A Fast and Accurate PEB Simulation Through Recurrent Neural Network

Gangmin Cho, Taeyoung Kim, Seohyun Kim, and Youngsoo Shin
School of Electrical Engineering, KAIST, Daejeon 34141, Korea

ABSTRACT

Post-exposure bake (PEB) consists of neutralization, diffusion, and catalysis steps, and are modeled by partial differential equations (PDEs). Commercial PEB simulation relies on numerical methods to explicitly solve PDEs in both spatial and temporal domains, and is very time consuming. A machine learning model has been applied to quickly predict the final inhibitor distribution with initial acid distribution as a model input. The accuracy, however, is not good enough; for different PEB condition comprising baking time and temperature, the model should be trained again, which is another limitation.

A recurrent neural network (RNN) is proposed for fast PEB simulation. The network is constructed around convolutional long short-term memory (convLSTM), which is a popular RNN for spatio-temporal prediction. Key inputs of convLSTM include the encoded values of acid and quencher distributions as well as their multiplication; acid and quencher distributions on next time step are obtained after the outputs of convLSTM pass through decoders. Once acid distribution is derived at time instance of interest, inhibitor distribution is extracted directly from its PDE. To accelerate RNN prediction, operations are skipped and the distribution at the next time step is simply copied from the one at the current time step if PEB reaction does not occur. Experiments have shown that the runtime of PEB simulation is reduced by 88.1% with smaller total PDE loss by 35.3%, compared to commercial tool.

Keyword: Post-exposure bake, partial differential equation, recurrent neural network, convolutional long short-term memory

1. INTRODUCTION

Rigorous lithography simulation has been used for mask verification. It is based on lithography model, which consists of optical model and resist model. Optical model describes exposure process. Using sum of coherent systems approximation, optical model generates intensity map, or aerial image, from mask pattern. PEB and development are then modeled by resist model. PEB simulation takes an aerial image and yields the inhibitor profile, which is provided to development simulation for final photoresist profile.

In high-resolution lithography using deep ultraviolet (e.g. KrF, ArF), chemically amplified resist (CAR) is used for photoresist. For CAR-based lithography, PEB is a mandatory process to amplify chemical reaction. The acid generated during exposure catalyzes the removal of the inhibitor from the baked photoresist, leaving the region of inhibitor insoluble during the development process. This catalysis is commonly expressed by:

$$\frac{\partial[I]}{\partial t} = -k_c[I][A], \tag{1}$$

DTCO and Computational Patterning III, edited by Neal V. Lafferty, Proc. of SPIE
Vol. 12954, 129540G · © 2024 SPIE · 0277-786X · doi: 10.1117/12.3010292

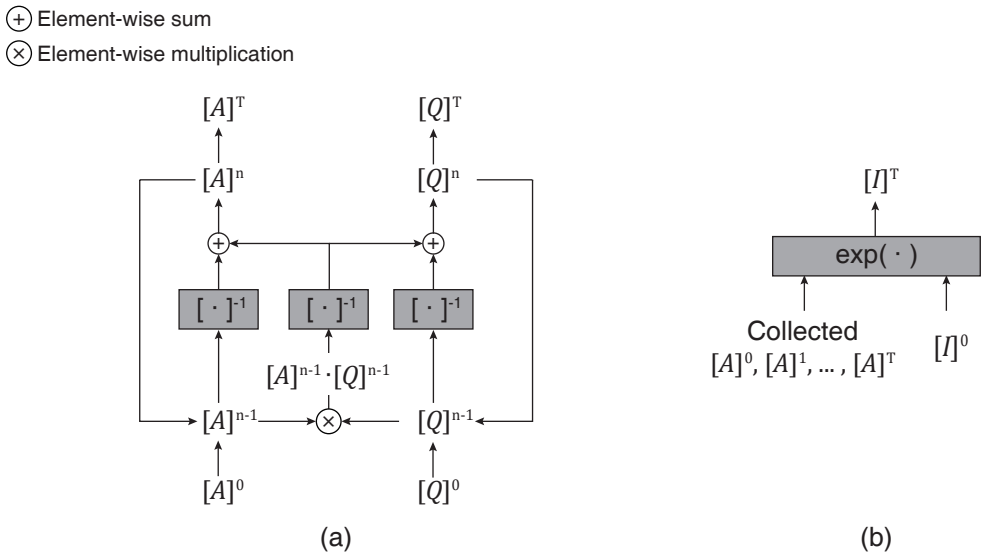

Figure 1. Numerical methods for PEB simulation: (a) neutralization-diffusion process and (b) catalysis process.

