs. 8, 9 and 10 respectively. In these figures, the first row and the second row show the mask patterns and the corresponding print images on the wafer. The third row shows the error patterns that indicate the differences between the target layouts and the print images, where the white, grey and black

Optics EXPRESS

colors represent 1, 0 and -1, respectively. Pattern errors (PE) of all print images are presented, where the PE is defined as the square of Euclidean distance between the print image and the target layout, i.e. PE = $\|\tilde{\mathbf{Z}} - \Gamma\{\mathbf{I} - t_r\}\|_2^2$.

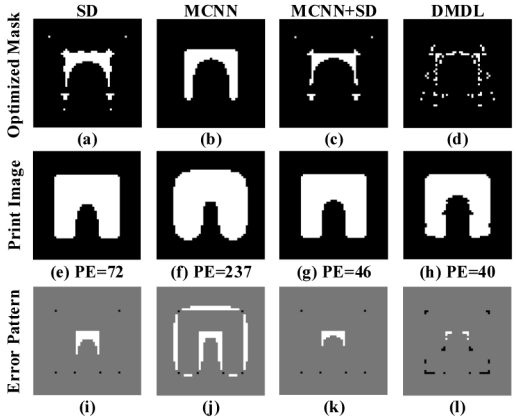

Fig. 8. Simulation results of different ILT methods using Target 1 as the testing layout.

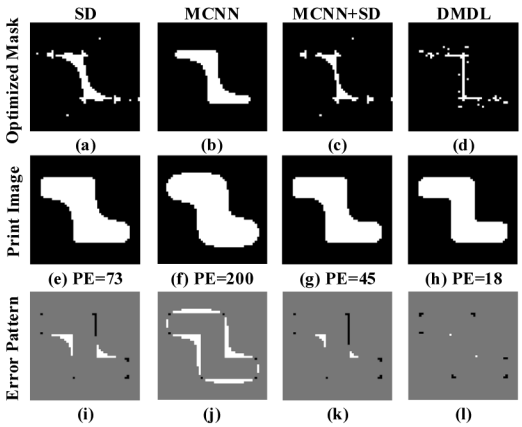

Fig. 9. Simulation results of different ILT methods using Target 2 as the testing layout.

From these simulations, we have the following observations. First, the MCNN can only obtain an approximate guess of the ILT solution, so the pattern errors (PEs) of MCNN are higher than those of the SD method. The MCNN has a shadow network structure with a single data transmission channel, and its prediction capacity is limited. In addition, the weighting parameter **T** within MCNN constrains the evolution of mask around the target pattern. Thus, the MCNN method falls short to insert SRAFs around the main features. On the other hand, the proposed DMDL approach can significantly improve the image fidelity compared to the SD and MCNN methods. What is more remarkable, the DMDL even outperforms the MCNN + SD method. As mentioned above, the MCNN + SD method uses the SD algorithm after MCNN to further refine the mask and improve the imaging performance. As a counterpart, the DMDL network can directly output the ILT solution without any following iterations, so the computational complexity is much lower than the MCNN + SD method. Another observation is that the DMDL method can automatically generate more SRAFs than other methods, thus effectively improving the image fidelity of lithography system. This characteristic of DMDL is attributed to the proposed

Optics EXPRESS

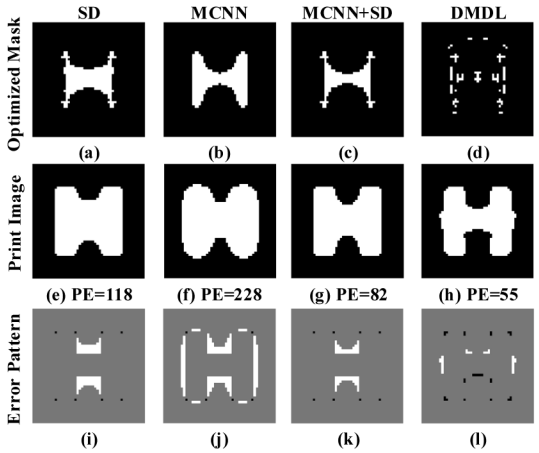

Fig. 10. Simulation results of different ILT methods using Target 3 as the testing layout.

dual-channel network architecture, which provides the data flow to predict the main feature pattern and SRAF pattern simultaneously.

