DeePEB: A Neural Partial Differential Equation Solver for Post Exposure Baking Simulation in Lithography

Qipan Wang[1], Xiaohan Gao[1,2], Yibo Lin[1,3*], Runsheng Wang[1,3], Ru Huang[1,3]
[1]School of Integrated Circuits, Peking University [2]School of Computer Science, Peking University
[3]Beijing Advanced Innovation Center for Integrated Circuits, Beijing, China
{qpwang,xiaogangao,yibolin,r.wang,ruhuang}@pku.edu.cn

ABSTRACT

Post Exposure Baking (PEB) has been widely utilized in advanced lithography. PEB simulation is critical in the lithography simulation flow, as it bridges the optical simulation result and the final developed profile in the photoresist. The process of PEB can be described by coupled partial differential equations (PDE) and corresponding boundary and initial conditions. Recent years have witnessed growing presence of machine learning algorithms in lithography simulation, while PEB simulation is often ignored or treated with compact models, considering the huge cost of solving PDEs exactly. In this work, based on the observation of the physical essence of PEB, we propose DeePEB: a neural PDE Solver for PEB simulation. This model is capable of predicting the PEB latent image with high accuracy and >100 × acceleration (compared to the commercial rigorous simulation tool), paving the way for efficient and accurate photoresist modeling in lithography simulation and layout optimization.

1 INTRODUCTION

Lithography simulation is one of the key steps in the integrated circuit (IC) design flow, verifying whether design patterns can be correctly printed in the photoresist. With continuous scaling of semiconductor feature size, lithography models become more and more complicated, increasing the complexity of lithography simulation and slowing down the closure of design verification [1].

Lithography simulation composes of two parts: optical simulation and photoresist simulation. Optical simulation describes the behavior of the light illuminated from the source, including passing through the mask, focusing and imaging in the photoresist, and generating the aerial image. Photoresist simulation models consecutive process steps, including post-exposure baking (PEB) and development. Fig. 1 sketches the entire lithography simulation flow for chemical amplified photoresist (CAR), which receives great popularity in modern high-resolution lithography. After exposure simulation, the latent image in the photoresist (distribution of the photoacid, product of the light-induced reaction) can be computed according to the aerial image. Then PEB activates the photoacid-catalyzed deprotection of the development inhibitor, converting it from insoluble to soluble. PEB simulation calculates the PEB latent image of unreacted inhibitor distribution from the latent image. The following development simulation simulates the etching process of the inhibitor and produces the final photoresist profile. To this end, metrology and inspection can be conducted to check the quality of the profile. During the entire flow, PEB simulation serves as a bridge between the optical simulation and the development calculation. According to our studies on the commercial rigorous simulation tool, Synopsys Sentaurus Lithography (S-Litho) [2], PEB simulation takes about 10 ∼ 30% of the runtime in the whole rigorous lithography simulation flow, as shown in Fig. 2.

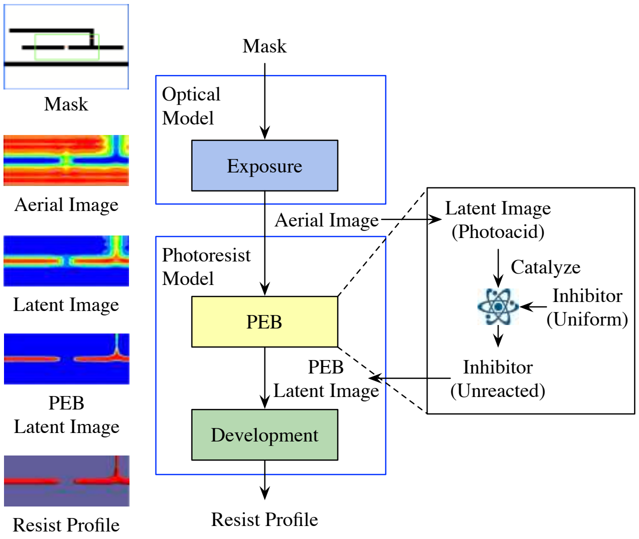


Figure 1: Middle column: a representative lithography simulation flow for CAR. Left column: example of the calculation results. Right column: the core catalytic reaction happening in PEB.

Recent advances in machine learning (ML) bring new opportunities for optical simulation [3–6], hotspot detection [7, 8], and photoresist modeling [9, 10]. For example, Ye et al. [4] propose TEMPO, capable of predicting 3d aerial image efficiently and accurately, and succeed to reduce the time consumed in exposure simulation greatly, as shown in Fig. 2. As a consequence, the PEB simulation now takes the longest time in the flow. However, the PEB process happening in the photoresist is typically ignored or treated with compact models in the previous work, bringing about a non-negligible error. Additionally, most studies regard lithography simulation as a black box modeling problem without considering

[*]Corresponding author

ICCAD '22, October 30-November 3, 2022, San Diego, CA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9217-4/22/10…$15.00
https://doi.org/10.1145/3508352.3549398

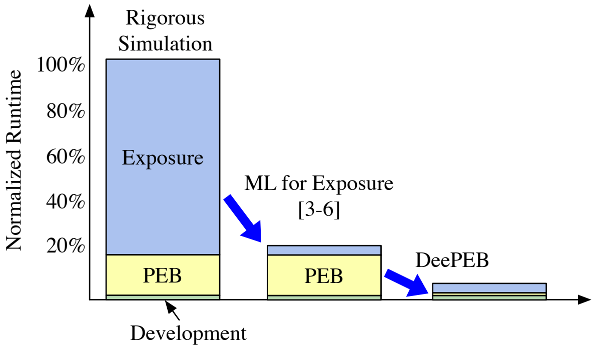


Figure 2: Typical runtime distribution ratio of each part in lithography simulation flow for different simulation schemes.

its fundamental physical interpretation. Thus, accurately and efficiently simulating the PEB process based on its physical essence is critical to the closure of lithography simulation.

To describe the complex transitions between multiple chemical states involved in the PEB process, various models have been proposed [11–14] to trace and approximate the physical and chemical reactions. These models typically solve partial differential equations (PDE) with numerical methods to calculate the PEB latent image. Traditional numerical PEB solvers discretize the governing equations in the space-time domain and derive the solutions by repeatedly evaluating the time advancement equations. Mesh-dependent algorithms such as finite element methods (FEM) and finite difference methods (FDM) [15] are usually exploited, imposing a trade-off between resolution and runtime. Fine discretization in terms of time and space is usually required to achieve convergence and high accuracy, at the cost of runtime.