where $[I]$ and $[A]$, functions of x, y, and z as well as time t, correspond to the distributions of inhibitor and acid, respectively; k_c is the catalysis coefficient. Inhibitor and quencher are uniformly distributed in initial photoresist. Acid and quencher diffuse and meet each other, neutralizing upon encounter and disappearing. To describe this neutralization-diffusion processes, resist model for CAR is typically given by PDEs:

$$\frac{\partial [A]}{\partial t} = -k_n[A][Q] + \nabla(D_A \nabla[A]), \tag{2}$$

$$\frac{\partial [Q]}{\partial t} = -k_n[A][Q] + \nabla(D_Q \nabla[Q]), \tag{3}$$

where $[Q]$ is the distribution of quencher, k_n is the neutralization coefficient, and D_A, D_Q are the diffusion coefficients of acid and quencher, respectively.[1]

PDEs may be solved using the finite difference method (FDM) in spatial and time domains. Discretizations of $[A]$, $[Q]$, and $[I]$ lead to the profiles of acid $[A]^n$, quencher $[Q]^n$, and inhibitor $[I]^n$, respectively at time step n. Discretized PDEs of Equation (2) and Equation (3) are solved to obtain $[A]^n$ (or $[Q]^n$). Complex matrix inversions of $[A]^{n-1}$, $[Q]^{n-1}$, and their element-wise multiplication are computed as illustrated in Figure 1(a). After iteratively computing $[A]^n$ and $[Q]^n$ starting from $[A]^0$ and $[Q]^0$, the profiles at the final time step $[A]^T$ and $[Q]^T$ are obtained. Once the acid profile is derived at all time steps, the final inhibitor profile $[I]^T$ is directly extracted from its discretized PDE of Equation (1) using simple exponential function[2] as shown in Figure 1(b).

For fast lithography simulation on optical and resist models, machine learning-based approaches have been studied. A neural PDE solver for PEB simulation, called DeePEB, is introduced.[3] It employs kernel approximation through fast Fourier transform (FFT) to predict $[I]^T$ given $[A]^0$ as an input. The lost feature during FFT is calibrated by using convolutional neural network. Although this approach is faster than standard numerical methods, it exhibits limited accuracy. Moreover, the machine learning

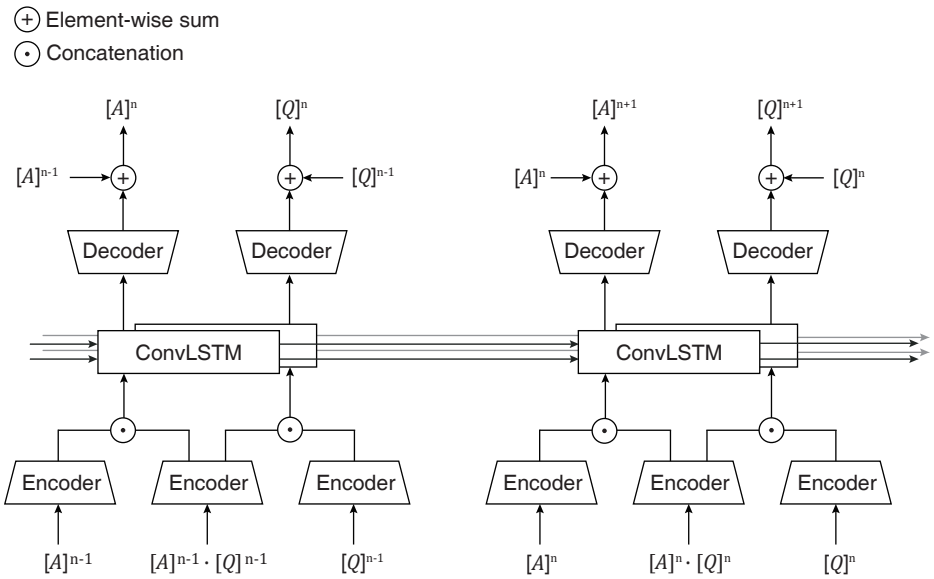

Figure 2. Recurrent neural network: overview.

model is not adaptable for different PEB conditions comprising baking time and temperature, the model should be re-trained. To avoid re-training, unsupervised learning algorithm using PDE-based loss function should be applied.

1.1 Motivation and Contributions

Recent studies have shown that RNN can be used to solve PDEs both in spatial and time domain, e.g., the Navier-Stokes equations are solved through LSTM with encoders and decoders for weather prediction.[4] In this paper, PEB simulation is performed through RNN, which is customized for efficient computation.

