Optical Diffraction-based Convolution for Semiconductor Mask Optimization

Young-Han Son   **Dong-Hee Shin**   **Deok-Joong Lee**
Hyun Jung Lee   **Hyeonyeong Nam**   **Tae-Eui Kam**
Department of Artificial Intelligence, Korea University
{yhson135,dongheeshin,deokjoong,hyulee,hy_nam,kamte}@korea.ac.kr

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 [1, 2, 3]. 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 alone accounts for approximately 30% of the overall manufacturing cost [4].

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 [5, 6, 7]. 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 [8].

To achieve the precision in lithography, traditional approaches have relied on trial-and-error experimentation or mathematical modeling to optimize lithographic parameters [9, 10, 11]. However, repeatedly executing the real lithography process is prohibitively expensive, and the complexity of light diffraction makes mathematical modeling both challenging [4, 12]. To address these constraints, the semiconductor industry has adopted simulation-based computational lithography, particularly those leveraging deep learning (DL) methods [13, 14, 15].

(a) Semiconductor lithography and light diffraction   **(b) Semiconductor mask optimization**
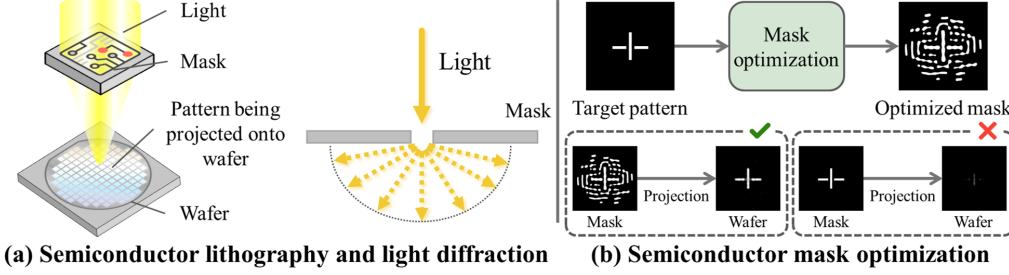
Figure 1: Deviations in mask-to-wafer projection by optical effects.

Among the various DL-based approaches, GAN-OPC [16] 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 [17] enhanced performance by achieving higher-resolution mask predictions by incorporating a UNet++ [18] backbone and residual blocks. TEMPO [19] further leveraged multi-domain image-to-image translation techniques to address the complexities associated with 3D mask designs.

Despite these achievements, some studies [20, 21] have highlighted limitations in their reliance on standard neural network architectures, which may struggle to capture the global information essential for accurate lithography modeling [22]. To overcome this limitation, researchers have explored the potential of Fourier Neural Operators (FNOs) [23]. Specifically, DOINN [20] 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 [21] further improved the framework by incorporating inductive lithography bias into the model architecture. While 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 [24, 25] 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 [26, 27]. 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 [1]. A key aspect of light diffraction is the phase factor, which quantifies changes in the wavefront of light when it encounters different material interfaces [28]. 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 [29].

To model this phase factor within our neural network, we introduce a *Optical Phase (OP) kernel* that functions as a transformation layer to encode the spatial phase variations induced by light propagation. Our framework was designed to be kernel-agnostic, allowing the use of either the Rayleigh–Sommerfeld-based diffraction kernels or the Hopkins diffraction kernel, as well as any other physically motivated kernels as needed. Unlike traditional convolution filters that focus on spatial features, our kernel is designed to capture the curvature and dynamic evolution of the light wavefront, thereby simulating the behavior of light as it propagates through space. By incorporating this physics-inspired operation into our neural network 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.

The main contributions of our work can be summarized as follows:

- We propose a **Opti**cal diffraction-based **Co**nvolutional 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 behavior.

- We introduce a *optical phase (OP) kernel* that encodes spatial phase variations caused by light propagation. This physics-inspired OP kernel significantly improves the generalization performance for OOD mask designs.
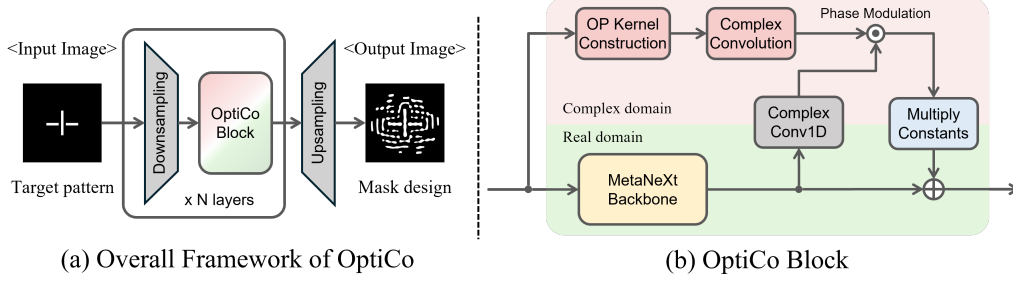
(a) Overall Framework of OptiCo   (b) OptiCo Block

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

2 Method

Convolution from Optical Diffraction. In wave optics, the propagation of an optical wave U from the mask plane (x, y) to a wafer plane (x', y') at distance z can be described by the Rayleigh-Sommerfeld (RS) diffraction integral [27].

