

# Optical Diffraction-based Convolution for Semiconductor Lithography

Young-Han Son<sup>1</sup> Dong-Hee Shin<sup>2\*</sup> Deok-Joong Lee<sup>1</sup> Hyun Jung Lee<sup>1</sup> Tae-Eui Kam<sup>1\*</sup>

<sup>1</sup>Korea University <sup>2</sup>Yonsei University  
yhson135@korea.ac.kr \*Corresponding author

## Abstract

*In recent years, the increasing demand for smaller and more powerful semiconductors highlighted the critical role of lithography—a key stage in semiconductor manufacturing responsible for precise mask design and wafer patterning. To meet these demands, the semiconductor industry has increasingly adopted computational lithography, employing machine learning and deep learning techniques to accelerate advancements in lithographic technology. Despite the various research efforts and successes in computational lithography, there remains a lack of explicit incorporation of physical principles. This gap limits the ability of existing methods to fully capture the complex physical phenomena inherent in lithography behaviors. To bridge this gap, we propose **OptiCo**, a novel convolutional neural network that seamlessly integrates optical diffraction principles into its architecture. At its core, OptiCo employs an optical phase kernel to model phase variations resulting from light propagation, effectively capturing the physical interactions among light, masks, and wafers. We evaluate OptiCo on semiconductor lithography benchmarks, demonstrating its superior performance in mask optimization tasks, with its remarkable generalization capabilities in OOD datasets.*

## 1. Introduction

Semiconductors are now indispensable to almost every facet of modern life, enabling technologies from smartphones and computers to artificial intelligence systems. Their critical role has brought semiconductor manufacturing to the forefront of technological innovation and industrial focus. Within this manufacturing ecosystem, lithography stands out as one of the most important stages [32, 55, 64]. In particular, the lithography process uses light passing through a mask to precisely project the patterns of transistors and circuits onto silicon wafers. Due to its technical demands and the precision required, the lithography stage accounts for approximately 30% of the overall manufacturing cost [29].

Over the years, the increasing demand for smaller and more powerful semiconductors has pushed lithography

technology to its physical limits. As manufacturers strive to further reduce circuit sizes, they face substantial challenges from the complex interactions between light, wafer, and mask patterns [1, 5, 15]. As shown in Figure 1, the image projected from the mask onto the wafer typically deviates from the intended design due to the effects of light diffraction, which are particularly pronounced with short-wavelength light (e.g., EUV light) used for fabricating smaller circuit feature sizes [47].

To achieve the precision in lithography, traditional approaches have relied on trial-and-error experimentation or mathematical modeling to optimize lithographic parameters [14, 30, 52, 62]. However, repeatedly executing the real lithography process is prohibitively expensive, and the complexity of light diffraction makes mathematical modeling both challenging [3, 29]. To address these constraints, the semiconductor industry has adopted simulation-based computational lithography, particularly those leveraging deep learning (DL) methods [2, 40, 46].

Among the various DL-based approaches, GAN-OPC [53] emerged as one of the earliest DL frameworks that utilize a generative adversarial network (GAN) to establish a direct mapping between input target wafer patterns and output mask designs. Building upon this foundation, DAMO [7] enhanced performance by achieving higher-resolution mask predictions by incorporating a UNet++ [66] backbone and residual blocks. TEMPO [58] further leveraged multi-domain image-to-image translation techniques to address the complexities associated with 3D mask designs.

Despite these achievements, some studies [50, 54] have highlighted limitations in their reliance on standard neural network architectures, which may struggle to capture the global information essential for accurate lithography modeling [20]. To overcome this limitation, researchers have explored the potential of Fourier Neural Operators (FNOs) [23]. Specifically, DOINN [54] introduced an FNO-based framework that employs Fourier transforms to effectively capture both low-frequency global information and high-frequency local details. Building on this approach, CFNO [50] further improved the framework by incorporating inductive lithography bias into the model architecture. While

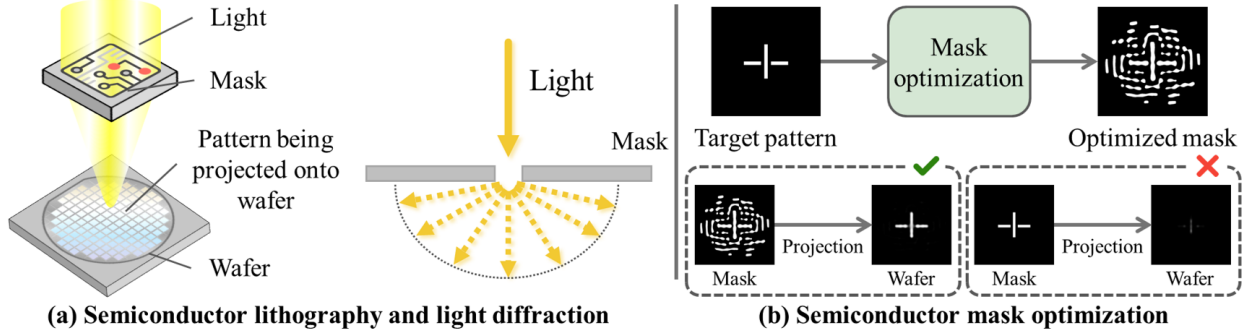


Figure 1. Illustrations of semiconductor lithography and mask optimization.

these FNO-based frameworks have demonstrated superior performance by implicitly modeling lithography in Fourier space, they did not explicitly account for the physical principles of optical diffraction theory [11, 19] that fundamentally govern lithography behavior.

To address these limitations, we propose an optical physics-inspired neural network that incorporates fundamental optical principles through the optical diffraction [4, 41]. In the context of semiconductor lithography, diffraction is particularly crucial as it determines how light propagates through the mask and ultimately forms patterns on silicon wafers [32]. A key aspect of light diffraction is the phase factor, which quantifies changes in the wavefront of light when it encounters different material interfaces [49]. For instance, as light interacts with the transparent and opaque regions of a mask, the phase factor predicts how the phase and amplitude of light waves are altered during these interactions, which influence the pattern of light projected onto wafers [13].

To model this phase factor within our neural network, we introduce an *Optical Phase (OP) kernel* that functions as a transformation layer to encode the spatial phase variations induced by light propagation. Our framework is designed to be formulation-flexible, supporting both Rayleigh–Sommerfeld-based diffraction formulations and the Hopkins TCC kernel, while remaining flexible to other kernels. Unlike traditional convolution filters that focus only on spatial features, our kernel explicitly accounts for phase variations induced by light propagation. By incorporating this physics-inspired operation into our architecture, we enhance its ability to generalize to out-of-distribution (OOD) mask designs. This capability is crucial for real-world semiconductor manufacturing, where mask patterns often deviate from the training data due to practical constraints. In addition, we adopt a simple Total Variation (TV) loss to suppress artifacts and smooth mask boundaries, further improving the manufacturability of the resulting masks.

Our main contributions can be summarized as follows:

- We propose a **Optical** diffraction-based **Convolutional** neural network (**OptiCo**) designed to effectively simulate the behavior of light for computational lithography.

- To the best of our knowledge, OptiCo is the first convolutional neural network framework to explicitly integrate the physical principles of optical diffraction into its core operations, thereby ensuring that the model inherently respects the underlying physics that govern lithography.
- We introduce a *optical phase (OP) kernel* that encodes spatial phase variations caused by light propagation. This physics-inspired and formulation-flexible OP kernel significantly improves the generalization performance for OOD mask designs, while the addition of a TV loss further enhances mask manufacturability.

## 2. Preliminaries

**Semiconductor lithography.** The primary goal of semiconductor lithography is to accurately model the interaction between mask patterns and wafers through optical projection systems. The mask  $M(x, y)$  can be defined as a binary function representing opaque and transparent regions, where  $(x, y)$  denotes spatial coordinates on the mask plane. The mask pattern is projected onto the wafer through an optical projection framework  $H$  [20, 44], producing an aerial image  $I(x', y') = H \cdot M(x, y)$  that represents the light intensity distribution. Here,  $(x', y')$  denotes spatial coordinates on the wafer.

The final wafer pattern is determined by the interaction between the aerial image  $I$  and the photoresist material on the wafer. This interaction is commonly modeled using a variable threshold resist (VTR) model [35], which captures how light intensity translates into physical features after development. In the VTR model, a dose function  $D(x', y') = I(x', y') \otimes G$  is computed, where  $G$  represents a spatial kernel encoding the effects of photoresist properties and exposure variability. The resulting resist image  $R$  is then obtained as a binary map by applying a dose threshold  $D_{th}$ , such as:

$$R(x', y') = \begin{cases} 1, & \text{if } D(x', y') \geq D_{th}, \\ 0, & \text{if } D(x', y') < D_{th}. \end{cases} \quad (1)$$

This overall process from a mask  $M$  to the resist image  $R$  is represented as a lithography simulator  $g$ .

**Problem formulation.** The *mask optimization* task aims to find an optimal mask  $M^*$  that produces a resist image that closely matches a desired target wafer pattern  $R^*$ ,

$$M^* = \arg \min_M \mathcal{L}(g(M), R^*), \quad (2)$$

where  $\mathcal{L}$  is the loss function measuring the difference between the simulated resist image  $R = g(M)$  and the target pattern. Conversely, the *lithography simulation* task focuses on predicting the resist image from a given mask. Accurate simulation is essential not only for validating designs but also for enabling efficient mask optimization, as it serves as the forward model in the optimization loop.

**Physics-inspired lithography models.** Early DL-based approaches to semiconductor lithography [7, 53, 57] primarily relied on architectures such as UNet++ and CGAN, focusing purely on data-driven learning without incorporating domain-specific physics. However, recent advancements have begun to integrate physics-inspired priors to improve model performance and generalizability. For example, DOINN [54] and CFNO [50] leverage FNOs [23] to implicitly encode optical physics in the frequency domain, demonstrating strong results on both lithography simulation and mask optimization. ILILT [51] takes this further by embedding Inverse Lithography Technology (ILT) [67]—a physics-based framework for mask optimization—directly into the learning process. Nitho [8] introduces a coordinate-based complex MLPs inspired by optical kernel regression [10], specifically designed for lithography simulation. Although these methods embed physics priors, they rely on indirect methods such as frequency-domain modeling or complex MLPs. In contrast, our method explicitly incorporates physical principles into both the architectural design and the core operations, leading to improved generalization and robustness to OOD sets. **The necessity of modeling diffraction in mask optimization** is further examined using Sub-Resolution Assist Features (SRAFs) in Appendix F.

## 3. Method

### 3.1. Convolution Operation from Light Diffraction

In wave optics, the propagation of an optical wave  $U$  from the aperture plane  $(x, y)$  to a target plane  $(x', y')$  at distance  $z$  can be described by the Rayleigh-Sommerfeld (RS) diffraction integral [41], where  $\lambda$  denotes the wavelength and  $k = 2\pi/\lambda$  is the wavenumber.

The RS integral admits several forms, including the Helmholtz–Kirchhoff, the Green’s function, as well as the Fresnel approximation. For clarity of derivation, we illustrate the Fresnel form here, while noting that our framework is fully compatible with other RS-based formulations.

$$U(x', y') = \frac{e^{jkz}}{j\lambda z} \iint_{-\infty}^{\infty} U(x, y) e^{\frac{jk}{2z} [(x'-x)^2 + (y'-y)^2]} dx dy. \quad (3)$$

This integral reveals that the diffraction can be modeled as a convolution with an optical propagation kernel. Recall that the convolution of a function  $f$  with a kernel  $h$  is mathematically defined as:

$$(f * h)(x', y') = \iint_{-\infty}^{\infty} f(x, y) h(x' - x, y' - y) dx dy. \quad (4)$$

The diffraction integral in Eq. 3 can be rewritten by isolating the quadratic dependence on  $(x' - x)$  and  $(y' - y)$ . This transforms the integral using convolution as follows:

$$\begin{aligned} U(x', y') &= \frac{e^{jkz}}{j\lambda z} \iint_{-\infty}^{\infty} U(x, y) h(x' - x, y' - y) dx dy \\ &= \frac{e^{jkz}}{j\lambda z} [U(x, y) * h(x', y')], \end{aligned} \quad (5)$$

where  $h(x', y')$  defines the Optical Phase (OP) kernel that characterizes diffraction through phase modulation. Different RS formulations yield different expressions of the OP kernel, as summarized below (Appendix A for derivations):

$$h(x, y) = \exp\left(\frac{jk}{2z} [x^2 + y^2]\right) \quad (\text{Fresnel approximation})$$

$$h(x, y) = \frac{\exp\left(jk\sqrt{x^2 + y^2 + z^2}\right)}{x^2 + y^2 + z^2} \left[1 + \frac{j}{k\sqrt{x^2 + y^2 + z^2}}\right] \quad (\text{Helmholtz–Kirchhoff})$$

$$h(x, y) = \frac{\exp\left(jk\sqrt{x^2 + y^2 + z^2}\right)}{x^2 + y^2 + z^2} \quad (\text{Green’s function})$$

### 3.2. OP Kernel-Based Complex Convolution

Building upon the foundation established in Section 3.1, where we demonstrated that the diffraction integral can be expressed as a convolution operation with the OP kernel, we introduce a complex convolution layer that incorporates this physically grounded kernel as a physics-inspired inductive bias. Unlike traditional convolution filters that focus only on spatial features and rely entirely on data-driven learning, our formulation integrates optical priors, which enables the layer to more accurately model wave propagation and thereby enhance lithography performance.