The proposed RNN architecture consists of encoder, decoder, and convLSTM using three profile inputs. Compared to only usage of two profile inputs ($[A]^{n-1}$ and $[Q]^{n-1}$), element-wise multiplication of two profiles becomes an additional input to derive the two profiles at next time step ($[A]^n$ and $[Q]^n$). While encoded multi-channel values ($x = 1$, $y = 1$, $z = 1$) are provided to LSTM, convLSTM should be applied for multi-channel images ($x = a$, $y = b$, $z = 1$; clip ratio $= a/b$, $a >> 1$, $b >> 1$) since the height of photoresist (z-axis) is much smaller than the size of mask clip (x and y-axis).

We also propose operation skipping method to reduce simulation runtime while maintaining the accuracy. When $t = 0$, reaction usually occurs at the surface of photoresist or at the center regions of mask pattern.[5] Therefore, skipping is applied for a particular cube when neutralization does not occur especially for bottom part of photoresist or far from the mask patterns. When $t = T$, diffusion becomes saturated and therefore skipping is also used.

The remainder of this paper is organized as follows. The proposed RNN for PEB simulation is presented in Section 2. Experimental results are given in Section 3, and the paper is concluded in Section 4.

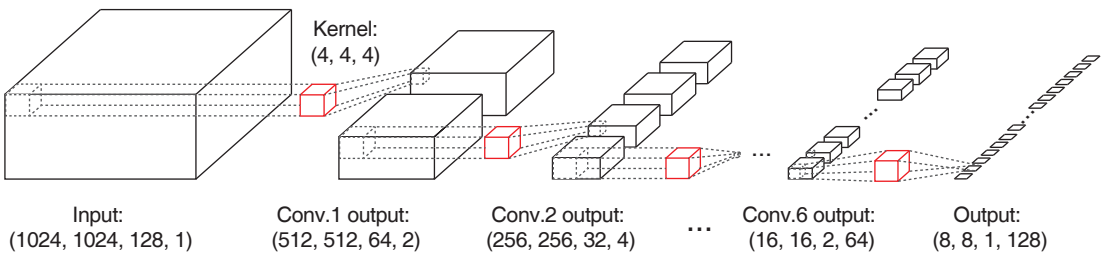

Figure 3. Encoder architecture.

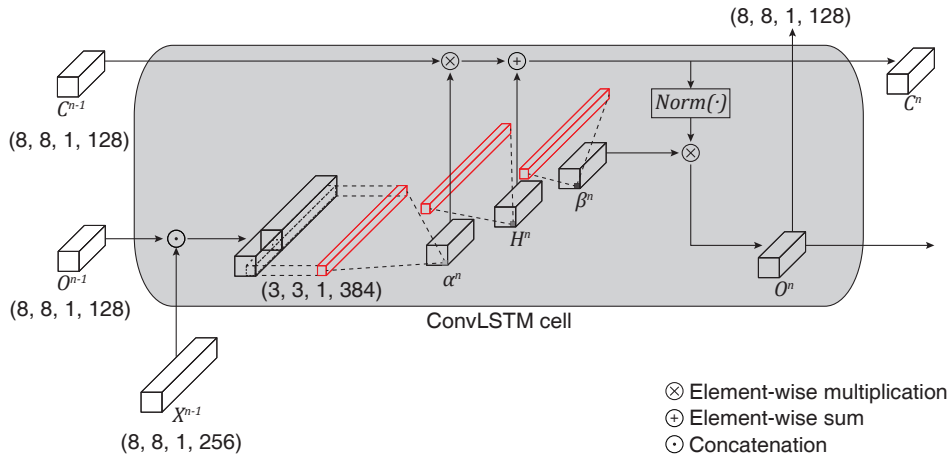

Figure 4. Architecture of convLSTM.

2. PEB SIMULATION THROUGH RNN

2.1 RNN Architecture

The proposed RNN model to solve PDEs is shown in Figure 2. A couple of convLSTM is used on each time step, with three distinct encoders and two distinct decoders. Three inputs are used: $[A]^{n-1}$, $[Q]^{n-1}$, and their element-wise multiplication. After each profile is encoded, encoded $[A]^{n-1}$ and encoded $[Q]^{n-1}$ are respectively concatenated with their element-wise multiplication. Two concatenated results are provided to one of convolutional long short-term memory (convLSTM), and the output of the model is decoded. The output of decoder refers to $\Delta[A]$ (or $\Delta[Q]$), and the final output $[A]^n$ is obtained by element-wise sum of $\Delta[A]$ and $[A]^{n-1}$. The kernels of two convLSTMs are shared to predict $[A]$ and $[Q]$ for each time step.