The RS integral has several formulations, including the Helmholtz–Kirchhoff, the Green's function, as well as the Fresnel approximation. For clarity of presentation, we illustrate the Fresnel form here, while our framework accommodates all formulations.

$$U(x', y') = \frac{e^{jkz}}{j\lambda z} \iint_{-\infty}^{\infty} U(x, y) e^{\frac{jk}{2z}[(x'-x)^2 + (y'-y)^2]} dx dy. \tag{1}$$

This diffraction 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 defined, the Fresnel diffraction integral in Eq. 1 can be rewritten by isolating the quadratic dependence on $(x' - x)$ and $(y' - y)$. This transforms the integral using convolution as follows:

$$U(x', y') = \frac{e^{jkz}}{j\lambda z} \left[U(x, y) * h(x', y') \right]; \quad h(x', y') = \exp\left(\frac{jk}{2z} \left[x'^2 + y'^2 \right] \right). \tag{2}$$

The kernel h acts as an Optical Phase (OP) kernel, characterizing diffraction by phase modulation.

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 $Q(x, y) = \exp\left(\frac{jk}{2z} \left[x^2 + y^2 \right] \right)$. Each element of $Q(x, y)$ corresponds to the phase term of diffraction. While we illustrate the OP kernel using the Fresnel form above, our framework is kernel-agnostic and can equally incorporate more general RS formulations or the Hopkins TCC kernel.

OP complex convolution. Given the diffraction-inspired OP kernel Q and learnable scalar weight λ, $W_{\text{eff}} = \lambda \cdot Q$, where the resulting complex effective kernel W_{eff} encodes optical diffraction, so that each convolution step inherently reflects the physical propagation of light. Since optical fields are inherently complex, we decompose the input U and weight W_{eff} into real (r) and imaginary (i) as:

$$U = U_r + jU_i; \quad W_{\text{eff}} = W_r + jW_i. \tag{3}$$

$$\text{OP}_{\text{conv}}(U) = \left[(U_r * W_r - U_i * W_i) + j \left(U_r * W_i + U_i * W_r \right) \right]. \tag{4}$$

OptiCo block. The OptiCo block integrates a MetaNeXt-based backbone [30] with the OP complex convolution in a unified layer. Since the OP convolution is inherently complex, we first embed Y_{backbone} into the complex domain using a per-pixel Complex Conv1D, yielding a complex feature. This embedding is then combined with OP_{conv} via a Hadamard product, performing phase modulation. The modulated output is multiplied by the constants and finally added back to the backbone feature, thereby injecting physically grounded diffraction effects into the representation.

$$Y_{\text{backbone}}(U) = \left(\text{DWConv}(\text{Norm}(U_r)W_1) \odot \sigma(\text{Norm}(U_r)W_2) \right) W_3 + U_r, \tag{5}$$

$$Y_{\text{phase}}(U) = \left[\text{ComplexConv1D}(Y_{\text{backbone}}(U)) \odot \text{OP}_{\text{conv}}(U) \right] \cdot \frac{e^{jkz}}{j\lambda z}, \tag{6}$$

$$Y_{\text{OptiCo}} = Y_{\text{backbone}}(U) + Y_{\text{phase}}(U). \tag{7}$$

3

Table 1: Main experiments comparing MSE, PVB, and EPE across competing methods.