Next, we compare the computational efficiency of the four ILT methods mentioned above. Table 1 lists the average runtimes over the three testing layouts for different methods. This table provides the details of runtimes including the training, testing, and iteration times for the four methods. Once the network is trained, it is not necessary to train the network again when optimizing the testing masks. Thus, the total optimization time in the last row represents the sum of testing time and the iteration time. The computational efficiency of different methods is evaluated by the total optimization time. The SD method does not include the training and testing stages, and the MCNN + SD method uses a set of iterations after the testing process to refine the mask. It is observed that the MCNN method is much faster than the DMDL method, since the DMDL includes more than 10-fold layers compared to the MCNN. But, the PEs of MCNN are much larger than those of DMDL. In contrast to the SD and MCNN + SD methods, DMDL can improve the computational efficiency by 13 times and 4.2 times, and further reduce the PEs by 57% and 35% on average, respectively. Therefore, the DMDL method achieves the best imaging performance for the three simple testing layouts, and significantly improves the computational efficiency compared to the SD and MCNN + SD methods.

Table 1. Comparison of the average runtimes based on simple testing layouts.

Algorithm	SD	MCNN	MCNN + SD	DMDL
Training time	-	0.50 s	0.50 s	8.79s
Testing time	-	0.46×10^{-2} s	0.46×10^{-2} s	4.7×10^{-2} s
Iteration time	60.88×10^{-2} s	-	19.45×10^{-2} s	-
Total optimization time	60.88×10^{-2} s	0.46×10^{-2} s	19.91×10^{-2} s	4.7×10^{-2} s

In addition, we study the effect of network depth on the performance of DMDL. Figure 11 shows the average PE of the three testing layouts obtained by DMDL with 5 to 50 layers. In this figure, the average PE is calculated for every 5 additional layers. It is observed that the average PE is first reduced by increasing the network depth, but the performance of DMDL cannot be further improved as the network includes more than 30 layers. In addition, increasing network depth means more computational complexity in both of training and testing processes. Therefore, we set the number of layers to be 30.

Optics EXPRESS

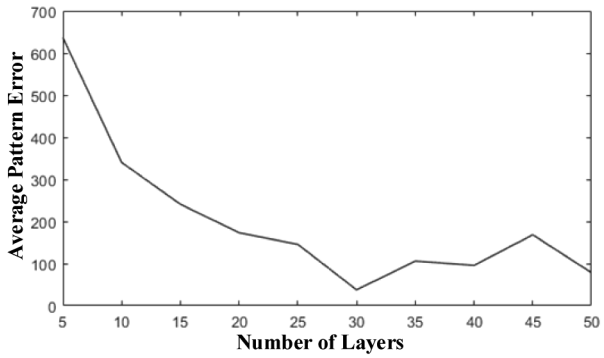

Fig. 11. The resulting average pattern error obtained by DMDL with different number of layers.

5.2. Simulations based on complex layout pattern

In this section, we use the simple layouts in Fig. 5 to train the DMDL network, but use a complex mask in Fig. 6(d) as the testing layout. The lateral dimension of complex testing layout is $N = 185$. From left to right, the first four columns in Fig. 12 show the simulation results of the SD, MCNN, MCNN + SD and DMDL methods. In the SD algorithm, the iteration number is set as 3000. In the MCNN + SD method, we first use the MCNN to obtain an approximate ILT solution, and then SD algorithm is run for 300 iterations to refine the mask pattern. Other simulation parameters are the same as those used in Figs. 8, 9 and 10. However, it shows that the DMDL network leads to higher PE than the SD and MCNN + SD methods. Thus, for the complex testing layout, it is hard to obtain the promising ILT solution from the DMDL network directly. In order to solve this problem, we add an auxiliary mask refine process based on SD algorithm to compose the DMDL + SD method. In particular, we use the DMDL network to produce an initial guess of the ILT solution, and then carry out the SD algorithm for 300 iterations to further optimize the mask pattern. The simulation results of DMDL + SD method is provided in the last column in Fig. 12.