The architecture of encoder is shown in Figure 3. The encoder input is a profile with single-channel, represented by (x, y, z, 1). Seven convolutional layers comprise the encoder. Each layer is associated with multi-channel kernels (4, 4, 4) with stride of (2, 2, 2). The output from each convolutional layer gradually decreases by half while the number of channels is doubled. Convolution stops when z is equal to 1, so the output is a 2D image with multi-channels represented by (x', y', 1, c).

ConvLSTM architecture is illustrated in Figure 4. After the outputs of two encoders (e.g. $[A]^{n-1}$ and element-wise multiplication of $[A]^{n-1}$ and $[Q]^{n-1}$) are concatenated, the concatenated output $[X]^{n-1}$ is

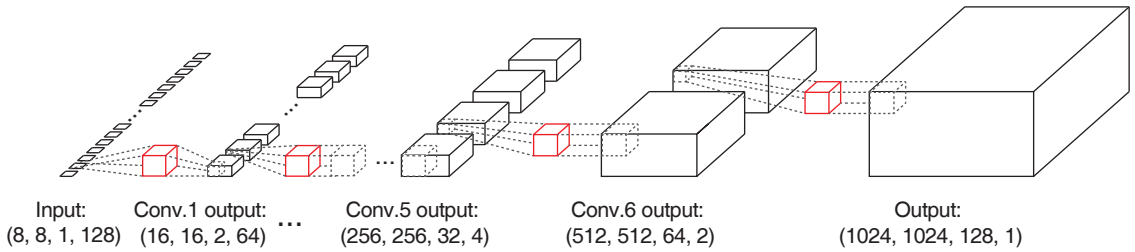

Figure 5. Decoder architecture.

provided to convLSTM. ConvLSTM receives long-term data and short-term data from the previous convLSTM. Short-term data O^{n-1} is concatenated with X^{n-1} and convolution is performed by using three kernels to obtain two scaling maps α, β, and a feature map H. Long-term data C^{n-1} is multiplied by α^n, and add H^n to generate next long-term data C^n. C^n is multiplied by β^n to obtain next short-term data O^n. Two outputs are provided to the next convLSTM, and only O^n is used for the decoder.

The architecture of decoder is illustrated in Figure 5. Decoder input is a 2D image with single-channel, represented by (x', y', 1, c). Similar to encoder, seven deconvolutional layers are used, and same sizes of multi-channel kernels and stride are used. The output from each convolutional layer is doubled this time but the number of channels is decreased by half. Deconvolution stops when c is equal to 1, and therefore the output is a profile with single-channel represented by (x, y, z, 1).

Element-wise sum of decoder output and original profile $[\mathrm{A}]^{n-1}$ (or $[\mathrm{Q}]^{n-1}$) is performed to obtain the next profile $[\mathrm{A}]^n$ (or $[\mathrm{Q}]^n$).

2.2 Operation Skipping

A set of cubes within the profile $[A]$ (or $[Q]$) is not used as RNN input when concentration of a certain cube and its surroundings are similar to that of previous time step. At time step n, the flow of RNN operation skipping is illustrated in Figure 6. For each cube, similarity between $[A]^{n-1}$ and $[A]^n$ is compared. Similarity is calculated based on concentration change rate for cube ($x = i$, $y = j$, $z = k$) and those located within the range which are affected by diffusion from such cube:

$$\mathrm{Similarity}([A]^{n-1} \cdot L,\ [A]^n \cdot L) = 1 - \frac{1}{|L|} \sum_{i,j,k \in L} \frac{[A]^{n-1}_{i,j,k} - [A]^n_{i,j,k}}{[A]^n_{i,j,k}}, \tag{4}$$

where L denotes the list of surrounding cubes and $|L|$ denotes the size of list L. If similarity of a cube is larger than the threshold, the concentration of cube itself is used in $[A]^{n+1}$ at the same location without using RNN. Otherwise, concentrations for cubes changes through RNN, and they are positioned at the same location in $[A]^{n+1}$.