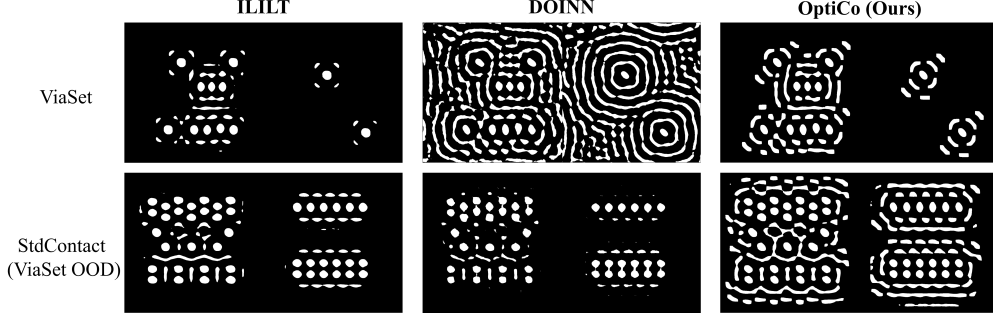| Method	MetalSet			ViaSet			StdMetal (MetalSet OOD)			StdContact (ViaSet OOD)			Average		
	MSE	PVB	EPE	MSE	PVB	EPE	MSE	PVB	EPE	MSE	PVB	EPE	MSE	PVB	EPE
DAMO	32579	**41173**	5.4	5081	9962	**0.0**	16120	**23796**	0.2	50445	35673	26.7	26056	27651	8.1
DOINN	36409	41929	7.4	4382	7836	**0.0**	25913	25749	4.5	72058	17968	55.8	34691	**23370**	16.9
CFNO	47814	46131	12.5	8949	9890	0.1	26809	26814	4.2	70740	**17950**	55.1	38578	25196	18.0
ILILT	30353	45353	3.2	4666	10065	**0.0**	14596	24969	0.1	38957	43869	7.1	22143	31064	2.6
OptiCo	**24033**	45327	**1.6**	**4339**	**7802**	**0.0**	**11293**	25183	**0.0**	**18909**	39309	**0.2**	**14535**	29373	**0.4**

Figure 3: 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.

3 Experiments

Dataset. To evaluate the performance of OptiCo, we used LithoBench [31], which is one of the most recent and widely known computational lithography benchmarks, including synthetic and real-world designs. Additional explanations about baselines and evaluation metrics are provided in Appendix D.

Main results. Table 1 shows that 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 OOD subtasks, which are inherently more challenging. Specifically, in ViaSet, most methods reach zero EPE, showing strong adaptation to the training set. However, on the OOD test set (StdContact), their EPE rises sharply, while OptiCo maintains robust performance. Additional comparative ablation studies on the OP kernel are provided in Appendix A.1. A.4 shows that the OP kernel works with different diffraction formulations (Helmholtz–Kirchhoff, Green's function, and Hopkins). Ablations with different backbone networks are in A.2, and the train dataset ratio is in A.3.

Visualization. We compared the mask patterns optimized by OptiCo with those from other competing methods in Figure 3. 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. Additional visualizations for all methods and other subtasks are provided in Appendix C.1. To further examine how the OP kernel influences internal representations, we provide additional intermediate feature visualizations comparing models with and without OptiCo in Appendix C.2.

4 Conclusion

In this paper, we propose OptiCo, a novel physics-inspired convolutional neural network that integrates optical diffraction principles directly into its architecture to enhance computational lithography. Unlike conventional approaches that rely solely on data-driven methods or Fourier-based frameworks, OptiCo introduces a physics-inspired OP kernel to explicitly model the diffraction behavior of light. Empirical evaluations on well-established lithography benchmarks demonstrate that our OptiCo consistently outperforms existing state-of-the-art methods in mask optimization tasks.

Acknowledgments and Disclosure of Funding

References

[1] Wayne M Moreau. *Semiconductor lithography: principles, practices, and materials*. Springer Science & Business Media, 2012.

[2] 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.

[3] 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.

[4] Xu Ma and Gonzalo R Arce. *Computational lithography*. John Wiley & Sons, 2011.

[5] Vivek Bakshi. *EUV lithography*. SPIE press, 2009.

[6] 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*, volume 10962, pages 175–183. SPIE, 2019.

[7] Iason Giannopoulos, Iacopo Mochi, Michaela Vockenhuber, Yasin Ekinci, and Dimitrios Kazazis. Extreme ultraviolet lithography reaches 5 nm resolution. *Nanoscale*, 16(33):15533–15543, 2024.

[8] 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.

[9] Chris A Mack. Thirty years of lithography simulation. In *Optical Microlithography XVIII*, volume 5754, pages 1–12. SPIE, 2005.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Mentor Calibre. `https://eda.sw.siemens.com/en-US/ic/calibre-design/`.

[15] Yuki Watanabe, Taiki Kimura, Tetsuaki Matsunawa, and Shigeki Nojima. Accurate lithography simulation model based on convolutional neural networks. In *Optical Microlithography XXX*, volume 10147, pages 137–145. SPIE, 2017.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] Haoyu Yang, Zongyi Li, Kumara Sastry, Saumyadip Mukhopadhyay, Mark Kilgard, Anima Anandkumar, Brucek 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.