To model the crucial PEB process in lithography simulation, in this work, we propose DeePEB, a neural PDE solver for efficient and accurate PEB simulation. PEB modeling of the positive-tone CAR is discussed, while our algorithm can be extended for other types of photoresist. We approximate the elementary physical dynamics in PEB with neural networks (NN). For one thing, we model the dynamic behaviors in the Fourier domain with a neural operator. For another, we capture the local high-frequency features with customized convolution operations. The major contributions of this paper are highlighted as follows:

- We propose DeePEB for PEB simulation, a physics-inspired neural network architecture to learn both low-frequency and high-frequency information by combining neural operator and convolution operations. To the best of our knowledge, this is the first work that seeks to apply physics-inspired deep learning algorithms for PEB simulation, with the potential of accelerating lithography simulation and layout optimization.
- We customize a superior norm loss for optimization of DeePEB, and validate its superiority over the widely-used mean squared loss.
- We evaluate DeePEB compared to the industry verified lithography simulation software S-Litho. DeePEB can give a prediction with >100x acceleration (as shown in Fig. 2), with the critical dimension error $\sim$ 1 nm at both X and Y directions. DeePEB also outperforms other learning-based methods with the normalized root mean squared error to be 5.7% and 1.1% for PEB latent image and corresponding development rate respectively.

The rest of this paper is organized as follows. Section 2 reviews the basic concepts and formulates the problems in our work. Section 3 provides a through explanation of the proposed DeePEB framework and optimization strategy. Section 4 demonstrates the effectiveness of our approaches with comprehensive results, followed by the conclusion in Section 5.

2 PRELIMINARIES

In this section, we will first set up the PDEs describing the PEB process in 2.1 and the development model in 2.2. Progress on Neural network-based PDE solvers is then summarized in 2.3. Finally we will formulate the problem discussed in this paper.

2.1 PDEs describing PEB

There exist various types of photoresists in lithography technology. Among them, positive-tone CARs are widely utilized in KrF (λ = 248 nm) and ArF (193 nm) exposure to meet sensitivity requirements in advanced technology nodes. They are normally composed of inhibitors of dissolution (typically polymer resin, impeded to development), light-sensitive agent called photoacid generator (PAG), base quencher, and other additives [16]. As soon as photoacid is generated by light-induced decomposition of PAG and the photoresist is heated, acid-catalyzed deprotection of inhibitor takes place. A tiny concentration of photoacid can induce complete decomposition in a large area, known as chemical amplification. This can be described by the common catalytic reaction equation [14]:

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

where [I], [A] represent the normalized concentration distribution of inhibitor and photoacid, and k_c is the reaction coefficient.

Meanwhile, a portion of photoacid will vanish due to thermal-induced absorption/evaporation (in/out-diffusion) and neutralization with the base quencher. In addition, heating of the photoresist will cause diffusion of photoacid and base molecules. The reaction-diffusion process can be simplified and described by [14]:

$$\frac{\partial[A]}{\partial t} = -k_r[A][B] + \nabla(D_A\nabla[A]) \tag{2a}$$

$$\frac{\partial[B]}{\partial t} = -k_r[A][B] + \nabla(D_B\nabla[B]) \tag{2b}$$

where [B] represent the normalized concentration distribution of base, k_r is the reaction coefficient, and D_A, D_B specifies the diffusion coefficients of photoacid and base. The diffusion terms are included in the most general form. Diffusion length is an important metric describing the diffusion process, derived from the diffusion coefficient D as: $L = \sqrt{2 \cdot DT_{PEB}}$ (T_{PEB} is the PEB duration time).

The above PEB equations are augmented by initial conditions (I.C.) for all components, and by suitable boundary conditions (B.C.) for the diffusing species. In the lateral direction, both periodicity B.C. and mirror B.C. are usually imposed to ensure the conservation of compounds, and the latter is more frequently used. Appropriate choice of the simulation domain is vital for avoiding the unrealistic solution at the boundary. In the vertical direction, in-/out-diffusion of photoacid molecules can be modeled by the Robin B.C.:

$$D_A \times \left.\frac{\partial[A]}{\partial z}\right|_{\text{top}} = h\left([A]_{\text{top}} - a^*\right) \tag{3}$$

here a^* is the saturation concentration, and h the sorption constant.

Typically, PEB simulation is initiated with spatially uniform base and inhibitor distribution ($[B(t=0)]$, $[I(t=0)]$), while the initial photoacid distribution $[A(t=0)]$ is calculated from 3D aerial image according to the Dill Model [17].

2.2 Development Models

The development stage following PEB is a surface-limited etching process. The development rate $R[I(x,y,z)]$ is assumed to be an isotropic function of the local inhibitor concentration only. The propagation of the development front $S_d(x,y,z)$ (photoresist profile at certain development arrival time t_d) can be described formally by the Eikonal equation [1]: $|\nabla S_{\mathrm{d}}(x,y,z)|^2 = \frac{1}{R[I(x,y,z)]^2}$.

Among the many models describing $R[I(x,y,z)]$, the Mack model and its variants receive great popularity [18]:

$$R(x,y,z) = R_{\max}\frac{(a+1)(1-[I])^n}{a+(1-[I])^n} + R_{\min},\ a=(1-M_{th})^n\frac{n+1}{n-1}$$

here R_{min} and R_{max} specifies the minimum and maximum development rate, M_{th} an experimentally determined threshold, and n the developer selectivity. Fast marching level set algorithm [19] is often adopted to simulate the development arrival time distribution and thus the final photoresist profile (the undeveloped part with arrival time larger than the development duration). In this paper we use the open-sourced tool Structured Eikonal Solver [20] for development simulation.

2.3 Neural network-based PDE solver

Recent years have witnessed rapid progress in NN-based PDE solvers. These methods can be roughly classified into the following categories according to their fundamental principles:

The neural surrogate models [21, 22] construct NNs as surrogate solutions of PDE(s), which are optimized according to the loss embedding PDE's governing equation(s), B.C.(s), and I.C.(s). Compared to traditional numerical algorithms, these methods are mesh-free and can provide continuous solutions, at the cost of time-consuming optimization for each new PDE instance. Such a huge cost can not meet the efficiency requirements in PEB simulation.

Finite dimension mappings establish the solution mapping between functions defined in finite-dimensional Euclidean space. Such data-driven approaches can provide solutions for a class of PDEs upon appropriate supervised training, but they are mesh-dependent and suffer from the constraint of resolution and discretization [23, 24]. PEB solvers based on these mappings are also unable to generalize between different grid sizes.