**OP kernel construction.** For a kernel size of  $(N, N)$ , we define the coordinates  $(x, y)$  relative to the center of the kernel and compute the complex exponential to construct the OP kernel  $Q(x, y) = \exp\left(\frac{jk}{2z} [x^2 + y^2]\right)$ . Each element of  $Q(x, y)$  corresponds to the phase term of diffraction. In this work, we primarily utilize the OP kernel in Fresnel form due to its effectiveness. However, our framework is formulation-flexible and can also incorporate other alternative RS formulations, as well as the Hopkins transmission cross-coefficient (TCC) kernel. The pseudo-code for our OP kernel is provided in Appendix C.

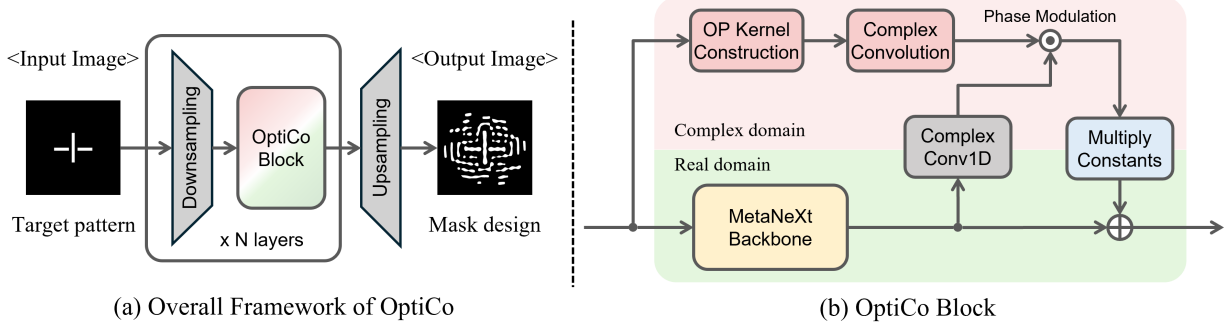


Figure 2. Overview of the OptiCo framework and the design of its block integrating OP kernel.

**OP complex convolution.** To integrate the diffraction-inspired OP kernel  $Q$  into our architecture, we can construct an effective kernel  $W_{\text{eff}}$  in two possible forms: either element-wise multiplication with a learnable complex weight  $W$ , or a stricter variant scaled only by a learnable scalar weight  $\lambda_\alpha$ :

$$W_{\text{eff}} = Q \odot W \quad \text{or} \quad W_{\text{eff}} = \lambda_\alpha \cdot Q, \quad (6)$$

where  $\odot$  denotes element-wise multiplication. The resulting complex effective kernel  $W_{\text{eff}}$  encodes optical diffraction, so that each convolution step inherently reflects the physical propagation of light.

Since optical fields are inherently complex, we decompose input  $U$  and effective kernel  $W_{\text{eff}}$  into real ( $r$ ) and imaginary ( $i$ ). The OP complex convolution is defined as:

$$U = U_r + jU_i; \quad W_{\text{eff}} = W_r + jW_i; \quad (7)$$

$$\text{OP}_{\text{conv}}(U) = \left[ (U_r * W_r - U_i * W_i) + j(U_r * W_i + U_i * W_r) \right]. \quad (8)$$

### 3.3. Architectural Design of the OptiCo Block

The OptiCo block integrates a MetaNeXt-based backbone [61] with the OP complex convolution in a unified layer. Our backbone block takes input  $U$  as:

$$Y_{\text{backbone}}(U) = \left( \text{DWConv}(\text{Norm}(U_r)W_1) \odot \sigma(\text{Norm}(U_r)W_2) \right) W_3 + U_r, \quad (9)$$

where DWConv is a depthwise convolution,  $\sigma$  an activation function, and  $W_1, W_2, W_3$  are the MLP parameters. Since the OP convolution is inherently complex, we first embed  $Y_{\text{backbone}}$  into the complex domain using a per-pixel ComplexConv1D (CCConv1D), yielding a complex feature. At each coordinate  $(p, q)$ , the channel vector  $x_{pq} \in \mathbb{R}^C$  is embedded into the complex domain through projections:

$$\text{CCConv1D}(x_{pq}) = (x_{pq,r}V_r - x_{pq,i}V_i) + j(x_{pq,r}V_i + x_{pq,i}V_r), \quad (10)$$

where  $V_r, V_i \in \mathbb{R}^{C \times C}$  are learnable projection matrices. This operation serves as a per-pixel channel embedding, producing complex-valued features that can interact with the OP kernel.

This embedding is combined with  $\text{OP}_{\text{conv}}$  via Hadamard product that performs phase modulation. The phase modulated output  $Y_{\text{phase}}$  is multiplied by the constants and finally added back to the backbone feature, injecting physically grounded diffraction effects into the mask representation.

$$Y_{\text{phase}}(U) = \frac{e^{jkz}}{j\lambda z} \left[ \text{CCConv1D}(Y_{\text{backbone}}(U)) \odot \text{OP}_{\text{conv}}(U) \right], \quad (11)$$

$$Y_{\text{OptiCo}} = Y_{\text{backbone}}(U) + Y_{\text{phase}}(U). \quad (12)$$

### 3.4. Total Variation Loss for Unwanted Mask Artifact Suppression

To account for practical constraints in mask production, we aim to not only minimize the discrepancy between the target wafer pattern  $R^*$  and the simulated resist image  $g(M)$  from the optimized mask  $M$ , but also enforce a regularization loss term to suppress *grainy* artifacts that hinder mask manufacturability. The primary loss is formally defined as:

$$\mathcal{L}_{\text{mse}}(M) = \|g(M) - R^*\|^2, \quad (13)$$

where  $g(\cdot)$  denotes the lithography simulator, which generates the corresponding resist image from a given optimized mask. To promote spatial smoothness in the mask and suppress high-frequency noise, we introduce a standard 2D total variation (TV) loss [63]:

$$\mathcal{L}_{\text{tv}}(M) = \sum_{p,q} \left| M_{p+1,q} - M_{p,q} \right| + \left| M_{p,q+1} - M_{p,q} \right|. \quad (14)$$

This loss term encourages spatial smoothness by penalizing abrupt changes between adjacent pixels, thereby reducing unwanted *grainy* artifacts. Combining these two terms yields our final objective:

$$\mathcal{L}_{\text{final}}(M) = \mathcal{L}_{\text{mse}}(M) + \lambda_{\text{tv}} \mathcal{L}_{\text{tv}}(M), \quad (15)$$

where  $\lambda_{\text{tv}}$  is the weighting parameter that controls the strength of the regularization.

Table 1. Comparative performance on mask optimization across competing methods. Lower is better.

Method	MetalSet			ViaSet			StdMetal (MetalSet OOD)			StdContact (ViaSet OOD)			Average		
	MSE	PVB	EPE	MSE	PVB	EPE	MSE	PVB	EPE	MSE	PVB	EPE	MSE	PVB	EPE
DAMO	32579	<b>41173</b>	5.4	5081	9962	<b>0.0</b>	16120	<b>23796</b>	0.2	50445	35673	26.7	26056	27651	8.1
DOINN	36409	41929	7.4	4382	7836	<b>0.0</b>	25913	25749	4.5	72058	17968	55.8	34691	<b>23370</b>	16.9
CFNO	47814	46131	12.5	8949	9890	0.1	26809	26814	4.2	70740	<b>17950</b>	55.1	38578	25196	18.0
ILILT	30353	45353	3.2	4666	10065	<b>0.0</b>	14596	24969	0.1	38957	43869	7.1	22143	31064	2.6
OptiCo	<b>24033</b>	45379	<b>1.6</b>	<b>4339</b>	<b>7802</b>	<b>0.0</b>	<b>11293</b>	25183	<b>0.0</b>	<b>18474</b>	39181	<b>0.1</b>	<b>14535</b>	29373	<b>0.4</b>

## 4. Experiments

### 4.1. Experimental Design and Configurations

**Dataset.** To evaluate the performance of our OptiCo framework, we utilize LithoBench [65], which is one of the most recent and widely recognized computational lithography benchmarks. Specifically, LithoBench provides over 100,000 layout tiles, consisting of four subtasks: two synthetic design sets (MetalSet, ViaSet) and two real-world sets treated as OOD (StdMetal, StdContact). Each subtask includes a variety of data types such as target images, reference masks, aerial images, and resist images. The resolution of each image is  $2048 \times 2048$  pixels, with each pixel representing a  $1nm^2$  of the semiconductor layout.

**Setup.** In this work, we conducted experiments primarily focused on mask optimization, where the model receives target designs as input and generates optimized masks as output. This task is essential for ensuring that the resulting lithographic masks enable the precise transfer of circuit layouts onto silicon wafers. In addition to mask optimization, we carried out an additional experiment on lithography simulation. In this task, the model takes optimized masks as input and produces the corresponding aerial and resist images as output. The primary objective of lithography simulation is to accurately predict the final wafer patterns, ensuring that the generated resist images closely align with the intended circuit layouts. We closely follow the original LithoBench hyperparameter configuration and training procedure, and the full hyperparameter settings and experimental details are provided in Appendix E.2.

**Evaluation metrics.** For the mask optimization task, we used three evaluation metrics: mean squared error (MSE), edge placement error (EPE) violations, and process variation band (PVB) area. Specifically, MSE measures the overall discrepancy between the simulated resist image and the target wafer pattern. EPE and PVB are industry-standard metrics commonly used in chip manufacturing to evaluate mask quality. EPE quantifies the deviation between the intended edge positions on the resist image and their actual positions. PVB measures the range of possible edge positions caused by manufacturing process variations. For the lithography simulation task, the performance of each

method was assessed using two evaluation metrics: MSE and intersection over union (IoU). Detailed explanations of these evaluation metrics are provided in Appendix E.1.

**Competing methods.** We compared our OptiCo framework against several prominent semiconductor lithography models from the LithoBench, as well as recent specialized models. In particular, DAMO [7] is a fully data-driven model that uses a UNet++ [66] backbone. In addition, DOINN [54] and CFNO [50] leverage FNO-based architectures that implicitly model diffraction effects, making them the most relevant comparative baselines. They transform the input image with a Fourier transform, embed it via MLP layers, and then concatenate those embedded features with local CNN outputs. Furthermore, ILILT [51] is a recent state-of-the-art mask optimization model that iteratively inputs simulated resist images along with target patterns, and Nitho [8] is a physics-inspired lithography simulation model based on optical kernel regression.

### 4.2. Experimental Results

**Mask optimization experiments.** Table 1 provides a comprehensive evaluation of all methods across different subtasks in the mask optimization task. Our OptiCo framework consistently achieves superior performance across most evaluation metrics and subtasks, highlighting the effectiveness of integrating optical diffraction principles into the neural network architecture. Notably, OptiCo demonstrates exceptional performance on the StdMetal and StdContact subtasks, which are inherently more challenging and demand high generalization capabilities for OOD tasks.

**Strengths in OOD.** In the ViaSet, nearly all methods achieve an EPE of zero, indicating strong adaptation to the training dataset. However, on the OOD test set (StdContact), most competing methods exhibit a sharp increase in EPE. In contrast, our OptiCo consistently maintains a robust performance on the OOD test set. This robustness further highlights the effectiveness of incorporating optical diffraction principles, which offer physics-inspired guidance and enhance the model’s generalization capability. Among competing methods, ILILT demonstrates strong performance. However, our OptiCo surpasses ILILT by explicitly incorporating physical principles via the OP kernel.

Table 2. Lithography simulation performance across competing methods. Lower MSE is better and higher IoU is better.