[21] 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.

[22] 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.

[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.

[24] Frederick H Dill. Optical lithography. *IEEE transactions on electron devices*, 22(7):440–444, 1975.

[25] Eugene Hecht. *Optics*. Pearson Education India, 2012.

[26] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013.

[27] Arnold Sommerfeld. *Partial differential equations in physics*. Academic press, 1949.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] Haoyu Yang and Haoxing Ren. Ililt: Implicit learning of inverse lithography technologies. In *International conference on machine learning*. PMLR, 2024.

[34] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[35] 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.

[36] 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.

[37] 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.

[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.

[39] Xiaoxiao Liang, Yikang Ouyang, Haoyu Yang, Bei Yu, and Yuzhe Ma. Rl-opc: Mask optimization with deep reinforcement learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[46] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[47] Young-Han Son, Dong-Hee Shin, and Tae-Eui Kam. FTMMR: Fusion transformer for integrating multiple molecular representations. *IEEE Journal of Biomedical and Health Informatics*, 2024.

[48] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[49] 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.

[50] Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? *arXiv preprint arXiv:2405.07992*, 2024.

[51] 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.

[52] Charles L Byrne. *Iterative optimization in inverse problems*. CRC Press, 2014.

[53] Haoyu Yang and Haoxing Ren. Gpu-accelerated inverse lithography towards high quality curvy mask generation. *arXiv preprint arXiv:2411.07311*, 2024.

[54] Chao Qian, Yang Yu, and Zhi-Hua Zhou. Subset selection by pareto optimization. *Advances in neural information processing systems*, 28, 2015.

[55] 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.

[56] Mingjie Liu, Haoyu Yang, Brucek 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.

[57] 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.

[58] Kramer Harrison. Optiland, January 2025. Version 0.2.3. Available at `https://github.com/HarrisonKramer/optiland`.

[59] 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.

A Additional Experiments and Results

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

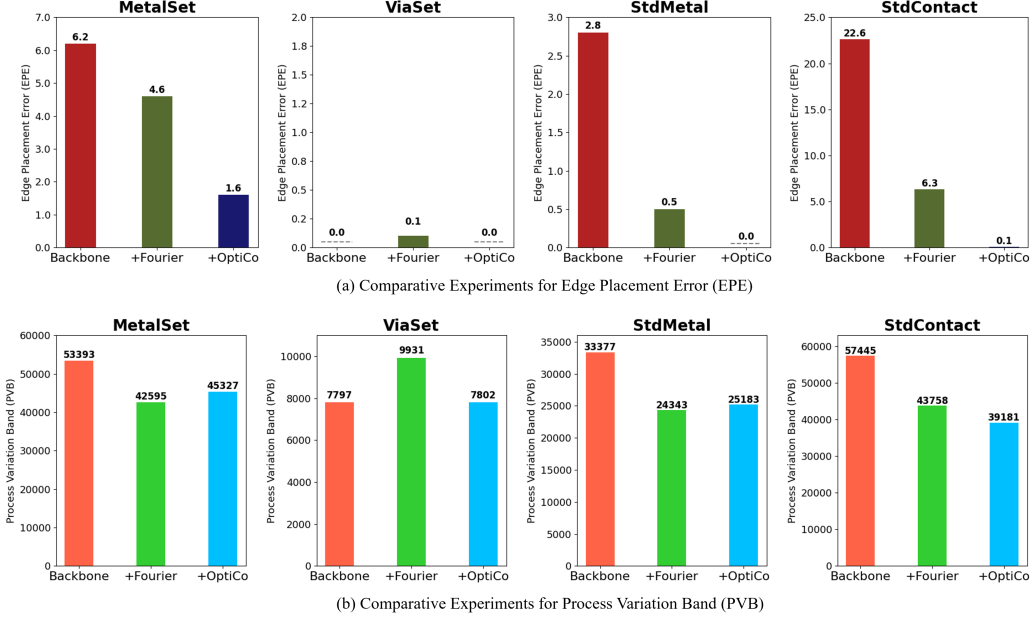(b) Comparative Experiments for Process Variation Band (PVB)

Figure 4: Comparative performance of the backbone network with the addition of the Fourier module 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.

A.1 Additional Comparative Experiments for Various Evaluation Metrics

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 4 (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 4 (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 parabolic phase variations inherent in Fresnel 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.