It has been proposed that a NN can approximate the mesh-free solution operator class for a class of PDEs with a single set of parameters [25]. Different from finite dimension mappings, these type of algorithms [26, 27] usually apply to different discretization schemes. Neural operators [27] stand out for their strong capability of approximation, benefiting from the iterative kernel integration. Each iterative layer follows the update rule:

$$u(x) = \sigma\left(Wv(x) + \int_D \kappa(x,y)v(y)\mathrm{d}v(y) + b(x)\right) \quad \forall x \in D \quad (4)$$

where $D \subset \mathbb{R}^d$ is the bounded function domain, v and u the input and output function of each layer, W a local linear transformation, $\kappa(x,y)$ a non-local integral kernel operator, and b the bias. Neural operators exhibit powerful efficiency and generalization ability, satisfying the requirements of PEB simulation.

2.4 Problem Formulation

In PEB simulation, what we care about most is the PEB latent image (final 3D inhibitor distribution), rather than the computation-intense intermediate states. Therefore, evaluating the performance of the PEB solver comes down to evaluating the quality of the predicted distribution of inhibitor, development rate, and the photoresist profile. Considering this, we choose the following metrics as criteria:

Root Mean Squared Error (RMSE):

It is straightforward to define such a criterion to discriminate between the predicted distribution and the ground truth:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\left\|G_{\mathrm{pred}} - G_{\mathrm{label}}\right\|^2},\ G = [I], [R]. \quad (5)$$

Normalized Root Mean Squared Error (NRMSE):

Since the value of the relative distribution could vary significantly (ten orders of magnitude, as the minimum gets very close to 0), we also choose the normalized root mean squared error (NRMSE) to quantify the model performance:

$$\mathrm{NRMSE} = \left\|G_{\mathrm{pred}} - G_{\mathrm{label}}\right\|_F / \|G_{\mathrm{label}}\|_F,\ G = [I], [R] \quad (6)$$

where $\|D\|_F = (\Sigma_{i,j} D_{i,j}^2)^{\frac{1}{2}}$ represents the Frobenius norm.

Critical Dimension (CD):

In addition to the direct comparison of inhibitor distribution and development rate, we also evaluate the simulated photoresist profile. Critical Dimension is a metric often used in both lithography simulation and industrial process development. We calculate the CD value for the contacts at different heights in the X and Y directions, and define the CD error as:

$$\mathrm{CD}_\alpha\ \mathrm{Error} = \sqrt{\frac{1}{n}(CD_{\alpha,\mathrm{pred}} - CD_{\alpha,\mathrm{label}})^2},\ \alpha = x, y \quad (7)$$

Based on these metrics, we then define the photoresist modeling problem studied in this work as follows:

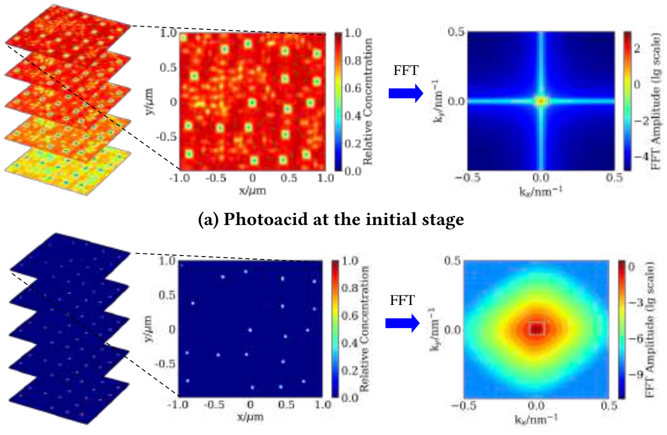
Given a set of initial photoacid distribution and corresponding final inhibitor distribution, the objective is to design and train a deep learning model that can accurately predict the inhibitor distribution of test mask patterns, with the RMSE, NRMSE, and CD error minimized, and runtime as short as possible.

3 DEEPEB FRAMEWORK

In this section, we first demonstrate the motivation of DeePEB, based on the physical essence of the PEB process. DeePEB composes of a Fourier branch learning the low-frequency features utilizing the neural operator structure, and a local branch capturing the local high-frequency behavior with convolution operations. Finally, we introduce our optimization strategy, including label normalization scheme and choice of the loss function.

3.1 Motivation

Our objective is to learn the PEB dynamics and predict the PEB latent image (i.e., the inhibitor distribution at the final stage) from the photoacid distribution at the initial stage. Mathematically speaking,



(a) Photoacid at the initial stage

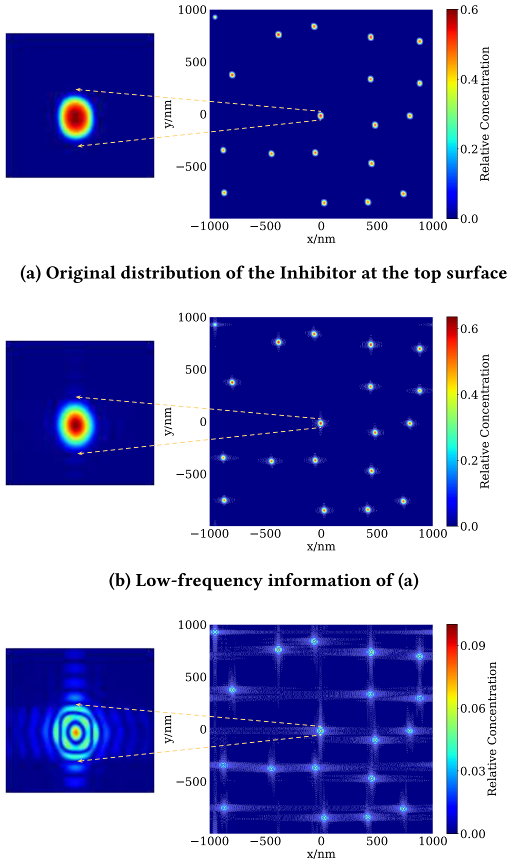
(b) Inhibitor at the final stage

Figure 3: Left column: 3D concentration distribution of photoacid and inhibitor. Middle column: top view of the 3D diagrams. Right column: Fourier information of the top surface distribution. The middle squares represent the low frequency area ($k_x, k_y < 0.05\,\text{nm}^{-1}$).

$\mathcal{G} : \mathcal{S}_{in} \rightarrow \mathcal{S}_{out}$ is the solution operator that satisfies the governing equations (Eq. 1 & 2) and corresponding boundary and initial conditions for PEB simulation, where $\mathcal{S}_{in}, \mathcal{S}_{out}$ are the function spaces defined on domain of input $[A]_{in}$ and output $[\hat{I}]_{\text{pred}}$. Our goal is to design a parameterized PEB solver $\mathcal{G}_\theta : \mathcal{S}_{in} \times \Theta \rightarrow \mathcal{S}_{out}$ that operates as similarly to $\mathcal{G}$ as possible, where $\theta \in \Theta$ is the NN parameter space. In this paper, we assume a uniform distribution of the resist, and a discrete cubic function domain is adopted. Since the photoresist spin onto the silicon substrate has a non-negligible thickness, calculating a 3D concentration distribution rather than a planar distribution uniform in the vertical direction is vital for the actual photoresist model.