Method	MetalSet			ViaSet			StdMetal (MetalSet OOD)			StdContact (ViaSet OOD)			Average		
	MSE <sub>aerial</sub>	MSE <sub>resist</sub>	IoU	MSE <sub>aerial</sub>	MSE <sub>resist</sub>	IoU	MSE <sub>aerial</sub>	MSE <sub>resist</sub>	IoU	MSE <sub>aerial</sub>	MSE <sub>resist</sub>	IoU	MSE <sub>aerial</sub>	MSE <sub>resist</sub>	IoU
DAMO	8.36e-6	7.48e-4	0.97	2.98e-6	1.47e-4	0.94	2.50e-5	1.50e-3	0.96	4.56e-5	1.64e-3	0.87	2.05e-5	1.01e-3	0.93
DOINN	9.45e-6	6.77e-4	0.97	2.02e-6	1.04e-4	<b>0.96</b>	1.36e-5	1.29e-3	0.96	2.72e-5	1.37e-3	0.89	1.31e-5	8.61e-4	0.95
CFNO	1.87e-5	1.47e-3	0.94	3.77e-6	2.13e-4	0.92	2.61e-5	2.29e-3	0.93	2.14e-5	2.20e-3	0.83	1.75e-5	1.54e-3	0.90
Nitho	9.27e-6	7.95e-4	0.97	2.24e-6	1.46e-4	0.95	2.80e-5	1.94e-3	0.95	2.22e-5	1.82e-3	0.85	1.54e-5	1.18e-3	0.93
OptiCo	<b>2.44e-6</b>	<b>4.18e-4</b>	<b>0.98</b>	<b>1.04e-6</b>	<b>8.88e-5</b>	<b>0.96</b>	<b>7.19e-6</b>	<b>8.59e-4</b>	<b>0.98</b>	<b>1.01e-5</b>	<b>1.24e-3</b>	<b>0.90</b>	<b>5.01e-6</b>	<b>6.38e-4</b>	<b>0.96</b>

Table 3. Ablation of the OP kernel across alternative diffraction formulations to assess formulation-flexible behavior.

Diffraction formulas	StdMetal (EPE)	StdContact (EPE)
w/o kernel	2.819	22.612
Helmholtz	0.207	0.303
Hopkins	0.125	0.461
Green’s	0.059	0.188
Fresnel	<b>0.044</b>	<b>0.079</b>

Table 4. Ablation of the strict OP kernel variant  $W_{\text{eff}}=\lambda_{\alpha}Q$ , which more faithfully reflects high-fidelity diffraction models.

Diffraction formulas	StdMetal (EPE)	StdContact (EPE)
w/o kernel	2.819	22.612
Helmholtz	0.786	5.497
Hopkins	<b>0.044</b>	<b>0.212</b>
Green’s	0.059	0.309
Fresnel	0.173	1.061

Table 5. An ablation study for each component in OptiCo block: OP complex convolution (OP), Multiply Constants (MC), ComplexConv1D (CC).

Ablation study			StdMetal (EPE)	StdContact (EPE)
OP	MC	CC	(EPE)	(EPE)
-	-	-	2.819	22.612
-	-	✓	0.657	7.491
✓	-	-	1.561	6.321
✓	✓	-	0.188	1.273
✓	✓	✓	<b>0.044</b>	<b>0.079</b>

**Additional experiments.** Table 2 presents the performance evaluation for the lithography simulation. The evaluation metric MSE<sub>resist</sub> and MSE<sub>aerial</sub> represent the MSE losses for the resist and aerial images. Consistent with our main findings, OptiCo outperforms all competing methods across subtasks, underscoring the robustness of OptiCo across diverse tasks. Furthermore, to evaluate the benefit of embedding optical physics, we performed a train **dataset-ratio ablation** in Appendix B.3, comparing physics-aware and purely data-driven methods. To investigate the compatibility of OptiCo, we conducted experiments with **different backbone networks**, such as DOINN and CFNO in Appendix B.2 and **a low resolution setting** in Appendix B.4, showing that OptiCo consistently adapts well across architectures and different resolution scales.

### 4.3. Ablation Study

**Alternative diffraction kernel.** The OP kernel is designed to embed diffraction principles into the network in a formulation-flexible manner. To examine this, we replaced the default Fresnel kernel with several alternative diffraction formulations. In Table 3, the OP kernel remains effective across these variants relative to the no-kernel baseline. Moreover, the simpler Green’s and Fresnel kernels achieve the best performance, indicating that lightweight formulations integrate more naturally with neural architectures.

**Strict OP kernel variant.** In Table 4, we examine a stricter effective kernel variant in which the OP kernel is scaled only by a scalar weight. This setting amplifies the importance of the diffraction model’s physical fidelity: the Hopkins TCC kernel achieves the highest performance, consistent with its role as the widely used standard

in computational lithography and indicating that our framework aligns well with high-fidelity physics models. While the paraxial Fresnel approximation experiences a noticeable drop, the non-paraxial Green’s formulation, which has better numerical aperture (NA) suitability, remains robust.

**Advantage of modeling in a complex domain.** In Table 5, we assess the contribution of each component in the OptiCo block. Since the phase is a crucial component of light, modeling in the complex domain is necessary. To validate this, we ablated OP complex convolution (OP), applying only ComplexConv1D (CC) in OptiCo. Even this simple modification leads to a notable performance gain, confirming the effectiveness of complex-domain modeling.

**Completeness of phase modulation.** In the OptiCo block, multiplying constants (MC) to OP are essential for preserving the complete diffraction formulation. When MC are ablated from OP, we observe a substantial performance drop, highlighting the importance of incorporating all formulation components. Finally, adding CC to OP completes the phase modulation and further enhances performance, underscoring the benefit of a faithful architectural design.

**Comparison with FNO.** While OptiCo explicitly incorporates diffraction principles, the Fourier Neural Operator (FNO) serves as a strong baseline, as its Fourier modeling can implicitly capture diffraction principles. To examine this distinction, we replaced OptiCo block with an FNO and compared the performance. As shown in Figure 3, substituting FNO yields a clear performance gain, consistent with our earlier finding on the importance of complex-domain modeling. Nevertheless, OptiCo achieves superior performance by explicitly embedding diffraction principles. Additional evaluation metrics are provided in Appendix B.1.

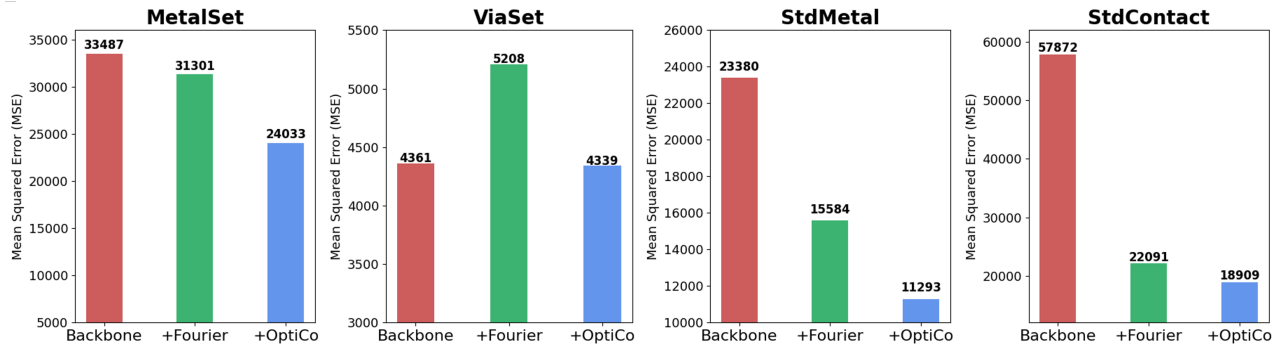


Figure 3. Comparative performance of the backbone network with the addition of Fourier module (FNO) and our OptiCo.

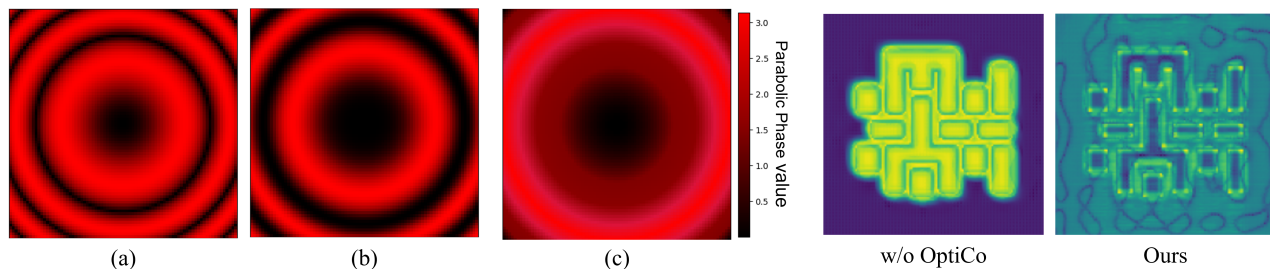


Figure 4. Visualization of the OP kernel across dynamic wavefront scales.

Figure 5. Intermediate feature visualization.

**Kernel size ablation.** A kernel that is too small results in a limited receptive field, making it insufficient to capture a diffraction, while a large kernel introduces complexity. In Appendix B.5, performance initially improves as the kernel size increases, but degrades when it becomes too large. We attribute this to the quadratic nature of a kernel ( $x^2 + y^2$ ), as steep phase variations hinder training. Nevertheless, OptiCo consistently outperforms strong baselines (ILILT).

#### 4.4. Diffraction-Inspired Representations

**Visualization of the OP kernel.** To provide an intuitive understanding of how the OP kernel encodes diffraction principles, we visualize its spatial structure across different wavefront scales. As illustrated in Figure 4, the images represent distinct configurations of the OP kernel,  $\exp\left(\frac{jk}{2z}[x^2 + y^2]\right)$ , each illustrating its dynamic adaptation to varying wavefront parameters. The accompanying color map on the right quantitatively represents parabolic phase values, where darker shades correspond to lower phases and brighter reds indicate higher accumulations. A key observation across all images is the presence of concentric rings with alternating intensity—an intrinsic characteristic of the parabolic phase term in diffraction. These rings indicate phase variations that reflect the OP kernel’s capacity to accurately model light propagation. Comparing images (a) through (c), we observe dynamic changes in the spacing, sharpness, and intensity of rings. Specifically, in (a), the rings are tightly packed, indicating high-frequency phase variations, while in (b) and (c), the rings are more widely spaced, reflecting a lower-frequency phase structure.

**Visualization of intermediate feature.** To further examine how the OP kernel influences internal representations, we provide intermediate feature visualizations comparing models with and without OptiCo in Figure 5. These results show that OptiCo focuses on diffraction-sensitive regions—particularly pattern corners and surrounding sub-resolution assist feature (SRAF) areas—highlighting OptiCo’s ability to capture optical principles. In summary, these visual representations underscore the OP kernel’s capacity to accurately model the complex optical lithography. Additional examples are provided in Appendix D.2.

**Visualization of diffraction-inspired mask patterns.** We compared the mask patterns optimized by OptiCo with those from other competing methods in Figure 6. Notably, OptiCo produces mask patterns featuring more distinct outer ring structures than its competitors. These diffraction-inspired ring structures arise from integrating light diffraction principles through the OP kernel, as shown in Figure 4. In particular, this phenomenon is especially pronounced when comparing ViaSet with its OOD test set, StdContact. OptiCo produces highly structured mask patterns in both datasets, clearly guided by the OP kernel’s explicit modeling of light diffraction. In contrast, DOINN, which is an FNO-based framework, forms complex diffraction patterns on the ViaSet but fails to replicate them on the StdContact (OOD test set). This discrepancy underscores the advantage of explicitly incorporating optical diffraction principles into semiconductor lithography through the use of our OP kernel. Additional visualizations for all methods and other subtasks are provided in Appendix D.1.

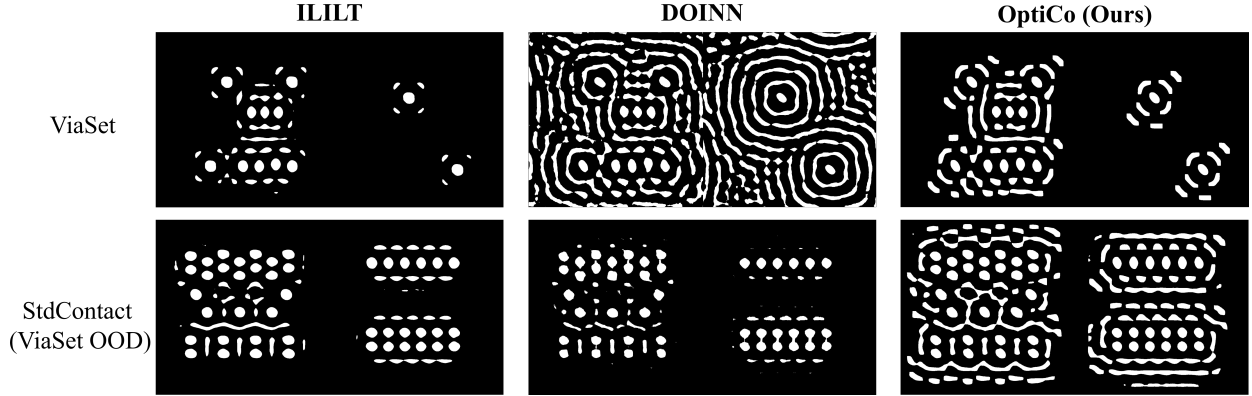


Figure 6. Visualization of optimized mask patterns. OptiCo produces clearer diffraction-inspired rings, especially on StdContact, highlighting the benefits of explicit diffraction modeling. In contrast, the Fourier-based DOINN shows complex rings on ViaSet but fails to preserve them on StdContact.