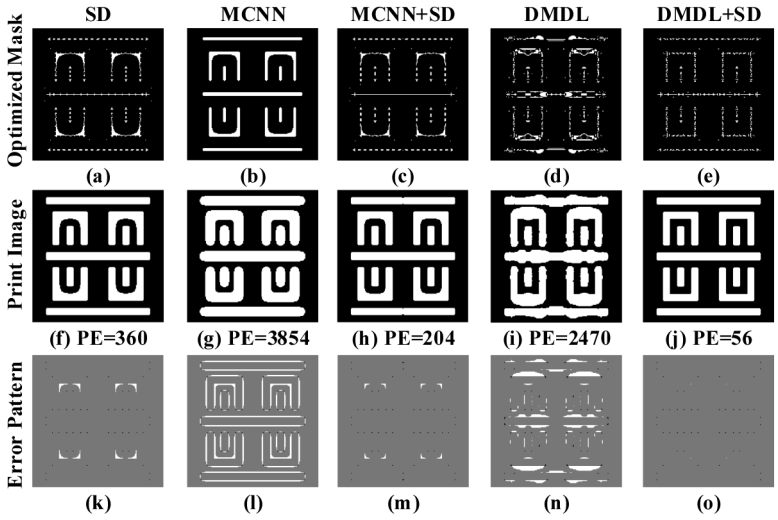

Fig. 12. Simulation results of different ILT methods using complex testing layout.

Optics EXPRESS

It shows that the DMDL method can automatically produce free-form SRAF pattern on the mask. In addition, the proposed DMDL + SD method can significantly improve the image fidelity compared with other ILT methods. Compared with the SD and MCNN + SD methods, the DMDL + SD method reduces the pattern error by 84% and 73%, respectively. It is very impressive that the DMDL is trained based on several simple layouts, but it still performs well on the complex testing layout. This provides an evidence for the generalization capability of the proposed DMDL network. The runtimes of different ILT methods using complex testing layout are provided in Table 2. Similarly, we use the total optimization time to compare the computational efficiency of different methods. The MCNN and DMDL methods without auxiliary refine process are very fast, but the resulting PEs are larger than the other three methods. The DMDL + SD improves the computational efficiency by 6.9 times compared to the SD method. On the other hand, the DMDL + SD method has comparable runtime to the MCNN + SD method, but the DMDL + SD method leads to much superior image fidelity over other ILT methods.

Table 2. Comparison of the runtimes based on complex testing layout.

Algorithm	SD	MCNN	MCNN + SD	DMDL	DMDL + SD
Training time	-	0.50 s	0.50 s	8.79 s	8.79 s
Testing time	-	0.55×10^{-2} s	0.55×10^{-2} s	15.23×10^{-2} s	15.23×10^{-2} s
Iteration time	463.07×10^{-2} s	-	52.02×10^{-2} s	-	51.81×10^{-2} s
Total optimization time	463.07×10^{-2} s	0.55×10^{-2} s	$52.\ 57\times10^{-2}$ s	15.23×10^{-2} s	67.04×10^{-2} s

It is known that the size of training data set has significant impact on the prediction capacity of deep network. In the previous simulations, the network parameters were optimized based on

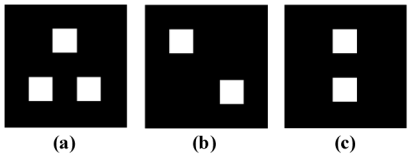

Fig. 13. The additional three training layouts.

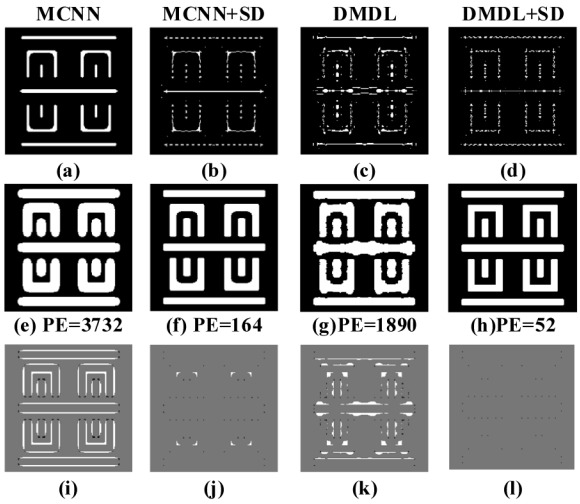

Fig. 14. The simulations of DMDL using different number of training layouts.