To begin with, we analyze the physical essence of the PEB process. Diagrams in the left and middle column of Fig. 3 give an example of the 3D distribution of photoacid and inhibitor, derived by numerically solving the PDEs by S-Litho for one test clip. We can observe the photoacid spread all over the space except inside the contacts, and the values on the top surface are typically larger than that at the bottom surface, a result of the less sheltered exposure there. In contrast, the inhibitor concentrates around the contacts, especially at the bottom surface, since a larger photoacid concentration corresponds to a more intense local decomposition of the inhibitor. The Fast Fourier Transform (FFT) results of the planar distributions at the top surface are also provided in the right column of Fig. 3. We can vividly see that the concentration information gather in low-frequency modes, as indicated by the squares. Furthermore, the dynamic transition from the above distribution to the below one concentrates in the low-frequency modes, along with prominent information exchange between low-frequency and high-frequency modes and different heights (not shown here). This is consistent with the intuition that the evolution of reactants over time will not behave dramatically or concentrate in a small range.

We then extract the low-frequency and high-frequency information of the final inhibitor distribution, by exerting Inverse FFT (IFFT) on the low-frequency and high-frequency modes (Fourier modes inside/out-of the square in Fig. 3). The results in Fig. 4 confirm the idea that low-frequency information is well representative to restore and retain the global pattern, and high-frequency modes affect the distribution around the contacts, exhibiting small-amplitude ripple features. What's more, the spatial period of the ripple is close to the lateral diffusion length of the acid (~ 10 nm). We then propose to learn the PEB process with two parts: Fourier learning branch to learn the global features in the Fourier domain, and the local learning branch to learn the limpid details around contacts with customized convolution operations, as illustrated in Fig. 5.



(a) Original distribution of the Inhibitor at the top surface

(b) Low-frequency information of (a)

(c) High-frequency information of (a)

Figure 4: The low-frequency (b) and high-frequency (c) information of the (top surface) inhibitor distribution (a) in Fig. 3, corresponding to IFFT amplitudes of Fourier modes inside/outside the square in the FFT amplitude pattern. Patterns of the center contact are enlarged and shown on the left.

To conclude, we construct DeePEB inspired by the following observations:

(1) most of the concentration information is contained in the low-frequency modes of the reactants;
(2) dynamics of the PEB process can be almost captured within low-frequency modes;
(3) the high-frequency information decides the local feature around the contacts.

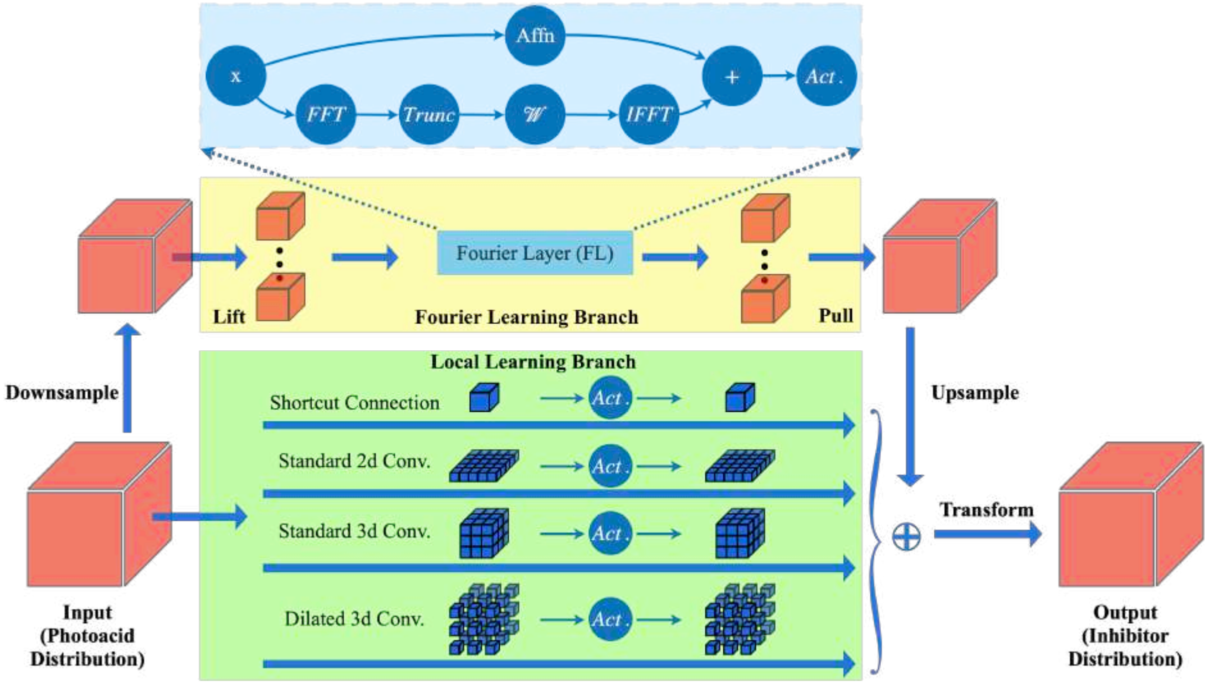


Figure 5: Schematic of DeePEB structure. In the yellow and green box are the Fourier learning branch and Local learning branch, respectively. The detailed structure of Fourier layer is shown in the blue box. "Conv." is the abbreviation of "convolution operation".

3.2 Learning Low-Frequency Information with Fourier Learning Branch

Based on the first two observations, we propose to learn the low-frequency dynamic behabvior of governing equations of PEB in the Fourier domain. Fourier learning branch is thus constructed to learn the low-frequency information motivated by Fourier Neural Operator [28, 29]. As mentioned in 2.3, a neural operator learns the representation of PDEs' solution operator $\mathcal{G}$ iteratively. FNO employs FFT to serve as the non-local integral kernel operator κ in the Fourier domain in Eq. 4.