#### 4.5. Manufacturability Improvement with TV Loss

**Artifact suppression.** In mask optimization, *grainy artifacts* refer to small, high-frequency variations or speckles present in the mask pattern. While these artifacts may appear insignificant, they present substantial challenges in practical manufacturing, as they can lead to unpredictable deviations in the final pattern on the wafer. As shown in Figure 7, we compare the grainy artifacts in optimized mask patterns with and without the total variation (TV) loss. The model without TV loss produces noticeable grainy artifacts (highlighted in red), which pose practical challenges in mask manufacturing. In contrast, integrating TV loss into our model largely suppresses these grainy artifacts, resulting in smoother boundaries and more coherent shapes. This highlights the advantage of TV loss in enhancing the manufacturability of the masks.

**Simplified mask and post-processing robustness.** We also assessed manufacturability by comparing shot counts, which reflect the complexity of the layout geometry. As in Table 6, incorporating TV loss leads to a slight degradation in EPE but a substantial improvement in shot count, reducing it to less than half of that without TV loss. For further analysis, we applied morphological post-processing to the mask images [52], a commonly used technique that suppresses artifacts and smooth boundaries, though it inevitably degrades performance. Intuitively, if the mask is clean, the performance drop after post-processing should remain small. Indeed, masks trained with TV loss exhibit only minor degradation in EPE and negligible changes in shot count, confirming that TV loss promotes cleaner masks.

## 5. Conclusion

In this paper, we propose OptiCo, a novel physics-inspired convolutional neural network that integrates diffraction principles directly into its architecture to enhance computational lithography. Unlike conventional approaches that

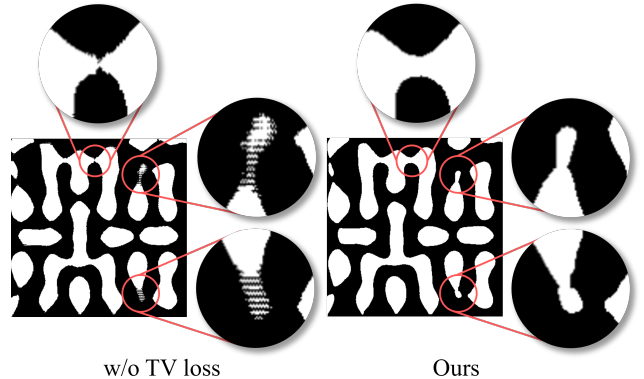


Figure 7. Comparison of mask patterns with and without the total variation (TV) loss, highlighting reduced grainy artifacts and smoother boundaries with coherent shapes in red circle.

Table 6. TV loss produces cleaner masks, leading to minimal EPE degradation and stable shot count after post-processing (PP).

Loss function	EPE ( $\downarrow$ )	Shot ( $\downarrow$ )
$\mathcal{L}_{\text{mse}}$	1.4	1523
$\mathcal{L}_{\text{mse}} + \text{PP}$	2.0 ( $\Delta+0.6$ )	1451 ( $\Delta-72$ )
$\mathcal{L}_{\text{mse}} + \mathcal{L}_{\text{tv}}$	1.6	715
$\mathcal{L}_{\text{mse}} + \mathcal{L}_{\text{tv}} + \text{PP}$	1.8 ( $\Delta+0.2$ )	712 ( $\Delta-3$ )

rely solely on data-driven methods or Fourier-based frameworks, OptiCo introduces a physics-inspired OP kernel to explicitly model the near-field diffraction behavior of light. Empirical evaluations on well-established lithography benchmarks demonstrate that our OptiCo consistently outperforms existing state-of-the-art methods in both lithography simulation and mask optimization tasks. Moreover, our in-depth analysis underscores the crucial role of the OP kernel in explicitly capturing the parabolic phase variations inherent to diffraction principles.

## Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT): the Artificial Intelligence Graduate School Program at Korea University (No. RS-2019-II190079), the Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration (No. 2022-0-00871), and the Artificial Intelligence Research Hub Project (No. RS-2024-00457882). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00212498).

## References

- [1] Vivek Bakshi. *EUV lithography*. SPIE press, 2009. 1
- [2] Shayak Banerjee, Zhuo Li, and Sani R Nassif. Iccad-2013 cad contest in mask optimization and benchmark suite. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 271–274. IEEE, 2013. 1
- [3] V Ye Banine, KN Koshelev, and GHPM Swinkels. Physical processes in euv sources for microlithography. *Journal of Physics D: Applied Physics*, 44(25):253001, 2011. 1
- [4] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013. 2
- [5] Kyle Braam, Kosta Selinidis, Wolfgang Hoppe, Hongyuan Cai, and Guangming Xiao. Euv mask synthesis with rigorous ilt for process window improvement. In *Design-Process-Technology Co-optimization for Manufacturability XIII*, pages 175–183. SPIE, 2019. 1
- [6] Charles L Byrne. *Iterative optimization in inverse problems*. CRC Press, 2014. 23
- [7] Guojin Chen, Wanli Chen, Yuzhe Ma, Haoyu Yang, and Bei Yu. Damo: Deep agile mask optimization for full chip scale. In *Proceedings of the 39th International Conference on Computer-Aided Design*, pages 1–9, 2020. 1, 3, 5, 22
- [8] Guojin Chen, Zehua Pei, Haoyu Yang, Yuzhe Ma, Bei Yu, and Martin Wong. Physics-informed optical kernel regression using complex-valued neural fields. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2023. 3, 5
- [9] Guojin Chen, Zixiao Wang, Bei Yu, David Z Pan, and Martin DF Wong. Ultrafast source mask optimization via conditional discrete diffusion. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(7): 2140–2150, 2024. 13
- [10] Nick Cobb. Sum of coherent systems decomposition by svd. *Berkeley CA*, 94720:1–7, 1995. 3
- [11] Frederick H Dill. Optical lithography. *IEEE transactions on electron devices*, 22(7):440–444, 1975. 2
- [12] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 22
- [13] Andreas Erdmann, Tim Fühner, Feng Shao, and Peter Evanschitzky. Lithography simulation: modeling techniques and selected applications. *Modeling Aspects in Optical Metrology II*, 7390:13–29, 2009. 2
- [14] Jhih-Rong Gao, Xiaoqing Xu, Bei Yu, and David Z Pan. Mosaic: Mask optimizing solution with process window aware inverse correction. In *Proceedings of the 51st Annual Design Automation Conference*, pages 1–6, 2014. 1
- [15] Iason Giannopoulos, Iacopo Mochi, Michaela Vockenhuber, Yasin Ekinci, and Dimitrios Kazazis. Extreme ultraviolet lithography reaches 5 nm resolution. *Nanoscale*, 16(33): 15533–15543, 2024. 1
- [16] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 23
- [17] Kramer Harrison. Optiland, 2025. Version 0.2.3. Available at <https://github.com/HarrisonKramer/optiland>. 23
- [18] Hongquan He, Zhen Wang, Jingya Wang, Tao Wu, Xuming He, Bei Yu, Jingyi Yu, and Hao Geng. Lithosim: A large, holistic lithography simulation benchmark for ai-driven semiconductor manufacturing. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. 20
- [19] Eugene Hecht. *Optics*. Pearson Education India, 2012. 2
- [20] Harold Horace Hopkins. The concept of partial coherence in optics. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 208(1093):263–277, 1951. 1, 2
- [21] Won Joon Jeong, Nicholas V Babusis, and Alexey Shashurin. Measurements of electron number density in high-energy nanosecond discharges by laser interferometry. In *AIAA SCITECH 2025 Forum*, page 0989, 2025. 23
- [22] Bentian Jiang, Lixin Liu, Yuzhe Ma, Hang Zhang, Bei Yu, and Evangeline FY Young. Neural-ilt: Migrating ilt to neural networks for mask printability and complexity co-optimization. In *Proceedings of the 39th International Conference on Computer-Aided Design*, pages 1–9, 2020. 21
- [23] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020. 1, 3, 22
- [24] Xiaoxiao Liang, Yikang Ouyang, Haoyu Yang, Bei Yu, and Yuzhe Ma. RI-opc: Mask optimization with deep reinforcement learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023. 21
- [25] Xiaoxiao Liang, Haoyu Yang, Kang Liu, Bei Yu, and Yuzhe Ma. Camo: Correlation-aware mask optimization with modulated reinforcement learning. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pages 1–6, 2024. 21
- [26] Mingjie Liu, Haoyu Yang, Brucec Khailany, and Haoxing Ren. An adversarial active sampling-based data augmentation framework for ai-assisted lithography modeling. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pages 1–9. IEEE, 2023. 23
- [27] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the

- 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 23
- [28] Pingchuan Ma, Zhengqi Gao, Meng Zhang, Haoyu Yang, Mark Ren, Rena Huang, Duane S Boning, and Jiaqi Gu. Maps: Multi-fidelity ai-augmented photonic simulation and inverse design infrastructure. *arXiv preprint arXiv:2503.01046*, 2025. 23
- [29] Xu Ma and Gonzalo R Arce. *Computational lithography*. John Wiley & Sons, 2011. 1
- [30] Chris A Mack. Thirty years of lithography simulation. In *Optical Microlithography XVIII*, pages 1–12. SPIE, 2005. 1
- [31] Tetsuaki Matsunawa, Bei Yu, and David Z Pan. Optical proximity correction with hierarchical bayes model. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 15(2): 021009–021009, 2016. 21
- [32] Wayne M Moreau. *Semiconductor lithography: principles, practices, and materials*. Springer Science & Business Media, 2012. 1, 2
- [33] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019. 20
- [34] Chao Qian, Yang Yu, and Zhi-Hua Zhou. Subset selection by pareto optimization. *Advances in neural information processing systems*, 28, 2015. 23
- [35] John Randall, Kurt G Ronse, Thomas Marschner, Anne-Marie Goethals, and Monique Ercken. Variable-threshold resist models for lithography simulation. In *Optical Microlithography XII*, pages 176–182. SPIE, 1999. 2
- [36] Chengping Rao, Pu Ren, Yang Liu, and Hao Sun. Discovering nonlinear pdes from scarce data with physics-encoded learning. *arXiv preprint arXiv:2201.12354*, 2022. 15
- [37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 22
- [38] Dong-Hee Shin, Young-Han Son, Jun-Mo Kim, Hee-Jun Ahn, Jun-Ho Seo, Chang-Hoon Ji, Ji-Wung Han, Byung-Jun Lee, Dong-Ok Won, and Tae-Eui Kam. MARS: Multiagent reinforcement learning for spatial–spectral and temporal feature selection in EEG-based BCI. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2024. 21
- [39] Dong-Hee Shin, Young-Han Son, Deok-Joong Lee, Ji-Wung Han, and Tae-Eui Kam. Dynamic many-objective molecular optimization: Unfolding complexity with objective decomposition and progressive optimization. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 6026–6034, 2024. 23
- [40] Siemens. Calibre. <https://eda.sw.siemens.com/en-US/ic/calibre-design/>, 1. 1
- [41] Arnold Sommerfeld. *Partial differential equations in physics*. Academic press, 1949. 2, 3
- [42] Young-Han Son, Dong-Hee Shin, and Tae-Eui Kam. FT-MMR: Fusion transformer for integrating multiple molecular representations. *IEEE Journal of Biomedical and Health Informatics*, 2024. 23
- [43] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 22
- [44] Harald Volkman. Ernst abbe and his work. *Applied optics*, 5(11):1720–1731, 1966. 2
- [45] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 22
- [46] Yuki Watanabe, Taiki Kimura, Tetsuaki Matsunawa, and Shigeki Nojima. Accurate lithography simulation model based on convolutional neural networks. In *Optical Microlithography XXX*, pages 137–145. SPIE, 2017. 1
- [47] Banqiu Wu and Ajay Kumar. Extreme ultraviolet lithography: A review. *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures Processing, Measurement, and Phenomena*, 25(6):1743–1761, 2007. 1
- [48] Xiaoqing Xu, Tetsuaki Matsunawa, Shigeki Nojima, Chikaaki Kodama, Toshiya Kotani, and David Z Pan. A machine learning based framework for sub-resolution assist feature generation. In *Proceedings of the 2016 on International Symposium on Physical Design*, pages 161–168, 2016. 21
- [49] Yunpeng Xu, Zihan Zang, Haoqiang Wang, Yanjun Han, Hongtao Li, Yi Luo, Lai Wang, Changzheng Sun, Bing Xiong, Zhibiao Hao, et al. Fabrication-integrated design for diffractive optical elements. *Optica*, 12(2):228–238, 2025. 2
- [50] Haoyu Yang and Haoxing Ren. Enabling scalable ai computational lithography with physics-inspired models. In *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, pages 715–720, 2023. 1, 3, 5, 22
- [51] Haoyu Yang and Haoxing Ren. Iilt: Implicit learning of inverse lithography technologies. In *International conference on machine learning*. PMLR, 2024. 3, 5
- [52] Haoyu Yang and Haoxing Ren. Gpu-accelerated inverse lithography towards high quality curvy mask generation. In *Proceedings of the 2025 International Symposium on Physical Design*, pages 42–50, 2025. 1, 8, 23
- [53] Haoyu Yang, Shuhe Li, Yuzhe Ma, Bei Yu, and Evangeline FY Young. Gan-opc: Mask optimization with lithography-guided generative adversarial nets. In *Proceedings of the 55th Annual Design Automation Conference*, pages 1–6, 2018. 1, 3, 21
- [54] Haoyu Yang, Zongyi Li, Kumara Sastry, Saumyadip Mukhopadhyay, Mark Kilgard, Anima Anandkumar, Brucec Khailany, Vivek Singh, and Haoxing Ren. Generic lithography modeling with dual-band optics-inspired neural networks. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pages 973–978, 2022. 1, 3, 5, 13, 22
- [55] Haoyu Yang, Yibo Lin, and Bei Yu. Machine learning for mask synthesis and verification. In *Machine Learning Applications in Electronic Design Automation*, pages 425–470. Springer, 2022. 1, 21
- [56] Haoyu Yang, Anthony Agnesina, and Haoxing Ren. Optimizing predictive ai in physical design flows with mini pixel batch gradient descent. In *Proceedings of the 2024 ACM/IEEE International Symposium on Machine Learning for CAD*, pages 1–7, 2024. 23