Optics EXPRESS

the nine training layouts in Fig. 5. Next, we increase the number of training layouts to study the impact of the training data size. In the following, we increase the number of training layouts to twelve by adding the three training layouts in Fig. 13. From left to right, Fig. 14 presents the simulation results of the MCNN, MCNN + SD, DMDL and DMDL + SD methods. All of the simulation parameters are the same as those used in previous simulations. According to the simulation results, increasing the number of training layouts can further improve the prediction capacity of the networks and thus reducing the pattern errors. However, it is observed that the predicted ILT results in Fig. 14 are not as good as the results in Fig. 8–Fig. 10. That is because the environments within the complex testing layout are different from the environments within the simple training layouts. Thus, the simple layouts cannot provide enough a-priori knowledge of the optical proximity effects in the complex layout. In the future, we will add more different layout patterns in the training data set to further improve the performance of DMDL network.

This paper provides a new concept for optimizing lithographic masks by model-driven deep neural networks. In order to apply the DMDL framework to actual semiconductor manufacturing, a large number of actual integrated circuit layouts should also be collected as training data. We will carry out more detailed research for this work in the future.

6. Conclusion

This paper developed a DMDL approach to effectively improve the computational efficiency of the ILT method, which can directly insert SRAFs on the mask pattern. Based on the inverse optimization model of the ILT problem, the dual-channel network architecture and the initial network parameters were derived systematically, where the data flow was separated into two paths to optimize the main feature pattern and SRAF pattern, respectively. Several advantages of the dual-channel network were proved, which alleviate the vanishing gradient problem, extend the depth of network, and automatically generate the SRAF pattern around the main feature. In order to avoid the time-consuming labelling process of training samples, an unsupervised training method was proposed based on the auto-decoding strategy using the lithography imaging model. The superiority of the proposed DMDL method over other ILT methods was shown in terms of both computational efficiency and imaging performance. In future work, some advanced strategies will be studied to further improve the prediction performance and computational efficiency of the DMDL method.

A. Appendix

According to Eq. (30), the gradient of cost function E' with respect to $\hat{\mathbf{M}}$ is

$$\nabla E'|_{\hat{\mathbf{M}}} = \nabla E|_{\hat{\mathbf{M}}} + \nabla R_Q|_{\hat{\mathbf{M}}}, \tag{35}$$

where $\nabla E|_{\hat{\mathbf{M}}}$ can be calculated as follows

$$\nabla E|_{\hat{\mathbf{M}}} = -4a_r \cdot \{[(\widetilde{\mathbf{Z}} - \mathbf{Z}') \odot \mathbf{Z}' \odot (\mathbf{1} - \mathbf{Z}') \odot (\hat{\mathbf{M}} \otimes \mathbf{h})] \otimes \mathbf{h}^{\circ}\}. \tag{36}$$

According to Eq. (29), the gradient of quadratic penalty is formulated as

$$\nabla R_Q|_{\hat{\mathbf{M}}} = -8\hat{\mathbf{M}} + 4. \tag{37}$$

According to Eq. (19), the gradient of $\mathbf{M}_b^{K+1}$ to $\mathbf{M}_g^K$ is

$$\nabla \mathbf{M}_b^{K+1}|_{\mathbf{M}_g^K} = a_m \odot \mathbf{M}_b^{K+1} \odot \mathbf{M}_b^{K+1}. \tag{38}$$

According to Eq. (13), the gradients of $\mathbf{M}_{mg}^k$ to $\mathbf{M}_{mb}^k$ and $\mathbf{M}_b^k$ are

$$\nabla \mathbf{M}_{mg}^k|_{\mathbf{M}_{mb}^k} = (\mathbf{M}_{mg}^k \otimes \mathbf{S}_m^{k\circ}), \tag{39}$$