The Fourier branch has five basic components: **(i)** downsample ($\mathcal{D}$) of the original input data to reduce the overall size with a 3d convolution operation; **(ii)** projection ($\mathcal{P}$) of the data into a higher-dimensional space (more channels than the original data) using a fully connected layer; **(iii)** one Fourier layer ($\mathcal{L}$) as the non-local kernel operator, performing both a non-local operation in the low-frequency Fourier domain and local linear operation in the spatial domain; **(iv)** projection ($\mathcal{Q}$) of the extracted features to the target dimension with a high dimension intermediate state using two fully connected layers and **(v)** upsample ($\mathcal{U}$) of the output data to the required resolution. The analytic expression of this branch reads (for input x):

$$\mathcal{F}_\theta(x) := \mathcal{U}(\mathcal{Q}(\mathcal{L}(\mathcal{P}(\mathcal{D}(x))))). \tag{8}$$

In this branch, the Fourier layer plays a key role to establish the solution mapping between the input and output functions. The basic structure of a Fourier layer is illustrated in the blue box in Fig. 5. Li et al. [28] formulate its analytical expression to be:

$$\mathcal{L}(x) = \sigma\left(\left(\mathcal{A} + FFT^{-1}\left(\mathcal{W} \cdot FFT\right)\right)(x)\right) \tag{9}$$

where $\mathcal{A}$ is the point-wise affine transformation, $\mathcal{W}$ the channel-wise linear transformation, and σ the activation function. To be more concrete, the input $\mathbf{x}$ undergoes two paths: in the top path, **Affn** operation ($\mathcal{A}$) defines a spatial affine mapping to inherit spatial information; while in the bottom path, the information of low-frequency dynamics in $\mathbf{x}$ is learned. Firstly, **FFT** operation extracts the frequency information in all spatial dimensions as new features: $\mathbf{x}(x, y, z) \xrightarrow{FFT} \widetilde{\mathbf{x}}(k_x, k_y, k_z)$, which are more representative than purely spatial domain features, as discussed before. Then operation **Trunc** discards all high-frequency components in the lateral directions, while all features in the vertical direction are retained: $\widetilde{\mathbf{x}}(k_x, k_y, k_z) \equiv 0,\ \forall k_{x,y} > 0.05\,\mathrm{nm}^{-1}$. Linear transformation $\mathcal{W}$ is applied to transfer information between remaining Fourier modes and different channels. Following operation **IFFT** transforms all features back to the spatial domain. Finally, the results of two paths are added and non-linearly activated by **Act.** (σ). Iterative kernel integration using Fourier layers is proven to be expressive enough to approximate any measurable operator mapping [29]. In this work, we find that simply one layer possesses sufficient learning ability.

As to the B.C., according to our experiments, the mirror B.C. on the lateral boundary is well learned as there exist almost no intense reactions. While in the vertical direction, the Robin B.C. is also captured since no truncation operates in the vertical direction.

3.3 Learning High-Frequency Information with Customized Convolution Layers

The local high-frequency information is under-learned by the Fourier layer discussed above, which focuses on the low-frequency dynamics and discards all high-frequency modes. To extract more local high-frequency information, we customize a local learning branch combining various types of convolution operations.

In the first place, we decide to keep the resolution of the original distribution since any operation related to downsampling will always lead to information loss, especially high-frequency information. Besides, a receptive field larger than 10 nm in the lateral direction is needed in order to learn the information within a whole "ripple", as discovered in Fig. 4(c). In the vertical direction, convolutions with large kernel size also enhance the learning of both the vertical diffusion and the Robin B.C. at the top surface. However, large standard (dilation-free) kernels combined with convolutions without downsampling will bring about a high computational burden and GPU memory usage, reducing the efficiency.

For this reason, a dilated 3d kernel combined with 3 standard kernels is adopted, processing the raw input data in parallel without downsampling. The green box in Fig. 5 illustrates these customized convolution layers: They work in parallel and bear a nonlinear activation operation **Act.** between two successive convolution operations. The three standard kernels learn the information within nearest neighbors, while the dilated 3d convolution enlarges the perceptive field to 10 nm in both directions (since the resolution of input data is 2 nm/pixel), with a significantly lower computation load compared to stack standard 3d kernel of size 5. In this way DeePEB can perceive the local high-frequency information, compensating for the disadvantages of simple Fourier layer.

To sum up, we formulate the parameterized PEB solver $\mathcal{G}_\theta$ from input to output ($\mathcal{G}_\theta : \mathcal{S}_{in} \times \Theta \rightarrow \mathcal{S}_{out}$) formally:

$$\begin{aligned} \hat{I}(\theta)_{\text{pred}} &= \mathcal{G}_\theta([A]_{in}) \\ &= \mathcal{F}_\theta([A]_{in}) + \Sigma_{i=1}^{4}(C_i \otimes (\sigma(C_i \otimes [A]_{in})) \end{aligned} \tag{10}$$

where $\mathcal{S}_{in}, \mathcal{S}_{out}$ are finite dimension spaces corresponding to domain of input $[A]_{in}$ and output functions $\hat{I}(\theta)_{pred}$, $\theta \in \Theta$ is the NN parameter space. $\mathcal{F}_\theta$ is defined in Eq. 8 and C_i, $(i = 1 - 4)$ are the convolution kernels, with $\otimes$ the corresponding convolution operation.

3.4 Label Normalization

Optimizing the neural network comes down to solving the empirical-cost minimization problem, that is, minimizing the target cost function (loss function concerning the predicted inhibitor concentration and the ground truth in our problem) by optimizing the parameters of DeePEB. Instead of directly comparing the output of NN with the golden truth concentration, we apply a quadratic negative logarithm transformation on the values of the golden truth, transforming it from $[I]$ to $-ln(-ln([I])/k_1)$. We propose this transformation based on that Eq.1 can be formally solved as:

$$[I](t) = [I](t=0) \cdot e^{-k_c \int_0^t [A](t')dt'} \tag{11}$$

For one thing, this transformation can avoid direct fitting of golden truth, featuring a value range of more than ten orders, a result of the exponential calculation in Eq. 11. For another, the catalysis equation describing the evolution of inhibitor and the reaction-diffusion equations for photoacid and base are then decoupled: any change in the catalysis velocity k_c or the initial photoacid distribution $[I](x, y, z, t = 0)$ will not lead to a change in the output, since the NN is merely learning the reaction-diffusion process and predicting $-ln(\int_0^{T_{PEB}} [A](t')dt')$ from $[A](t = 0)$ now.

3.5 Loss Function

Many functions have been reported to work as the criteria for the training of NN. In DeePEB we choose the superior norm loss (maximum squared error in practice), rather than the mean squared loss, which is commonly employed in the regression problem. Superior norm loss seeks to minimize the maximum deviation between the predicted result and the golden truth. Theoretically, the sup-norm loss can provide both a stronger distance measure and a more balanced prediction compared to the mean squared error, as suggested in [29]. We experimentally validate this idea, by comparing the performance of sup-norm loss and mean squared loss in 4.3.