- [57] Wei Ye, Mohamed Baker Alawieh, Yibo Lin, and David Z Pan. Lithogan: End-to-end lithography modeling with generative adversarial networks. In *Proceedings of the 56th Annual Design Automation Conference 2019*, pages 1–6, 2019. [3](#), [21](#), [22](#)
- [58] Wei Ye, Mohamed Baker Alawieh, Yuki Watanabe, Shigeki Nojima, Yibo Lin, and David Z Pan. Tempo: Fast mask topography effect modeling with deep learning. In *Proceedings of the 2020 International Symposium on Physical Design*, pages 127–134, 2020. [1](#), [22](#)
- [59] Shuo Yin, Su Zheng, Ziyang Yu, and Bei Yu. Torchlitho 2.0: Differentiable lithography simulation engine for large-scale layouts. In *2025 IEEE 16th International Conference on ASIC (ASICON)*, pages 1–4. IEEE, 2025. [20](#)
- [60] Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? *arXiv preprint arXiv:2405.07992*, 2024. [23](#)
- [61] Weihao Yu, Pan Zhou, Shuicheng Yan, and Xinchao Wang. Inceptionnext: When inception meets convnext. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5672–5683, 2024. [4](#)
- [62] Ziyang Yu, Guojin Chen, Yuzhe Ma, and Bei Yu. A gpu-enabled level-set method for mask optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(2):594–605, 2022. [1](#)
- [63] Chuanjun Zheng, Daming Shi, and Wentian Shi. Adaptive unfolding total variation network for low-light image enhancement. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4439–4448, 2021. [4](#)
- [64] Cheng Zheng, Guangyuan Zhao, and Peter So. Close the design-to-manufacturing gap in computational optics with a ‘real2sim’ learned two-photon neural lithography simulator. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–9, 2023. [1](#)
- [65] Su Zheng, Haoyu Yang, Binwu Zhu, Bei Yu, and Martin Wong. Lithobench: Benchmarking ai computational lithography for semiconductor manufacturing. *Advances in Neural Information Processing Systems*, 36, 2023. [5](#), [19](#), [22](#)
- [66] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer, 2018. [1](#), [5](#), [22](#)
- [67] Binwu Zhu, Su Zheng, Ziyang Yu, Guojin Chen, Yuzhe Ma, Fan Yang, Bei Yu, and Martin DF Wong. L2o-ilt: Learning to optimize inverse lithography techniques. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023. [3](#), [21](#)

## A. Derivation of OP Kernel from Diffraction Integrals

In scalar diffraction theory, the unknown field  $U$  is obtained by representing it via the Green's function and a boundary integral. The Rayleigh–Sommerfeld (RS) diffraction integral serves as the standard starting point for exact diffraction calculations, and controlled approximations yield simpler and computationally convenient expressions.

**Helmholtz–Kirchhoff form.** Derived from the Helmholtz–Kirchhoff integral theorem with Kirchhoff boundary conditions, it expresses the field as an aperture integral and retains the obliquity factor  $z/r$  (angular dependence), it also keeps the near-field correction  $1 + j/kr$ . Accordingly, it is the RS representation with the fewest approximations. Let  $r = \sqrt{(x' - x)^2 + (y' - y)^2 + z^2}$ ,

$$\begin{aligned}
 U(x', y') &= \frac{1}{j\lambda} \iint_{-\infty}^{\infty} U(x, y) \frac{e^{jk r}}{r} \frac{z}{r} \left(1 + \frac{j}{kr}\right) dx dy \\
 U(x', y') &= \frac{z}{j\lambda} \iint_{-\infty}^{\infty} U(x, y) \frac{e^{jk r}}{r^2} \left(1 + \frac{j}{kr}\right) dx dy \\
 U(x', y') &= \frac{z}{j\lambda} \cdot U(x, y) * \left[ \frac{\exp\left(jk\sqrt{x^2 + y^2 + z^2}\right)}{x^2 + y^2 + z^2} \left(1 + \frac{j}{k\sqrt{x^2 + y^2 + z^2}}\right) \right] \\
 h(x, y) &= \frac{\exp\left(jk\sqrt{x^2 + y^2 + z^2}\right)}{x^2 + y^2 + z^2} \left(1 + \frac{j}{k\sqrt{x^2 + y^2 + z^2}}\right)
 \end{aligned} \tag{16}$$

As you can see from the form of the equation, the Helmholtz–Kirchhoff kernel is relatively complex, which makes it less practical when used as a convolutional kernel in a neural network. This observation is consistent with our ablation study in Table 3, where the Helmholtz-based kernel shows significantly degraded performance compared to the alternatives.

**Green's function form.** The Green's-function form is the canonical Rayleigh–Sommerfeld diffraction integral. In our convention, it is derived directly from the free-space Green's function for the scalar Helmholtz equation, without explicitly invoking the Kirchhoff boundary condition on the opaque screen. Modeling the aperture as a planar boundary distribution and enforcing the Sommerfeld radiation condition omits the near-field correction  $(1 + j/kr)$ , resulting in a simplified but widely used expression. This form is frequently adopted in practice due to its reduced complexity while still capturing the essential diffraction behavior.

$$\begin{aligned}
 U(x', y') &= \frac{1}{j\lambda} \iint_{-\infty}^{\infty} U(x, y) \frac{e^{jk r}}{r} \frac{z}{r} dx dy \\
 U(x', y') &= \frac{z}{j\lambda} \iint_{-\infty}^{\infty} U(x, y) \frac{e^{jk r}}{r^2} dx dy \\
 U(x', y') &= \frac{z}{j\lambda} \cdot U(x, y) * \left[ \frac{\exp\left(jk\sqrt{x^2 + y^2 + z^2}\right)}{x^2 + y^2 + z^2} \right] \\
 h(x, y) &= \frac{\exp\left(jk\sqrt{x^2 + y^2 + z^2}\right)}{x^2 + y^2 + z^2}
 \end{aligned} \tag{17}$$

This form is widely employed due to its reduced complexity while still providing an accurate description of diffraction. As shown in Tables 3 and 4, the Green's function form consistently performs well, underscoring its practical value and motivating further study.

**Fresnel approximation form.** The Fresnel approximation employed in the main text is one of two key approximations to the Rayleigh-Sommerfeld integral under near-field conditions. The primary objective of the Fresnel approximation is to approximate the distance  $r$ , which assumes that distance  $z$  is sufficiently large compared to the aperture size of the mask. Under this condition, the oblique factor  $z/r$  can be approximated to 1, as known as paraxial approximation.

$$\frac{z}{r} \approx 1 \tag{18}$$

Here,  $r$  is approximated via a binomial expansion, known as the Fresnel approximation.

$$r \approx z + \frac{(x - x')^2 + (y - y')^2}{2z}. \quad (19)$$

Starting with Green's function form, we substitute  $r$  and oblique factor  $z/r$  from the equation.

$$\begin{aligned} U(x', y') &= \frac{1}{j\lambda} \iint_{-\infty}^{\infty} U(x, y) \frac{e^{jkr}}{r} \frac{z}{r} dx dy \\ U(x', y') &= \frac{e^{jkz}}{j\lambda z} \iint_{-\infty}^{\infty} U(x, y) e^{\frac{jk}{2z}[(x'-x)^2 + (y'-y)^2]} dx dy. \\ U(x', y') &= \frac{e^{jkz}}{j\lambda z} \left[ U(x, y) * \exp\left(\frac{jk}{2z} [x^2 + y^2]\right) \right], \\ h(x, y) &= \exp\left(\frac{jk}{2z} [x^2 + y^2]\right) \end{aligned} \quad (20)$$

We adopt this kernel as our primary optical-phase (OP) kernel. Its simplicity enables accurate and efficient diffraction modeling.

**Hopkins transmission cross-coefficient (TCC) model.** The Hopkins model is a widely used framework in computational lithography to simulate the optical imaging process. The Hopkins model calculates the aerial image  $I(x', y')$ , representing the light intensity at the wafer plane, as follows:

$$\begin{aligned} I(x', y') &= \iiint_{-\infty}^{\infty} \mathcal{F}(M)(f', g') \mathcal{F}(M)^*(f'', g'; ; ) \mathcal{T}(f', g'; f'', g'') \\ &\quad \exp(-j2\pi[(f' - f'')x' + (g' - g'')y']) df' dg' df'' dg'', \end{aligned} \quad (21)$$

where  $H$  denotes the optical transfer function (OTF) of the projector system;  $\mathcal{F}(M)$  represents the mask pattern  $M$  transformed by fast Fourier transform (FFT);  $M^*$  is the complex conjugate of the mask;  $(x', y')$  denotes the spatial coordinates on the image plane;  $(f, g)$ ,  $(f', g')$  and  $(f'', g'')$  represent the normalized frequency-domain coordinates.

The most important term is  $\mathcal{T}$  the *transmission cross coefficient* (TCC),  $\mathcal{T}$ , which encodes how the source distribution  $J$  couples with the pupil transfer function  $H$  to form image intensity.

$$\mathcal{T}(f', g'; f'', g'') = \iint J(f, g) H(f + f', g + g') H^*(f + f'', g + g'') df dg. \quad (22)$$

To approximate the TCC efficiently, we use the SVD-based *sum of coherent sources* (SOCS) decomposition:

$$\mathcal{T}(f', g'; f'', g'') = \sum_{i=1}^r \alpha_i h_i(f', g') h_i^*(f'', g''), \quad (23)$$

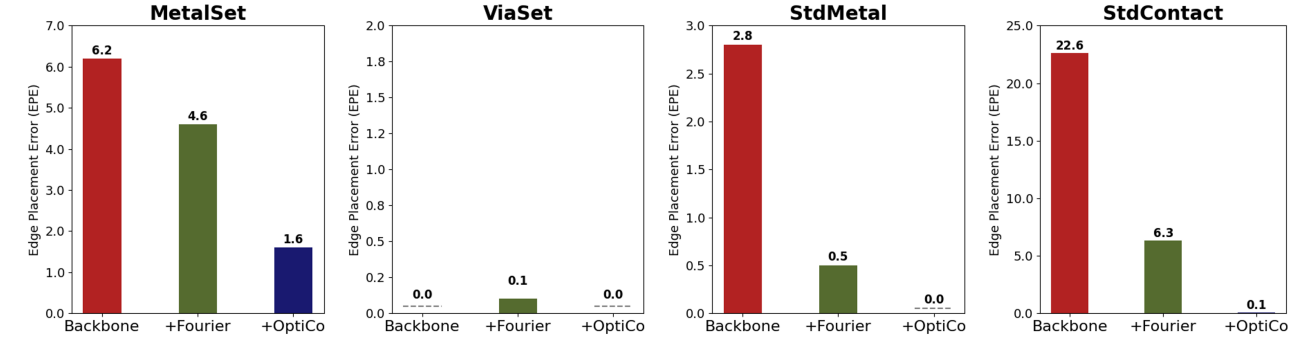
where  $\alpha_i$  and  $h_i$  are the  $i$ -th eigenvalue and eigenfunction of the TCC.