Optics EXPRESS

$$\begin{aligned}\nabla \mathbf{M}_{mg}^{k}|_{\mathbf{M}_{b}^{k}} &= [\mathbf{T}^{k} \odot (\mathbf{M}_{b}^{k} \otimes \mathbf{D}_{m}^{k})] \otimes \mathbf{W}_{m}^{k} \odot (\mathbf{1} - 2\mathbf{M}_{b}^{k}) \odot \mathrm{Q} \odot \mathbf{M}_{mg}^{k} \\ &+ \mathbf{M}_{mg}^{k} \odot \mathbf{M}_{b}^{k} \odot (\mathbf{1} - \mathbf{M}_{b}^{k}) \odot \mathrm{Q} \otimes \mathbf{W}_{m}^{k\circ} \odot \mathbf{T}^{k} \otimes \mathbf{D}_{m}^{k\circ}\end{aligned}. \tag{40}$$

According to Eq. (14), the gradient of $\mathbf{M}_{sg}^{k}$ to $\mathbf{M}_{b}^{k}$ is

$$\begin{aligned}\nabla \mathbf{M}_{sg}^{k}|_{\mathbf{M}_{b}^{k}} &= [\mathbf{T}^{k} \odot (\mathbf{M}_{b}^{k} \otimes \mathbf{D}_{s}^{k})] \otimes \mathbf{W}_{s}^{k} \odot (\mathbf{1} - 2\mathbf{M}_{b}^{k}) \odot \mathrm{Q} \odot \mathbf{M}_{sg}^{k} \\ &+ \mathbf{M}_{sg}^{k} \odot \mathbf{M}_{b}^{k} \odot (\mathbf{1} - \mathbf{M}_{b}^{k}) \odot \mathrm{Q} \otimes \mathbf{W}_{s}^{k\circ} \odot \mathbf{T}^{k} \otimes \mathbf{D}_{s}^{k\circ}\end{aligned}. \tag{41}$$

According to Eqs. (15), (17) and (19), the gradients of $\mathbf{M}_{mb}^{k}$ and $\mathbf{M}_{b}^{k}$ to $\mathbf{M}_{mg}^{k-1}$ are

$$\nabla \mathbf{M}_{mb}^{k}|_{\mathbf{M}_{mg}^{k-1}} = a_m \odot \mathbf{M}_{mb}^{k} \odot \mathbf{M}_{mb}^{k}, \quad \nabla \mathbf{M}_{b}^{k}|_{\mathbf{M}_{mg}^{k-1}} = a_m \odot \mathbf{M}_{b}^{k} \odot \mathbf{M}_{b}^{k}. \tag{42}$$

According to Eqs. (39)–(42), the gradient of $\mathbf{M}_{g}^{k}$ to $\mathbf{M}_{mg}^{k-1}$ is

$$\nabla \mathbf{M}_{g}^{k}|_{\mathbf{M}_{mg}^{k-1}} = \nabla \mathbf{M}_{mg}^{k}|_{\mathbf{M}_{mk}^{b}} \cdot \nabla \mathbf{M}_{mb}^{k}|_{\mathbf{M}_{mg}^{k-1}} + \nabla \mathbf{M}_{mg}^{k}|_{\mathbf{M}_{b}^{k}} \cdot \nabla \mathbf{M}_{b}^{k}|_{\mathbf{M}_{mg}^{k-1}} + \nabla \mathbf{M}_{sg}^{k}|_{\mathbf{M}_{b}^{k}} \cdot \nabla \mathbf{M}_{b}^{k}|_{\mathbf{M}_{mg}^{k-1}}. \tag{43}$$

The gradient of $\mathbf{M}_{mg}^{k}$ to $\mathbf{S}_{m}^{k}$ is

$$\nabla \mathbf{M}_{mg}^{k}|_{\mathbf{S}_{m}^{k}} = \Pi\{\mathbf{M}_{mb}^{k} \otimes \mathbf{M}_{mg}^{k}\}, \tag{44}$$

where $\Pi\{\cdot\}$ is the window function to truncate the $N_f \times N_f$ central part of the matrix in the argument. The dimension of the gradient matrix of $\mathbf{M}_{mg}^{k}$ to $\mathbf{S}_{m}^{k}$ is greater than the matrix dimension of $\mathbf{S}_{m}^{k}$. Therefore, we use the window function $\Pi\{\cdot\}$ to keep the matrix dimension of $\mathbf{S}_{m}^{k}$ unchanged during the training process. The gradient of $\mathbf{M}_{mg}^{k}$ to $\mathbf{W}_{m}^{k}$ is