Given: $D_A = (D_x, D_y, D_z)$ and $(L_{skip}, L_{surround}) \leftarrow (\varnothing, \varnothing)$
Input: $[A]^{n-1}$ and $[A]^n$
Output: $[A]^{n+1}$
L1: **for** $(i, j, k) = (0, 0, 0)$ to (X, Y, Z)
L2: insert $(i \pm D_x\Delta t,\ j \pm D_y\Delta t,\ k \pm D_z\Delta t)$ to $L_{surround}$
L3: **if** Similarity$([A]^{n-1} \cdot L_{surround},\ [A]^n \cdot L_{surround}) > threshold$
L4: **then** insert (i, j, k) to L_{skip}
L5: $[A]^{n+1} \leftarrow [A]^n \cdot L_{skip}$
L6: $[A]^{n+1} \leftarrow \text{RNN}([A]^n \cdot {L_{skip}}^c)$

Figure 6. Algorithm of prediction using RNN with operation skipping at time step n.

2.3 Loss Function and Training

A loss function of PDEs is a key for RNN model training. It is given by transposing the right-hand side of Equation (2) and Equation (3) to the left:

$$\mathcal{L}_{1,n} = \frac{[A]^{n+1} - [A]^n}{\Delta t} + k_n[A]^n \cdot [Q]^n - \delta_A * [A]^n - \delta_A * [A]^{n+1}, \tag{5}$$

$$\mathcal{L}_{2,n} = \frac{[Q]^{n+1} - [Q]^n}{\Delta t} + k_n[A]^n \cdot [Q]^n - \delta_Q * [Q]^n - \delta_Q * [Q]^{n+1}, \tag{6}$$

$$\delta_A = \frac{D_A}{2\Delta x^2}\delta, \quad \delta_Q = \frac{D_Q}{2\Delta x^2}\delta, \tag{7}$$

where δ_A and δ_Q are the kernels that are proportional to D_A and D_B, respectively with the coefficient $\delta = [[[0, 0, 0], [0, 1, 0], [0, 0, 0]], [[0, 1, 0], [1, -6, 1], [0, 1, 0]], [[0, 0, 0], [0, 1, 0], [0, 0, 0]]]$.

L2 norm of each loss function is used to calculate the PDE loss at each time step. Total PDE loss is given by root mean square (RMS) of PDE loss over all time steps:

$$\mathcal{L}_n = \sqrt{\mathcal{L}_{1,n}^2 + \mathcal{L}_{2,n}^2}, \tag{8}$$

$$\mathcal{L}_{total} = \sqrt{\sum_{n=0}^{N} \mathcal{L}_n^2}. \tag{9}$$

The kernels of encoder, convLSTM, and decoder are updated in training process, while total PDE loss is minimized.

3. EXPERIMENTAL RESULTS

Experiments are conducted with sample contact layouts from 28nm technology. MB-OPC (model-based OPC) and SRAF (sub-resolution assist feature) insertion are applied to the layouts, and 1000 clips (with each 2um × 2um clip centered at a contact) are extracted for RNN training and testing. Linux machine with AMD Ryzen Threadripper 3990X CPU and Nvidia RTX 3090 GPU is used for experiments. RNN

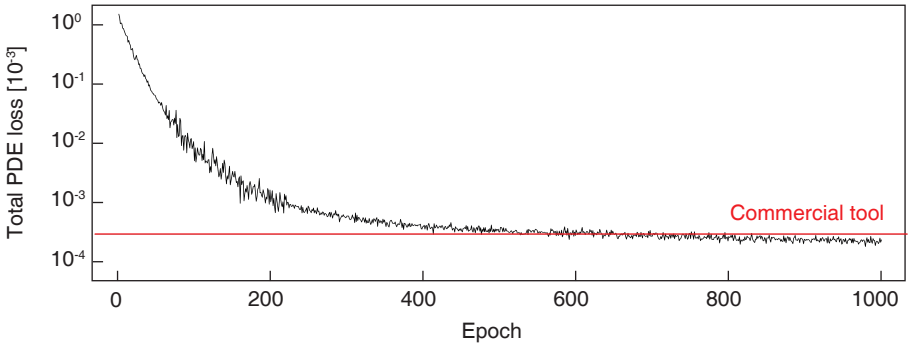

Figure 7. Total PDE loss compared to commercial tool.

model is constructed by using Pytorch library; the encoder consists of seven convolutional layers and the decoder is made of seven deconvolutional layers.

The height of photo-resist is set to 512nm. The size of a cube is (2nm, 2nm, 2nm), so the number of cubes for $[A]$ and $[Q]$ is (1024, 1024, 256). PEB simulation is configured with time interval of 10^{-7} seconds and total baking time of 100 seconds.