With convolution theorem, the Hopkins intensity reduces to a sum of coherent images:

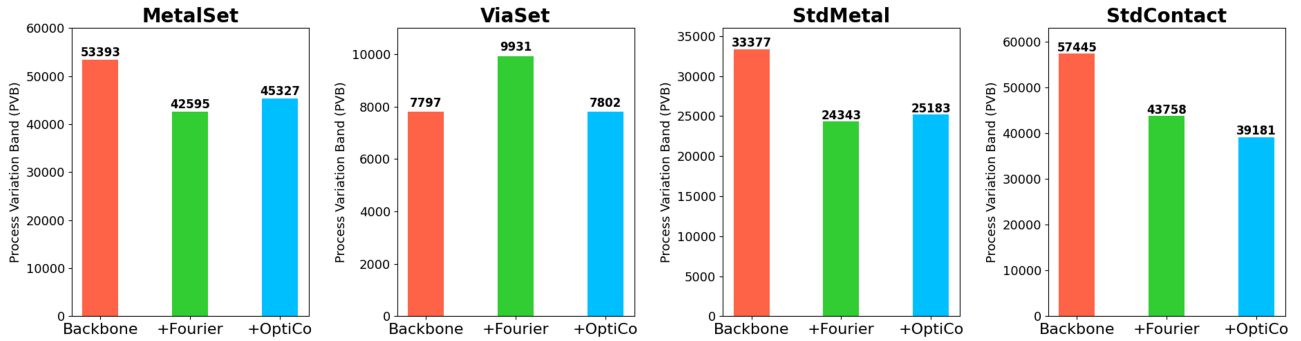
$$\begin{aligned} I(x, y) &= \sum_{i=1} \alpha_i | (h_i * M)(x, y) |^2 \\ &= \sum_{i=1} \alpha_i \left| \mathcal{F}^{-1}(\mathcal{F}(h_i) \odot \mathcal{F}(M))(x, y) \right|^2, \end{aligned} \quad (24)$$

where  $*$  denotes spatial convolution and  $\odot$  is pointwise multiplication.

This description of the Hopkins model closely follows [9, 54].



(a) Comparative Experiments for Edge Placement Error (EPE)



(b) Comparative Experiments for Process Variation Band (PVB)

Figure 8. Comparative performance of the backbone network with the addition of the Fourier module (FNO) and our OptiCo module that contains the OP kernel. The results highlight the superior performance of our OptiCo module with the OP kernel in reducing EPE and PVB.

## B. Additional Experiments and Results

### B.1. Additional Evaluation Metrics for Fourier Module Comparison

In the main manuscript, we conducted comparative experiments across three configurations: 1) a vanilla backbone network, 2) a backbone network with a Fourier module, and 3) a backbone network integrated with our OptiCo module, which incorporates the OP kernel, evaluated using the MSE metric. In this section, we extend the comparative experiments to other evaluation metrics, such as Edge Placement Error (EPE) and Process Variation Band (PVB).

As illustrated in Figure 8 (a), our OptiCo framework demonstrated exceptional performance in the EPE metric as well. This was particularly evident on OOD datasets such as StdMetal and StdContact, where OptiCo exhibited remarkable improvements. These results highlight the effectiveness of integrating physics-inspired Fresnel diffraction principles into the neural network architecture, allowing OptiCo to accurately model light propagation and diffraction effects.

Similarly, Figure 8 (b) presents the results for the PVB metric, where OptiCo consistently delivered superior performance, particularly on OOD datasets. While the Fourier-based module outperformed the vanilla backbone network—highlighting the advantages of operating within the Fourier space to implicitly model global light propagation patterns—our OptiCo module explicitly modeled these patterns through the OP kernel. This explicit modeling approach enabled OptiCo to achieve superior performance, clearly demonstrating the advantages of explicitly capturing the phase variations inherent in diffraction over implicit modeling approaches.

These additional results for comparative performance collectively validate the robustness and generalization capabilities of our OptiCo framework across diverse evaluation metrics and challenging OOD datasets.

## B.2. OptiCo with Different Backbone Networks

Table 7. Performance comparison of OptiCo framework integrated with various backbone networks such as DOINN and CFNO.

Subtask	Method	MSE( $\downarrow$ )	PVB( $\downarrow$ )	EPE( $\downarrow$ )
MetalSet	DOINN	36409	<b>41929</b>	7.4
	+ OptiCo (ours)	<b>28322</b>	44846	<b>3.1</b>
	CFNO	47814	46131	12.5
	+ OptiCo (ours)	<b>38586</b>	<b>45743</b>	<b>7.5</b>
ViaSet	DOINN	4382	7836	<b>0.0</b>
	+ OptiCo (ours)	<b>4362</b>	<b>7819</b>	<b>0.0</b>
	CFNO	8949	9890	0.1
	+ OptiCo (ours)	<b>4406</b>	<b>8505</b>	<b>0.0</b>
StdMetal (MetalSet OOD)	DOINN	25913	25749	4.5
	+ OptiCo (ours)	<b>15589</b>	<b>25408</b>	<b>0.8</b>
	CFNO	26809	26814	4.2
	+ OptiCo (ours)	<b>22542</b>	<b>25328</b>	<b>2.5</b>
StdContact (ViaSet OOD)	DOINN	72058	<b>17968</b>	55.8
	+ OptiCo (ours)	<b>23213</b>	40113	<b>2.2</b>
	CFNO	70740	<b>17950</b>	55.1
	+ OptiCo (ours)	<b>23150</b>	42731	<b>1.7</b>
Average	DOINN	34691	<b>23370</b>	16.9
	+ OptiCo (ours)	<b>17871</b>	29546	<b>1.5</b>
	CFNO	38578	<b>25196</b>	18.0
	+ OptiCo (ours)	<b>22171</b>	30577	<b>2.9</b>

In our main experiments, we employed a MetaNeXt-based backbone neural network within the OptiCo framework to demonstrate its efficacy in lithography simulation and mask optimization tasks. To further validate the versatility and compatibility of OptiCo, we extended its integration to a range of other FNO-based backbone networks, specifically DOINN and CFNO, which also serve as competing methods in our study.

As detailed in Table 7, our OptiCo framework was systematically integrated with both DOINN and CFNO backbones. The results demonstrate that OptiCo seamlessly integrates with diverse backbone architectures and consistently enhances performance across most evaluation metrics. Notably, its incorporation with the DOINN and CFNO backbones yielded significant improvements on challenging OOD datasets such as StdMetal and StdContact.

These experiments highlight that OptiCo is not limited to a single backbone architecture but can be seamlessly applied to various backbone networks. The consistent performance enhancements observed across different backbones indicate that the benefits of incorporating the Optical Phase (OP) kernel are universally applicable, regardless of the underlying neural network architecture. This adaptability is particularly valuable in computational lithography, where different backbone networks may be preferred based on specific task requirements or computational constraints.

## B.3. Dataset Ablation

Physics-based inductive biases are known to substantially improve data efficiency [36]. To quantify this effect, we conducted a dataset ablation study varying the training dataset size (10%, 25%, 50%, and 100%). Specifically, we compared two physics-aware models—ILILT and our OptiCo model—with their purely data-driven counterparts—DAMO and a variant of OptiCo without the OP kernel. As shown in Table 8 and Figure 9, models incorporating physics priors achieve significantly lower EPE scores at all data regimes, demonstrating their superior efficiency under limited data.

Table 8. Average EPE comparisons from dataset ablation study (lower is better). Models with physics-based priors (ILILT and OptiCo) show significantly better data efficiency under limited data.

Training ratio	Without physics prior		With physics prior	
	DAMO	Ours w/o OP	ILILT	Ours
10%	73.2	45.8	7.3	6.4
25%	35.0	17.3	5.5	1.1
50%	12.9	11.9	4.5	0.8
100%	8.1	8.1	2.6	0.4

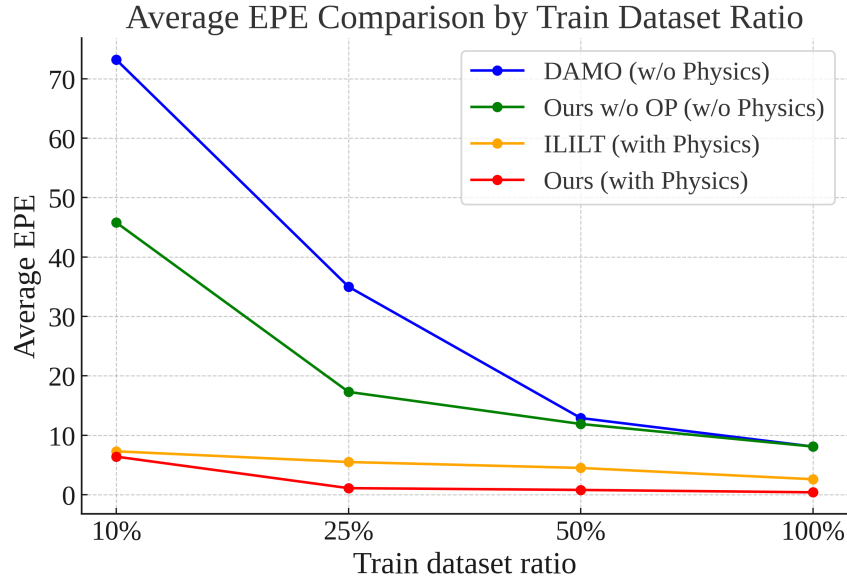


Figure 9. Ablation studies of training dataset ratio. We compare the performance of models with and without physics priors across ratios.

#### B.4. Resolution Reduction Analysis for Larger Layout

In practical semiconductor manufacturing, real layout patterns are often much larger than those used in our dataset, making direct optimization at full resolution computationally infeasible. To examine whether our method can generalize to larger-scale layouts under such hardware constraints, we simulate a low-resolution setting by downsampling  $1024 \times 1024$  mask layouts to  $256 \times 256$  using average pooling (a  $1/16$  reduction in pixel count). This setup effectively corresponds to evaluating our model on a larger physical layout, since reducing the resolution of a fixed-size image is equivalent to increasing the layout area density. This experiment therefore serves as an indirect but practical indicator of scalability—if the method remains stable at lower resolutions, it is likely to perform reliably even when extended to physically larger layouts.

Table 9. **Effect of resolution reduction on mask optimization performance.** Reducing the input resolution by  $16 \times$  still preserves strong performance, indicating robustness and scalability across layout sizes.

Method (Resolution)	Metal( $\downarrow$ )	Via( $\downarrow$ )	Metal OOD( $\downarrow$ )	Via OOD( $\downarrow$ )	Average EPE( $\downarrow$ )
DAMO ( $1024 \times 1024$ )	5.4	0.0	0.2	26.7	8.1
DOINN ( $1024 \times 1024$ )	7.4	0.0	4.5	55.8	16.9
CFNO ( $1024 \times 1024$ )	12.5	0.1	4.2	55.1	18.0
ILILT ( $1024 \times 1024$ )	3.2	0.0	0.1	7.1	2.6
<b>Ours (<math>256 \times 256</math>)</b>	<b>4.1</b>	<b>0.0</b>	<b>0.4</b>	<b>2.5</b>	<b>1.7</b>
<b>Ours (<math>1024 \times 1024</math>)</b>	<b>1.6</b>	<b>0.0</b>	<b>0.0</b>	<b>0.1</b>	<b>0.4</b>

Despite a  $16\times$  reduction in pixel count, our method maintains strong performance and remains comparable to the full-resolution baselines. This indicates that the optimization mechanism is robust to spatial scaling and that our model can be effectively applied to larger real-world layouts without significant degradation in quality.

### B.5. Kernel Size Ablation

Since OptiCo is a convolution-based framework, the OP kernel size is a critical factor in its performance. A kernel that is too small results in a limited receptive field, making it insufficient to capture diffraction. We conduct an ablation study on the OP kernel size in Table 3. Increasing the kernel size initially improves performance, but beyond a certain point, it begins to degrade, indicating that the performance does not improve monotonically. We attribute this trend to the quadratic nature of the OP,  $(x^2 + y^2)$ . When the kernel becomes too large, steep phase variations may disrupt effective learning. Nevertheless, OptiCo consistently outperforms the strong baseline ILILT across all tested kernel sizes, highlighting the robustness.

Table 10. Ablation on the kernel size.