$$\nabla \mathbf{M}_{mg}^{k}|_{\mathbf{W}_{m}^{k}} = \Pi\{[\mathbf{T}^{k} \odot (\mathbf{M}_{b}^{k} \otimes \mathbf{D}_{m}^{k})] \otimes [\mathbf{M}_{mg}^{k} \odot \mathbf{M}_{b}^{k} \odot (\mathbf{1} - \mathbf{M}_{b}^{k}) \odot Q]\}. \tag{45}$$

The gradient of $\mathbf{M}_{mg}^{k}$ to $\mathbf{D}_{m}^{k}$ is

$$\nabla \mathbf{M}_{mg}^{k}|_{\mathbf{D}_{m}^{k}} = \Pi\{\mathbf{M}_{b}^{k} \otimes [\mathbf{T}^{k} \odot (\mathbf{M}_{mg}^{k} \odot \mathbf{M}_{b}^{k} \odot (\mathbf{1} - \mathbf{M}_{b}^{k}) \odot Q \otimes \mathbf{W}_{m}^{k\circ})]\}. \tag{46}$$

According to Eqs. (31) and (35)–(46), we can calculate the derivatives of cost function with respect to the elements of $\mathbf{S}_{m}^{k}, \mathbf{D}_{m}^{k}$ and $\mathbf{W}_{m}^{k}$, respectively. Similarly, we can also calculate the derivatives of cost function with respect to the elements of $\mathbf{S}_{s}^{k}, \mathbf{D}_{s}^{k}$ and $\mathbf{W}_{s}^{k}$, respectively.

Funding

National Natural Science Foundation of China (61675021); Fundamental Research Funds for the Central Universities (2018CX01025).

Disclosures

The authors declare no conflicts of interest.

References

1. A. K. Wong, *Resolution Enhancement Techniques in Optical Lithography* (SPIE, 2001).
2. F. Schellenberg, "A little light magic," IEEE Spectrum **40**(9), 34–39 (2003).
3. X. Ma and G. R. Arce, *Computational Lithography* (John Wiley and Sons, 2010).
4. G. E. Moore, "Cramming more components onto integrated circuits," Proc. IEEE **86**(1), 82–85 (1998).
5. N. B. Cobb, "Fast optical and process proximity correction algorithms for integrated circuit manufacturing," PhD thesis (University of California, Berkeley, 1998).