To conclude, the training target during the learning process of DeePEB reads:

$$\underset{\theta}{min}\, \mathbb{E}_{([A],[I]) \in \mathcal{D}_{train}} [max||\hat{\mathrm{I}}(\theta)_{\text{pred}} - (-\mathrm{Ln}(-\mathrm{Ln}([\mathrm{I}])/\mathrm{k}_1)||^2] \tag{12}$$

where θ represents the undetermined parameters in the NN, $\mathbb{E}$ is the expectation of loss results of training dataset $\mathcal{D}_{train}$, and $\hat{I}(\theta)_{\text{pred}}$, $[I]$ are the predicted and labeled data tensor corresponding to input tensor $[A]_{in}$. Eventually, the predicted final inhibitor distribution (PEB latent image) is computed as:

$$[\mathrm{I}]_{\text{pred}} = \mathrm{Exp}(-\mathrm{k}_1 \times \mathrm{Exp}(-\hat{\mathrm{I}}(\theta)_{\text{pred}})) \tag{13}$$

4 EXPERIMENTAL RESULTS

In this section, we first elaborate on the experimental setup in our work. Then we compare the performance of DeePEB with other learning-based algorithms. Finally, we investigate the results of DeePEB to provide a deeper understanding of our algorithm.

4.1 Experimental Setup

We obtain 100 (enough in our experiments) mask clips of size 2 × 2 μm^2 from [30]. The fabricated contact size and distribution pattern of the masks are compatible with technology nodes of 28nm and below. S-Litho is employed to run the rigorous simulation. For the exposure simulation, the wavelength and numerical aperture are set to be λ = 193 nm and NA = 1.35. A simulation window of 2 × 2 μm with an optical influence range of $5\lambda/NA$ is configured and the simulation resolution is set to be 0.5 nm in all directions. While for the PEB simulation and development simulation, the resolutions in X, Y, and Z directions are set to be 2 nm, 2 nm, and 1 nm respectively. Important parameters in the simulation are summarized in Table 1.

Resist			
Resist thickness	80 nm	SiON Thickness	30 nm
PEB			
Normal Diffusion Length $L_{N,A}, L_{N,B}$	70, 15 nm	Lateral Diffusion Length $L_{L,A}, L_{L,B}$	10, 10 nm
catalysis rate k_c	0.9 /s	reaction rate k_r	8.6993 /s
sorption constant h_A, h_B	0.027, 0	saturation concentration a^*, b^*	0.9, 0
[I](t=0)	1.0	[B](t=0)	0.4
Baseline Time step	0.1 s	Duration	90 s
Develop			
R_{max}	40 nm/s	R_{min}	0.0003 nm/s
M_{th}	0.5	n	30
Duration	60 s		

Table 1: Critical parameters in photoresist simulation.

For all experiments, we use the Adam optimizer [31] to train for 500 steps, with the initial learning rate being 0.01, and decaying using the step function with the step size of 50 and decay ratio of 0.7.

We use LeakyReLU as the non-linear activation function instead of ReLU. Models are implemented with PyTorch [32], and trained on two Nvidia A40 GPUs for about 16 h. We randomly sample 80% of the data for training while the remaining clips are used for testing. We set the batch size to be 30. For the restriction of GPU memory, we sum up the gradient of 30 clips one by one and finally update the model during each step.

4.2 Performance Comparison between ML-based PEB Solvers

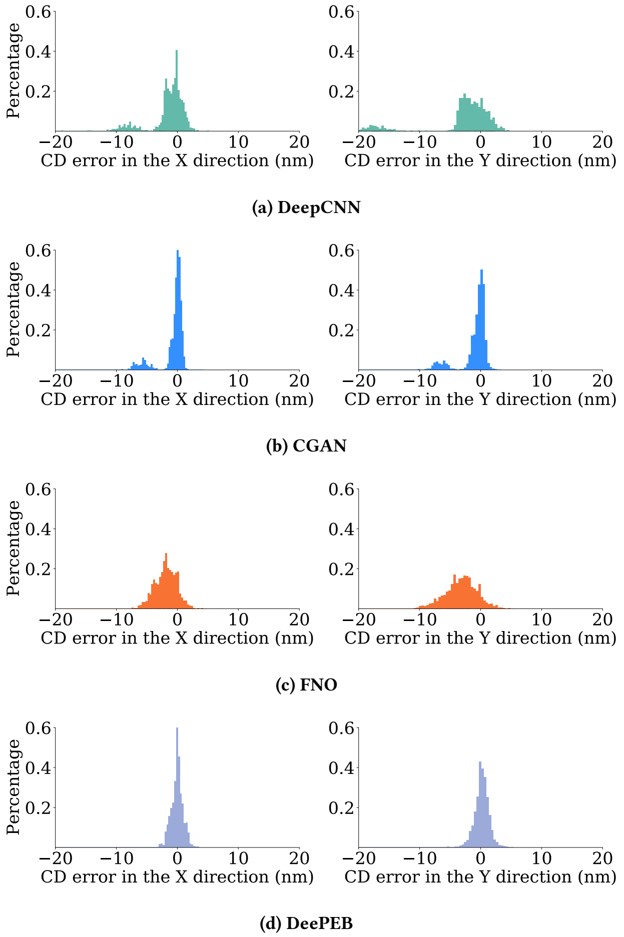
We first compare DeePEB with other ML-based photoresist models: DeepCNN, CGAN, and FNO. DeepCNN is modified from [33], where we come up with a ResNet ([34])-like structure to adapt to our problem. CGAN (Conditional-Generative-Adversarial-Network [35]) is adapted from TEMPO [4] to accommodate our 3D settings. All these networks are trained with label normalization and mean squared loss. More details can be found in our implementations [1].

Table 2 provides a detailed comparison between different methods on the test dataset using the criteria defined in 2.4 (RMSE, NRMSE, and CD Error). DeePEB outperforms other learning-based methods in all error-concerning metrics. In addition to the direct comparison of 3D concentration and rate distribution, we also evaluate the effectiveness of our proposed methods based on the quality of generated resist profiles by the CD error. Fig. 6 shows the normalized count of CD errors of different methods: The CD errors of DeePEB concentrate in $0 \sim 1$ nm, while that of the other methods spread over a larger range of values. Meanwhile, the CD errors of all methods center at the negative axis, representing that the predicted CD values are usually smaller than the golden truth. As demonstrated in the following section, this is a result of the under-learning of high-frequency components, which leads to a larger development rate at the edge area and thus a smaller developed contact size.