3.1 Comparison with Commercial Tool

We compare the RNN model with commercial tool.[6] As shown in Figure 7, when the number of epochs is 200, total PDE loss of our method is more than twice compared with that of commercial tool. However, once the number of epochs reaches to 591, proposed RNN achieves less total PDE loss than commercial tool. Saturation of total PDE loss occurs after epoch is 923, and RNN model consistently outperforms the commercial tool.

3.2 Assessment of Operation Skipping

We assess the operation skipping method in terms of runtime and total PDE loss. As shown in Figure 8(a), operation skipping allows 71.5% reduction in runtime (260 minutes to 74 minutes) when it is applied to the proposed method. Compared to commercial tool, the total runtime is reduced by 88.1%. Total PDE loss is compared in Figure 8(b). Skipping does not affect total PDE loss that much (2.1 vs 2.2); the proposed method with skipping achieves 35.3% reduction in total PDE loss compared to commercial tool.

4. CONCLUSION

PEB is a crucial step in optical lithography. A commercial PEB simulation explicitly solves the PDEs through FDM, which is time consuming. We have proposed RNN through unsupervised learning algorithm with loss function obtained by PDEs. Operation skipping has been applied to accelerate the PEB simulation. Experiments have shown that runtime is reduced by 88.1% and total PDE loss is

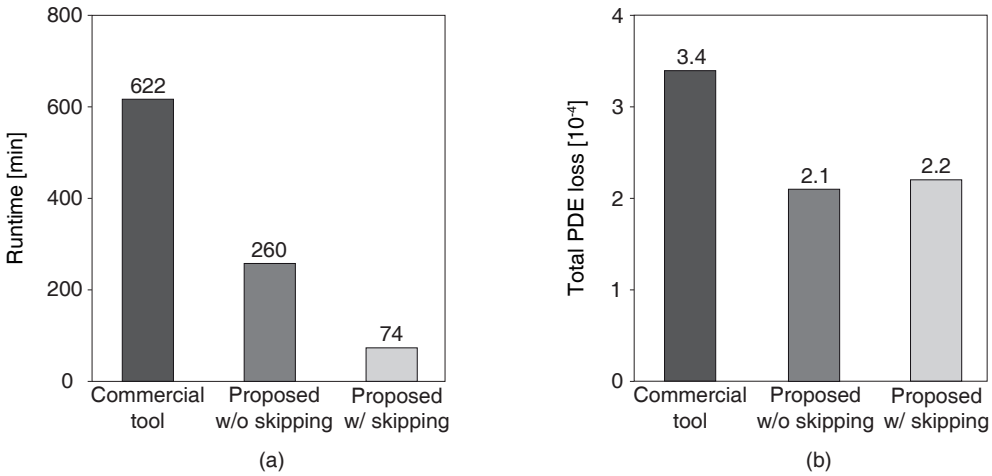

Figure 8. Assessment of operation skipping: (a) runtime and (b) total PDE loss.

decreased by 35.3%, when our proposed method with operation skipping is compared to commercial tool.

Acknowledgement

This work was partly supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-00754, Software Systems for AI Semiconductor Design), National Research Foundation of Korea (NRF) of MSIT (No.2019R1A2C2003402), and BK21 FOUR (Fostering Outstanding Universities for Research) funded by National Research Foundation of Korea (NRF).

REFERENCES

1. D. Matiut *et al.*, "New models for the simulation of post-exposure bake of chemically amplified resists," in *Proc. SPIE Advances in Resist Technology and Processing XX*, Jun. 2003, pp. 1132–1142.
2. E. H. Croffie *et al.*, "Survey of chemically amplified resist models and simulator algorithms," in *Proc. SPIE Advances in Resist Technology and Processing XVIII*, Aug. 2001, pp. 983–991.
3. Q. Wang *et al.*, "DeePEB: A neural partial differential equation solver for post exposure baking simulation in lithography," in *Proc. International Conference on Computer-Aided Design*, Oct. 2022, pp. 1–9.
4. R. Zhang, Y. Liu, and H. Sun, "Physics-informed multi-LSTM networks for metamodeling of nonlinear structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 369, pp. 113 226–1–113 226–16, Jun. 2020.
5. L. Singh, "Effect of nanoscale confinement on the physical properties of polymer thin films," Ph.D. dissertation, Georgia Institute of Technology, Oct. 2004.
6. *Sentaurus Lithography*, Synopsys, Mountain View, CA, USA, Jun. 2018.