Optics EXPRESS

6. A. Poonawala and P. Milanfar, "Mask design for optical microlithography—an inverse imaging problem," IEEE Trans. on Image Process **16**(3), 774–788 (2007).
7. Y. Granik, "Fast pixel-based mask optimization for inverse lithography," J. Micro/Nanolith. MEMS MOEMS **5**(4), 043002 (2006).
8. J. Yu and P. Yu, "Impacts of cost functions on inverse lithography patterning," Opt. Express **18**(22), 23331–23342 (2010).
9. X. Ma, Z. Wang, J. Zhu, S. Zhang, G. R. Arce, and S. Zhao, "Nonlinear compressive inverse lithography aided by low-rank regularization," Opt. Express **27**(21), 29992–30008 (2019).
10. A. Tritchkov, S. Kobelkov, S. Rodin, K. Sakajiri, E. Egorov, and S. Woo, "Use of ILT-based mask optimization for local printability enhancement," Proc. SPIE **9256**, 92560X (2014).
11. M. Julien, K. Jeroen, D. Peter, and D. Kristin, "Metal1 patterning study for random logic applications with 193i, using calibrated OPC for litho and etch," Proc. SPIE **9052**, 90520Q (2014).
12. Y. Shen, N. Wong, and E. Y. Lam, "Level-set-based inverse lithography for photomask synthesis," Opt. Express **17**(26), 23690–23701 (2009).
13. Y. Shen, N. Jia, N. Wong, and E. Y. Lam, "Robust level-set-based inverse lithography," Opt. Express **19**(6), 5511–5521 (2011).
14. S. Shen, P. Yu, and D. Z. Pan, "Enhanced DCT2-based inverse mask synthesis with initial SRAF insertion," Proc. SPIE **7122**, 712241 (2008).
15. X. Ma and G. R. Arce, "Pixel-based OPC optimization based on conjugate gradients," Opt. Express **19**(3), 2165–2180 (2011).
16. X. Ma, Y. Li, and L. Dong, "Mask optimization approaches in optical lithography based on a vector imaging model," J. Opt. Soc. Am. A **29**(7), 1300–1312 (2012).
17. X. Wu, S. Liu, W. Lv, and E. Y. Lam, "Robust and efficient inverse mask synthesis with basis function representation," J. Opt. Soc. Am. A **31**(12), B1–B9 (2014).
18. W. Lv, Q. Xia, and S. Y. Liu, "Mask-filtering-based inverse lithography," J. Micro/Nanolith. MEMS MOEMS **12**(4), 043003 (2013).
19. R. Luo, "Optical proximity correction using a multilayer perceptron neural network," J. Opt. **15**(7), 075708 (2013).
20. S. Choi, S. Shim, and Y. Shin, "Machine learning (ML)-guided OPC using basis functions of polar Fourier transform," Proc. SPIE **9780**, 97800H (2016).
21. K. Luo, Z. Shi, X. Yan, and Z. Geng, "SVM based layout retargeting for fast and regularized inverse lithography," J. Zhejiang Uni. - Sci. C **15**(5), 390–400 (2014).
22. X. Ma, S. Jiang, J. Wang, B. Wu, Z. Song, and Y. Li, "A fast and manufacture-friendly optical proximity correction based on machine learning," Microelectron. Eng **168**, 15–26 (2016).
23. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science **313**(5786), 504–507 (2006).
24. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**(7553), 436–444 (2015).
25. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (The MIT Press, 2016).
26. W. Sim, K. Lee, D. Yang, J. Jeong, J. Hong, S. Lee, and H. Lee, "Automatic correction of lithography hotspots with a deep generative model," Proc. SPIE **10961**, 1096105 (2019).
27. D. S. Woldeamanual, A. Erdmann, and A. Maier, "Application of deep learning algorithms for lithographic mask characterization," Proc. SPIE **10694**, 1069408 (2018).
28. H. Yang, S. Li, Y. Ma, B. Yu, and E. Young, "GAN-OPC: mask optimization with lithography-guided generative adversarial nets," in *Proceedings of IEEE Conference on Design Automation Conference* (IEEE, 2018), pp.1–6.
29. Y. Zhang and W. Ye, "Deep learning based inverse method for layout design," https://arxiv.org/abs/1806.03182.
30. X. Ma, Q. Zhao, H. Zhang, Z. Wang, and G. R. Arce, "Model-driven convolution neural network for inverse lithography," Opt. Express **26**(25), 32565–32584 (2018).
31. X. Su, P. Gao, Y. Wei, and W. Shi, "SRAF rule extraction and insertion based on inverse lithography technology," Proc. SPIE **10961**, 109610P (2019).
32. Y. Hu, A. Huber, J. Anumula, and S. Liu, "Overcoming the vanishing gradient problem in plain recurrent networks," https://arxiv.org/pdf/1801.06105.
33. M.D. Levenson, N.S. Viswanathan, and R.A. Simpson, "Improving resolution in photolithography with a phase-shifting mask," IEEE Trans. Electron Devices **29**(12), 1828–1836 (1982).
34. X. Ma, G. R. Arce, and Y. Li, "Optimal 3D phase-shifting masks in partially coherent illumination," Appl. Opt. **50**(28), 5567–5576 (2011).
35. H. H. Hopkins, "On the diffraction theory of optical images," Proc. R. Soc. Lond. A **217**(1130), 408–432 (1953).
36. X. Ma, C. Han, Y. Li, L. Dong, and G. R. Arce, "Pixelated source and mask optimization for immersion lithography," J. Opt. Soc. Am. A **30**(1), 112–123 (2013).
37. D. Li and D. Yu, *Deep Learning: Methods and Applications* (Now Publishers, 2014).