What's more, DeePEB consumes ~ 1.22 s in average compared to ~ 147 s of S-Litho, realizing a speed up as high as $> 100\times$ while keeping high accuracy. The significant acceleration benefits from the well-optimized FFT, convolution, and matrix multiplication operation implemented in GPU. Considering the acceptable CD error, DeePEB is strongly recommended for the early stage lithography simulation in mask optimization and hotspot detection where speed has higher priority than accuracy.

4.3 Analysis of Results and Ablation Study

We then delve into the results of DeePEB. Table 3 provides the comparison between prediction and ground-truth of inhibitor and development rate distribution at the bottom and top surface respectively, for the same test clip of Fig. 3. We can find that the prediction results resemble the actual distributions to a great extent. Specifically, the NRMSE of predicted Inhibitor concentration at the bottom and top area (both 10 nm in height) are 6.11% and 8.77%, and the NRMSE of development rate at the top area is 1.20%. Most of the prediction error concentrates at the edge of the contacts as expected: the inhibitor within contacts of the golden truth exhibits plateaus distribution, featuring intense change at the contact edge and prominent high-frequency modes correspondingly. In DeePEB, the complex reactions among these high-frequency modes are learned with customized convolution operations, while the longer-range interactions are almost ignored, leaving directions for future improvements. Besides, the relative error between the predicted inhibitor concentration and the golden truth at the top area ($\sim 8\%$) is larger than that at the whole domain ($\sim 6\%$). This is owing to the smaller area of non-zero values at the top, rather than under-learning of the top B.C. of photoacid.



(a) DeepCNN

(b) CGAN

(c) FNO

(d) DeePEB

Figure 6: Percentage count of the CD error in the X and Y direction of different methods.

Luckily, the under-learning of the high-frequency information does not bring much error to the final predicted development rate and so the resist profile, as shown in the predicted rate distribution at the top surface in Table 3. This comes from the saturation of development rate for both high (gets R_{min} when [I]>0.7) and low (R_{max} when [I]<0.4) inhibitor concentration values, smoothing out the variation of the under-fitting.

Finally, we conduct 4 groups of ablation experiments to study the effectiveness of each technique utilized in DeePEB architecture. Superior norm loss is replaced with the mean squared loss, which is commonly used in the regression problem, for groups 1, 2, and 3. Their detailed implementation and results are summarized in Table 4. We can observe that each technique employed in DeePEB polish its performance, and the local learning branch do help DeePEB to capture more information from the data.

[1] https://github.com/Brilight/DeePEB.git

Algorithm	Inhibitor		Rate		CD Error/nm		Runtime/s
	RMSE/E-3	NRMSE/%	RMSE	NRMSE/%	x	y	
DeepCNN [33]	8.25	12.53	0.65	1.63	3.14	6.26	0.99
CGAN [4]	7.67	12.55	0.50	1.26	2.12	2.45	5.25
FNO [28]	7.91	11.68	0.68	1.69	2.34	3.71	0.97
DeePEB	3.99	5.70	0.48	1.19	0.98	1.24	1.22

Table 2: Performance comparison between different PEB solvers. Here runtime contains both the time of data processing and transfering between CPU and GPU.

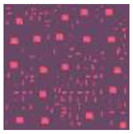
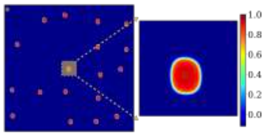
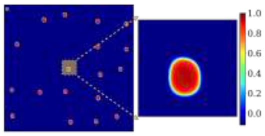
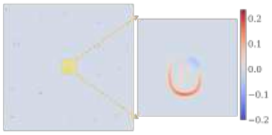
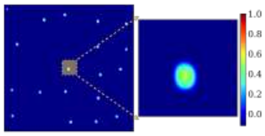
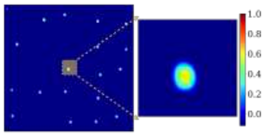
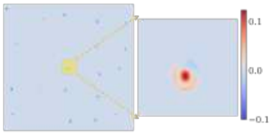
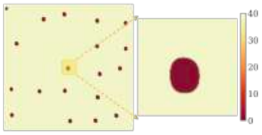
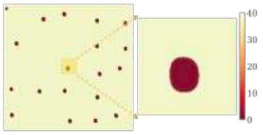
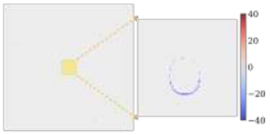
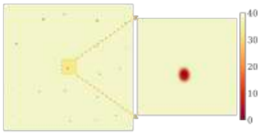
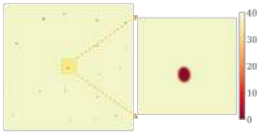
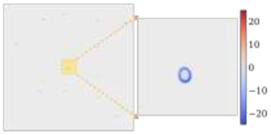
Mask	Distribution Type	Golden Truth	Prediction	Error
		Global Center	Global Center	Global Center
	Inhibitor at Bottom			
	Inhibitor at Top			
	Rate at Bottom			
	Rate at Top			

Table 3: An example of the predicted distribution of Inhibitor and development rate at different heights (the top surface and the bottom surface). Corresponding golden truth and the error between them ($[G]_{\text{pred}} - [G]_{\text{label}}$) are also provided. Enlarged pictures of the distribution around the center contact and the color bar are displayed on the right of each figure.

	Technique				NRMSE/%		CD Error/nm	
	Fourier Branch	Label Normalization	Local Learning	Superior Norm Loss	Inhibitor	Rate	X	Y
1	✓				15.36	2.00	3.32	4.75
2	✓	✓			11.68	1.69	2.34	3.71
3	✓	✓	✓		10.41	1.70	1.79	2.63
4	✓	✓	✓	✓	5.70	1.19	0.98	1.24

Table 4: Results of the ablation study.

5 CONCLUSION

PEB simulation acts as the bridge between the aerial image and the final resist profile in lithography simulation. To accelerate the PEB simulation without sacrificing accuracy, we propose DeePEB, a neural PDE solver. We construct DeePEB based on the observation of the physical essence of PEB: most of the dynamic information of the PEB process is contained in low-frequency modes of related reactants, and the high-frequency information affects the local features. So we combine both neural operator and customized convolution operations for learning the solution operator of PEB. Our algorithm is validated with an industry-strength software S-Litho under real manufacturing conditions, exhibiting high efficiency and accuracy. Comparison with other learning-based methods further demonstrates its power. We believe that DeePEB can shed more light on lithography simulation and early-stage mask optimization due to its performance and runtime advantages.