Kernel size	StdMetal (EPE↓)	StdContact (EPE↓)
3	0.048	0.364
5	0.048	0.345
7	0.055	0.139
9	0.052	0.097
11	<b>0.044</b>	<b>0.079</b>
13	0.055	0.182
15	0.050	0.221
17	0.048	0.242
ILILT	0.111	7.055

## C. Pytorch-like Algorithm for OP Kernel Construction

For clarity, we provide a PyTorch-style algorithm for constructing the OP kernel used in our complex convolution layer in Section 3.2. This code snippet directly reflects the OP formulation for Fresnel approximation form, where the kernel encodes the parabolic phase term derived from the Fresnel diffraction. Each element of the constructed OP kernel corresponds to the complex exponential,  $Q(x, y) = \exp\left(\frac{jk}{2z}(x^2 + y^2)\right)$ .

```
def create_Q_kernel(k, l, kernel_size):
    N = kernel_size
    center = (N - 1)/2
    x = torch.arange(N) - center
    y = torch.arange(N) - center
    X, Y = torch.meshgrid(x, y, index='ij')

    exponent = 1j*k*(X**2 + Y**2)/(2*l)
    Q = torch.exp(exponent)
    return Q
```

Algorithm 1. PyTorch implementation of the OP kernel

## D. Additional Visualization of Mask Patterns and Intermediate Features

### D.1. Additional Sample Mask Pattern Visualizations

In our main manuscript, we presented visualizations of sample mask patterns for strong competing methods such as DOINN and ILILT. To complement the main manuscript, Figure 10 presents a more comprehensive set of visualizations to allow for a broader and more detailed comparison across other competing methods. These additional visualizations allow for a more thorough comparison and highlight the distinctive advantages of our approach.

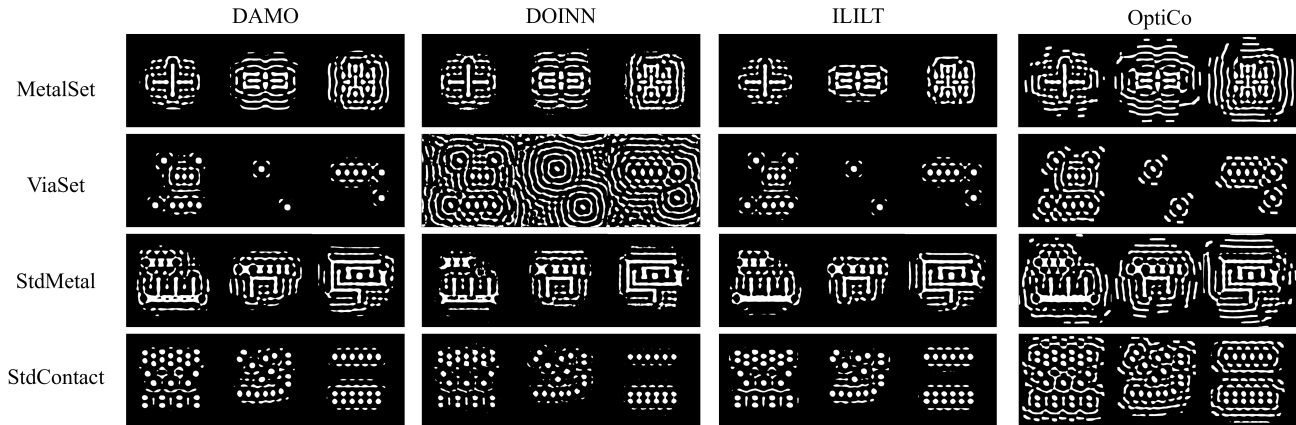


Figure 10. Visualization of additional optimized mask patterns across all competing methods.

As previously discussed, it is evident from the visualizations that our OptiCo consistently generates mask patterns with more distinct outer ring structures compared to other methods. This clarity in the outer rings underscores the effectiveness of integrating light diffraction principles through the *Optical Phase (OP) kernel* within our neural network architecture. The pronounced ring structures indicate accurate modeling of spatial phase variations, which are crucial for precisely capturing lithographic behavior in computational lithography. In contrast, other competing methods struggle to generate well-defined mask patterns, particularly on OOD datasets such as StdMetal and StdContact. This performance degradation can be attributed to their reliance on implicit modeling approaches, which lack the explicit phase-aware transformations provided by OptiCo.

## D.2. Intermediate Feature Visualization

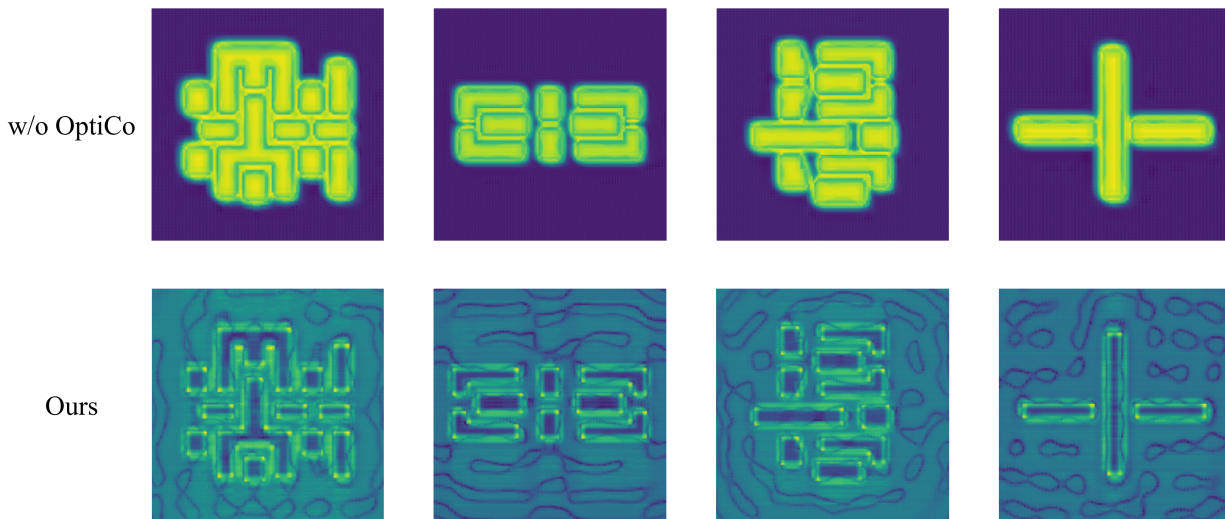


Figure 11. Visualization of intermediate features with and without the physics-informed OptiCo block. The intermediate features produced by OptiCo explicitly emphasize critical corner regions highlighted by bright features around them. In contrast, intermediate features from the baseline method focus only on the mask regions, neglecting the critical corners.

To better understand how the OP kernel influences OptiCo’s internal representations, we visualize intermediate feature maps from models with and without OptiCo in Figure 11. Pattern corners are key regions in lithography due to their sensitivity to light diffraction, making them important targets for effective modeling. We observe that OptiCo consistently emphasizes corner regions and nearby sub-resolution assist features (SRAFs, see Appendix F.1 for details) through strong activations.

In contrast, the baseline model without OptiCo tends to highlight only the main mask shapes, failing to distinguish critical corners from other regions and largely ignoring surrounding diffraction-sensitive areas. These results suggest that the OP kernel encourages the network to focus on phase-sensitive areas such as corners, thereby enhancing the model’s understanding of the optical lithography process.

## E. Experimental Details

### E.1. Details of Evaluation Metrics in Semiconductor Computational Lithography

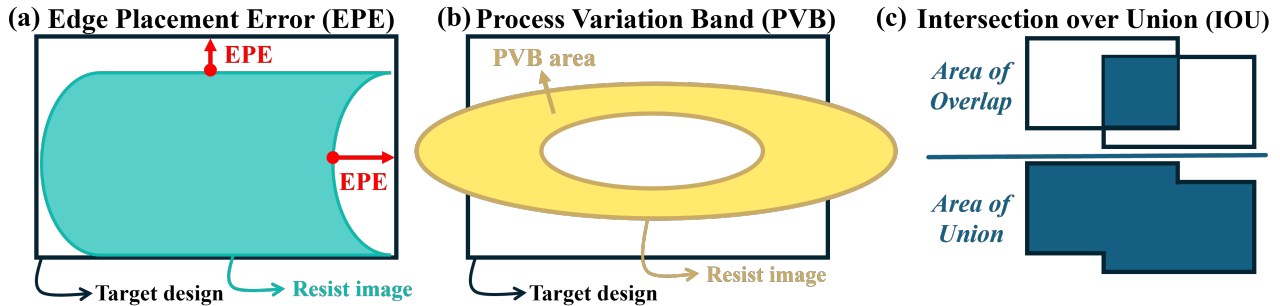


Figure 12. Illustrations of various evaluation metrics in computational lithography: (a) Edge Placement Error (EPE), which quantifies the deviation of edges from their expected positions; (b) Process Variation Band (PVB), showing the tolerance range of variations; and (c) Intersection over Union (IoU), measuring the overlap accuracy between predicted and reference regions.

In this subsection, we provide detailed explanations of the evaluation metrics employed in our study—Edge Placement Error (EPE), Process Variation Band (PVB), and Intersection over Union (IoU)—clarifying their definitions, significance, and roles within the context of computational lithography. As shown in Figure 12 (a), EPE quantifies the deviation between the intended edge positions of the target design on the mask and the actual edge positions observed in the resist image resulting from the computational lithography process. In other words, this metric is expressed as the distance between the predicted and target edge positions. In chip manufacturing, even minor deviations in edge placement can result in significant functional or performance degradation in the fabricated semiconductor devices. As a result, EPE serves as a vital indicator of whether a given mask design can be reliably reproduced with the required level of precision. On the other hand, PVB represents the range of possible edge positions that may arise due to variations in the manufacturing process. These variations stem from factors such as exposure dose, focus shifts, and resist chemistry fluctuations. As depicted in Figure 12 (b), PVB is typically visualized as a band encompassing the range of edge deviations under different process conditions, thereby offering insights into the robustness of a given mask design. A smaller PVB indicates a mask design that is less sensitive to process variations, which is essential for ensuring yield and performance consistency under diverse manufacturing conditions. Finally, IoU is a widely used evaluation metric in image segmentation tasks. In computational lithography, it is employed to assess the spatial overlap between predicted resist images and their corresponding true reference regions. Specifically, as illustrated in Figure 12 (c), IoU is defined as the ratio of the intersection area to the union area of the predicted and ground truth resist regions. This metric provides a holistic measure of the accuracy with which the predicted patterns replicate the intended design.

## E.2. Hyperparameters and Training Strategies

### E.2.1. Overall Training Procedure of Mask Optimization

Our method follows the standard training methodology provided by LithoBench [65] for both mask optimization and lithography simulation tasks. This approach has been uniformly applied to all models presented in our experiments to ensure fair comparison. For the mask optimization task, LithoBench includes a **pretraining stage**, since untrained mask optimization models tend to generate blank simulated resist images. We follow this procedure as it also significantly reduces computational overhead during training.

Briefly, the difference between pretraining and training lies in the labels used for the MSE calculation.

**Pretraining stage.** Given a target wafer pattern (image)  $R^*$ , LithoBench provides a reference mask  $M_{\text{ref}}$ . During the pretraining stage, the mask optimization model is trained to generate a mask  $M$  that follows this reference mask  $M_{\text{ref}}$  using the MSE loss:

$$\mathcal{L}_{\text{pretrain}} = \mathcal{L}_{\text{mse}}(M, M_{\text{ref}}). \quad (25)$$

**Training stage.** The generated mask  $M$  passes through a lithography simulator  $g(\cdot)$  to obtain a simulated resist image  $g(M)$ . The model is trained to minimize the difference between this simulated resist image and the ground-truth target wafer pattern (image)  $R^*$ .

$$\mathcal{L}_{\text{training}} = \mathcal{L}_{\text{mse}}(g(M), R^*). \quad (26)$$

### E.2.2. Hyperparameters

To ensure a fair comparison, all competing methods adopt the same training procedures and hyperparameters. For MetalSet, we employ 2 epochs of pretraining, then 8 epochs of training, with a batch size of 8. For ViaSet, we adopt 1 epoch of pretraining, then 2 epochs of training, again using a batch size of 8. In the OP kernel, we set the kernel size to 11. We consistently use the default learning rate, its scheduling, and additional hyperparameters from LithoBench to maintain fair comparisons across different models.

Table 11. Hyperparameter configurations.

Setting	Hyperparameter	Configuration
MetalSet	Pretrain epochs	2
	Train epochs	8
ViaSet	Pretrain epochs	1
	Train epochs	2
Default settings	Batch size	8
	Optimizer	Adam
	Learning rate	0.001
	Learning rate decay	0.1
	Learning rate decay policy	Step
	Learning rate decay epoch	Train epochs // 2

### E.2.3. Implementation Details of ILILT

We incorporate a CFNO backbone to ILILT but refine the iterative unrolling procedure to reduce computational overhead. Rather than summing unrolled losses starting at step  $T/2$ , we update its model at every unrolling step, lowering cost and accelerating training. Also, we do not apply an exponential weighting to the loss across different unrolling depths, we update its loss directly at each unrolled step. In our experiments, we use the unrolling depth to  $T = 2$ .

### E.2.4. Computational Cost

We present additional analysis on computational cost. Our approach achieves significantly faster speeds than the current state-of-the-art ILILT model. Moreover, adding the OptiCo block to our backbone does not noticeably increase the overall computational cost. In fact, OptiCo requires less overhead compared to incorporating an FNO-based module, demonstrating a favorable trade-off between accuracy and efficiency. All experiments were conducted using PyTorch and TorchLitho [18, 33, 59] with an NVIDIA RTX 3090 GPU, except for ILILT, which was tested on an NVIDIA A6000.

Table 12. Runtime comparison of classical solvers and DL-based methods.

Classical Solver	Inference (s/img)	DL-based Model	Training (s/epoch)	Inference (s/img)
Abbe simulator	0.191	DOINN	1793	0.107
Hopkins simulator	3.946	CFNO	2246	0.129
PixelILT	3.188	DAMO	5930	0.149
CurvyILT	12.476	ILILT	21237	0.441
		Backbone	6954	0.138
		Backbone+FNO	8412	0.162
		Backbone+OptiCo (Ours)	7880	0.151

## F. Related Work

### F.1. Diffraction-Inspired Mask Optimization Techniques

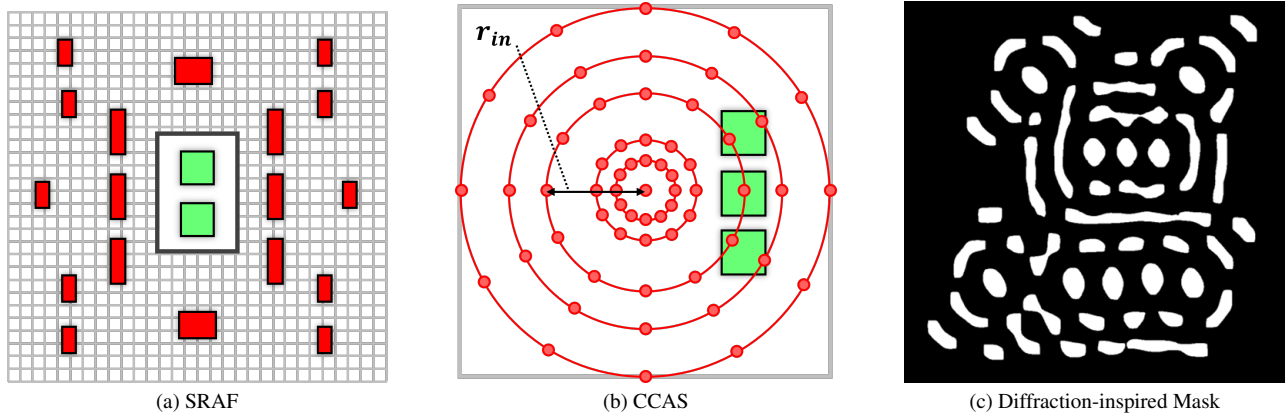


Figure 13. Visualization of diffraction-inspired mask optimization techniques. (a) Sub-Resolution Assist Features (SRAF), which deliberately shape nearby diffraction to improve printed pattern fidelity. (b) Concentric Circle Area Sampling (CCAS), a diffraction-motivated feature extractor that samples radial propagation patterns to guide assist feature placement. (c) Diffraction-inspired mask generated by our method, where ring-like structures similarly originate from diffraction-based priors. Green regions denote the target wafer design patterns.

As chip sizes and technology nodes continue to shrink, light propagation in lithography is dominated by diffraction rather than conventional ray optics. In this regime, the printed contour of a design is strongly affected by the phase and amplitude of neighboring diffracted light fields. To compensate with these wave effects, Sub-Resolution Assist Features (SRAFs) were introduced, small helper shapes that placed around to the target pattern. These helpers do not print on the wafer, but they adjust how light spreads, reinforcing the desired shape and the final print [48]. SRAFs are one of the earliest and clearest examples of using diffraction behavior directly in mask design.

Building on the same physical intuition, the Concentric Circle Area Sampling (CCAS) layout [31] injects prior knowledge of wave-based propagation into the mask optimization process. CCAS draws a set of concentric rings around a target pattern and uses as a hand-crafted, diffraction-inspired feature extractor to assist SRAF placement [55]. The circular pattern reflects the fact that diffraction from a small feature naturally spreads in a radially structured manner.

Our diffraction-inspired OP kernel design in the main text directly follows this line of work. The rings surrounding each target pattern play the same conceptual role as SRAFs and CCAS: they provide a structured, diffraction-motivated circular support region where the mask can be modified to control the nearby optical field. Thus, the “diffraction-inspired OP kernel” in our method can be viewed as a **fully learnable, CCAS-style form of diffraction engineering** that supports SRAF-like behavior in modern mask optimization.

### F.2. Mask Optimization

Mask optimization in computational lithography is the process of iteratively refining the mask layout to ensure that the patterns printed on the wafer closely match the intended design. This process can be viewed as the inverse of lithography simulation: rather than predicting the resist image (wafer pattern) from a given mask, mask optimization starts with the desired wafer pattern and works backward to compute the mask design that will produce that pattern under the constraints of the lithography process. Similar to LithoGAN’s [57] pioneering use of GAN for lithography simulation, GAN-OPC [53] emerged as one of the early DL methods that employ a conditional GAN architecture to improve mask optimization. Following this, Neural-ILT [22] proposed a fine-grained, end-to-end correction framework based on the UNet architecture, which significantly improved the mask optimization process. Subsequently, L2O-ILT [67] proposed an innovative approach by embedding iterative optimization algorithms into the neural network architecture, enabling the generation of high-quality masks. More recently, reinforcement learning (RL) algorithms have been actively applied to mask optimization frameworks to achieve superior performance. RL is particularly well-suited for this mask optimization task because it can optimize non-differentiable objectives by utilizing feedback in the form of rewards, allowing the model to learn optimal policies through trial and error [38]. For instance, RL-OPC [24] and CAMO [25] have successfully incorporated RL algorithms into their

frameworks and demonstrated outstanding performance by leveraging the exploratory capabilities of RL agents to navigate the complex design space.

### F.3. Lithography Simulation

Lithography simulation is the computational process of modeling and predicting the performance of a lithographic system, assessing how effectively a mask pattern transfers onto silicon wafers. In recent years, a variety of machine learning (ML) and deep learning (DL) methods have been developed to advance computational lithography simulation [65]. One of the pioneering frameworks in this domain is LithoGAN [57], which was among the first DL models to utilize a Generative Adversarial Network (GAN) for creating a direct correspondence between input mask designs and their resulting wafer patterns. Building on the foundation laid by LithoGAN, DAMO [7] introduced enhancements by incorporating a UNet++ backbone [37, 45, 66] and residual blocks into its architecture. These modifications allowed DAMO to achieve higher-resolution mask predictions, thereby improving the fidelity and precision of the simulated wafer patterns. Further advancing the field, TEMPO [58] addressed the challenges associated with 3D masks in Extreme Ultraviolet (EUV) lithography. To be more specific, TEMPO employed multi-domain image-to-image translation techniques to accurately predict the intensity of aerial images at varying resist heights. In addition to these developments, recent studies have explored the application of Fourier Neural Operators (FNO) [23] to enhance lithography simulation. DOINN [54] was one of the first to introduce an FNO-based framework, utilizing Fourier transforms to capture both low-frequency global information and high-frequency local details. Building upon this success, CFNO [50] further refined and advanced the approach by integrating inductive lithography biases into the model architecture inspired by ViT [12].

### G. Backbone Neural Networks

In this section, we provide an overview of backbone neural networks used in computational lithography, discussing their general role and emerging trends in the field. Additionally, we present detailed descriptions and visualizations of the specific backbone architectures employed in our study.

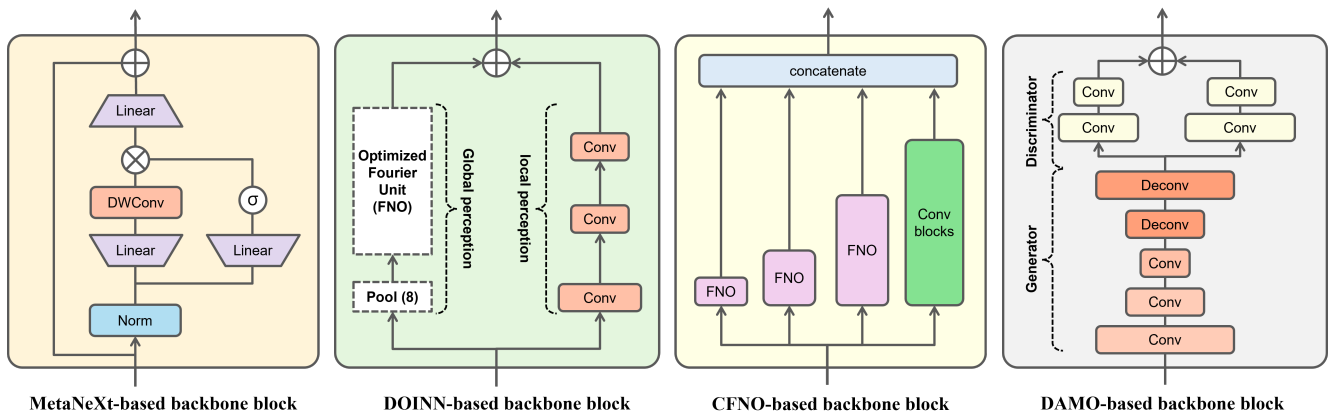


Figure 14. Visualizations of the backbone neural network used in our study, including MetaNeXT, DOINN, CFNO, and DAMO.

Backbone neural networks play a critical role in both lithography simulation and mask optimization by extracting features and encoding information from input mask or resist images. Traditionally, convolutional neural networks (CNNs) have been the backbone of choice in these tasks due to their inherent ability to capture spatial hierarchies and intricate patterns within images. In particular, CNNs leverage localized receptive fields and weight sharing, which not only enhance their computational efficiency but also enable them to recognize and process spatial features effectively. However, despite their strengths, CNNs face limitations in capturing global context and long-range dependencies within data. The localized nature of convolutional filters restricts their ability to integrate information across distant regions of an image. To address these challenges, Fourier Neural Operators (FNOs) have been introduced as an alternative backbone architecture. FNOs operate in the Fourier domain, where they can efficiently model global patterns and capture long-range dependencies by transforming spatial data into the frequency domain [23]. In addition to FNOs, transformer [43] architectures have emerged as a powerful option for backbone networks. Specifically, transformer architectures utilize self-attention mechanisms, which enable them to weigh the significance of different parts of the input data dynamically. This capability allows the transformer to capture

global dependencies and intricate relationships within the data [42]. Recently, Mamba architectures [16], which integrate spatial attention with efficient computation, have emerged as a compelling alternative to transformers, and they also show promise as potential backbone networks.

As shown in Figure 14, we present visualizations of the backbone neural networks used in our study, including MetaNeXt, DOINN, CFNO, and DAMO. Each architecture employs distinct mechanisms for feature extraction and representation. Notably, MetaNeXt utilizes depth-wise convolution (DWConv) blocks [27, 60], which efficiently capture spatial features by applying independent convolutional filters to each channel. This lightweight yet powerful design enables MetaNeXt to extract fine-grained details while maintaining computational efficiency. Due to its simplicity and effectiveness, MetaNeXt serves as our primary backbone neural network. On the other hand, DOINN and CFNO integrate Fourier Neural Operator (FNO) blocks, which operate in the Fourier domain to implicitly model light behavior. By transforming spatial data into frequency space, these architectures effectively capture both global and local features, improving their ability to model lithographic processes. Given the promising aspects of FNO-based approaches in computational lithography, we conducted additional experiments to evaluate the versatility and compatibility of OptiCo when applied to these architectures. Our results confirm that OptiCo is not limited to a single backbone but can be seamlessly incorporated into various network architectures, including FNO-based backbones like DOINN and CFNO, further demonstrating its adaptability and effectiveness in computational lithography.

## H. Limitations and Future Study

In this paper, we focused on lithography modeling by incorporating fundamental physical principles through the Fresnel diffraction and demonstrated strong performance in both lithography simulation and mask optimization tasks. However, we think several avenues exist for further improving the mask optimization task. Future work could investigate advanced optimization techniques, such as progressive optimization [39], inverse optimization [6, 52], Pareto optimization [34], reducing mini-pixel errors [56], and data augmentation [26] to address the inherent complexities of the mask optimization task in computational lithography. Another promising direction is to consider source-level modeling jointly with mask optimization [17, 28], which could further improve fidelity under real optical conditions and interferograms [21].