ACKNOWLEDGMENT

The authors would like to thank Xiangyu Zhou and Mariya Braylovska from Synopsys, and Xinming Wei and Zizheng Guo from Peking University for helpful discussions. This project is supported in part by the NSFC (62034007), NSFC (62125401), and the 111 Projects (B18001).

REFERENCES

[1] C. A. Mack, *Fundamental Principles of Optical Lithography: The Science of Microfabrication*, 2007.

[2] Synopsys, "Sentaurus Lithography," https://www.synopsys.com/silicon/mask-synthesis/sentaurus-lithography.html, 2016.

[3] W. Ye, M. B. Alawieh, Y. Lin, and D. Z. Pan, "Lithogan: End-to-end lithography modeling with generative adversarial networks," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2019, pp. 1–6.

[4] W. Ye, M. B. Alawieh, Y. Watanabe, S. Nojima, Y. Lin, and D. Z. Pan, "Tempo: Fast mask topography effect modeling with deep learning," in *Proceedings of the 2020 International Symposium on Physical Design*, 2020, pp. 127–134.

[5] H. Yang, Z. Li, K. Sastry, S. Mukhopadhyay, M. Kilgard, A. Anandkumar, B. Khailany, V. Singh, and H. Ren, "Generic lithography modeling with dual-band optics-inspired neural networks," in *59th Annual Design Automation Conference (DAC) 2022*, 2022.

[6] X. Zhou, M. Bohn, and M. Braylovska, "Lithography simulation using machine learning," Apr. 21 2022, uS Patent App. 17/467,682.

[7] H. Yang, S. Jing, Z. Yi, Y. Bei, and E. Young, "Layout hotspot detection with feature tensor generation and deep biased learning," in *the 54th Annual Design Automation Conference 2017*, 2017.

[8] Ding, Duo, Torres, J., Andres, Pan, David, and Z., "High performance lithography hotspot detection with successively refined pattern identifications and machine learning." *tcad*, vol. 30, no. 11, pp. 1621–1634, 2011.

[9] S. Shim, S. Choi, and Y. Shin, "Machine learning-based 3D resist model," in *Optical Microlithography XXX*, A. Erdmann and J. Kye, Eds., vol. 10147, International Society for Optics and Photonics. SPIE, 2017, pp. 408 – 417. [Online]. Available: https://doi.org/10.1117/12.2257904

[10] Y. Watanabe, T. Kimura, T. Matsunawa, and S. Nojima, "Accurate lithography simulation model based on convolutional neural networks," in *Optical Microlithography XXX*, vol. 10147. International Society for Optics and Photonics, 2017, p. 101470K.

[11] R. A. Ferguson, J. M. Hutchinson, C. A. Spence, and A. R. Neureuther, "Modeling and simulation of a deep-ultraviolet acid hardening resist," *Journal of Vacuum Science & Technology B Microelectronics Processing & Phenomena*, vol. 8, no. 6, pp. 1423–1427, 1990.

[12] H. Fukuda and S. Okazaki, "Kinetic model and simulation for chemical amplification resists," *Journal of the Electrochemical Society*, vol. 137, no. 2, p. 675, 1990.

[13] M. Weiß, H. Binder, and R. Schwalm, "Modeling and simulation of a chemically amplified duv resist using the effective acid concept," *Microelectronic Engineering*, vol. 27, no. 1, pp. 405–408, 1995.

[14] D. Matiut, A. Erdmann, B. Tollkuehn, and A. Semmler, "New models for the simulation of post-exposure bake of chemically amplifed resists," *Proceedings of Spie the International Society for Optical Engineering*, vol. 5039, 2003.

[15] E. H. Croffie, Y. Lei, M. Cheng, and A. R. Neureuther, "Survey of chemically amplified resist models and simulator algorithms," in *International Symposium on Microlithography*, 2001.

[16] W. D. Hinsberg, F. A. Houle, M. I. Sanchez, and G. M. Wallraff, "Chemical and physical aspects of the post-exposure baking process used for positive-tone chemically amplified resists," *IBM Journal of Research & Development*, vol. 45, no. 5, pp. 667–682, 2001.

[17] F. H. Dill, W. P. Hornberger, P. S. Hauge, and J. M. Shaw, "Characterization of positive photoresist," *Electron Devices IEEE Transactions on*, vol. 22, no. 7, pp. 445–452, 1975.

[18] C. Mack, M. J. Maslow, A. Sekiguchi, and R. A. Carpio, "New model for the effect of developer temperature on photoresist dissolution," *International Society for Optics and Photonics*, vol. 3333, p. 1218, 1998.

[19] J. A. Sethian, "Fast-marching level-set methods for three-dimensional photolithography development," *Proc Spie*, vol. 2726, pp. 262–272, 1996.

[20] W.-K. Jeong and R. T. Whitaker, "A fast iterative method for eikonal equations," *SIAM Journal on Scientific Computing*, vol. 30, no. 5, pp. 2512–2534, 2008. [Online]. Available: https://github.com/SCIInstitute/StructuredEikonal

[21] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999118307125

[22] . Weinan, E. and B. Yu, "The deep ritz method: A deep learning-based numerical algorithm for solving variational problems," *Communications in Mathematics & Statistics*, vol. 6, no. 1, pp. 1–12, 2018.

[23] X. Guo, W. Li, and F. Iorio, "Convolutional neural networks for steady flow approximation," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 481–490.

[24] A. Li, R. Chen, A. B. Farimani, and Y. J. Zhang, "Reaction diffusion system prediction based on convolutional neural network," *Scientific Reports*, vol. 10, no. 1, p. 3894, 2020.

[25] T. Chen and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE Trans Neural Netw*, vol. 6, no. 4, pp. 911–917, 1995.

[26] L. Lu, P. Jin, and G. E. Karniadakis, "Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators," 2019.

[27] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, and A. Anandkumar, "Neural operator: Learning maps between function spaces," 2021.

[28] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier neural operator for parametric partial differential equations," *arXiv preprint arXiv:2010.08895*, 2020.

[29] N. Kovachki, S. Lanthaler, and S. Mishra, "On universal approximation and error bounds for fourier neural operators," 2021.

[30] Y. Lin, M. Li, Y. Watanabe, T. Kimura, T. Matsunawa, S. Nojima, and D. Z. Pan, "Data efficient lithography modeling with transfer learning and active data selection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 10, pp. 1900–1913, 2019.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[33] Y. Watanabe, T. Kimura, T. Matsunawa, and S. Nojima, "Accurate lithography simulation model based on convolutional neural networks," in *Spie Advanced Lithography*, 2017.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[35] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *Computer Science*, pp. 2672–2680, 2014.