Table 2: 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	**41929**	7.4
	+ OptiCo (ours)	**28322**	44846	**3.1**
	CFNO	47814	46131	12.5
	+ OptiCo (ours)	**38586**	**45743**	**7.5**
ViaSet	DOINN	4382	7836	**0.0**
	+ OptiCo (ours)	**4362**	**7819**	**0.0**
	CFNO	8949	9890	0.1
	+ OptiCo (ours)	**4406**	**8505**	**0.0**
StdMetal (MetalSet OOD)	DOINN	25913	25749	4.5
	+ OptiCo (ours)	**15589**	**25408**	**0.8**
	CFNO	26809	26814	4.2
	+ OptiCo (ours)	**22542**	**25328**	**2.5**
StdContact (ViaSet OOD)	DOINN	72058	**17968**	55.8
	+ OptiCo (ours)	**23213**	40113	**2.2**
	CFNO	70740	**17950**	55.1
	+ OptiCo (ours)	**23150**	42731	**1.7**
Average	DOINN	34691	**23370**	16.9
	+ OptiCo (ours)	**17871**	29546	**1.5**
	CFNO	38578	**25196**	18.0
	+ OptiCo (ours)	**22171**	30577	**2.9**

A.2 OptiCo with Different Backbone Networks

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 2, 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.

A.3 Dataset Ablation

Physics-based inductive biases are known to substantially improve data efficiency [32]. 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 3 and Figure 5, models incorporating physics priors achieve significantly lower EPE scores at all data regimes, demonstrating their superior efficiency under limited data.

Table 3: 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.

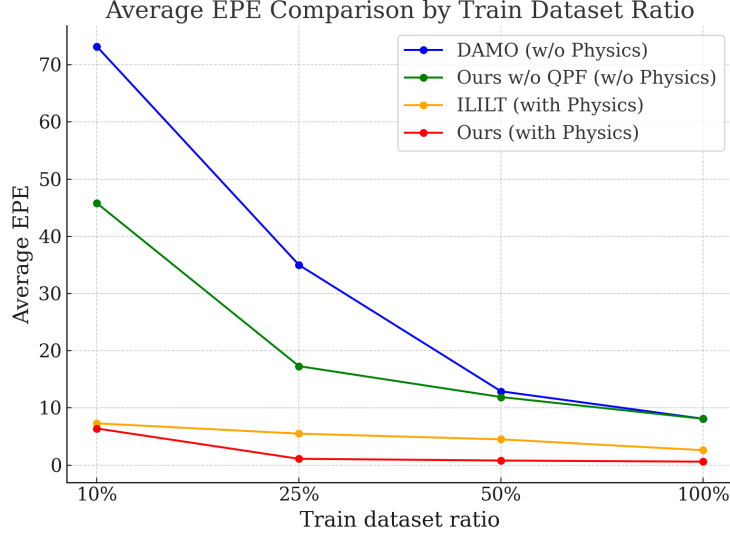| Training ratio	Without physics prior		With physics prior	
	DAMO	Ours w/o OP	ILILT	Ours
10%	73.2	45.8	7.3	6.4
25%	35.0	17.3	5.5	1.1
50%	12.9	11.9	4.5	0.8
100%	8.1	8.1	2.6	0.4

Figure 5: Ablation studies of training dataset ratio. We compare the performance of models with and without physics priors across varying ratios of training data.

A.4 Ablation on Alternative Diffraction Kernels

In the main text, we illustrated the OP kernel using the Fresnel approximation for clarity. However, our framework is *kernel-agnostic* and can readily incorporate other diffraction formulations. To verify this property, we conducted ablation experiments by replacing the Fresnel kernel with alternative diffraction kernels.

Helmholtz–Kirchhoff. The RS kernel derived from the scalar Helmholtz equation with Kirchhoff boundary conditions, including the obliquity factor:

$$h(x, y, z) = \frac{\exp\left(jk\sqrt{x^2 + y^2 + z^2}\right)}{x^2 + y^2 + z^2} \left(1 + \frac{j}{k\sqrt{x^2 + y^2 + z^2}}\right).$$

Green-function. An equivalent spatial-domain expression omitting the obliquity factor, often used as a simplified RS kernel:

$$h(x, y, z) = \frac{\exp\left(jk\sqrt{x^2 + y^2 + z^2}\right)}{x^2 + y^2 + z^2}.$$

Fresnel. The paraxial approximation of the RS/Angular-Spectrum propagation that yields the Quadratic Phase Factor (QPF) kernel, which is used in the main text:

$$h(x, y, z) = \exp\left(\frac{jk}{2z}\left(x^2 + y^2\right)\right).$$

11

Hopkins method and TCC kernel. Aerial-image intensity written with the transmission cross coefficient (TCC):

$$I(x,y) = \iint \mathcal{F}(M_{f_1})\mathcal{F}(M_{f_2}^*)TCC(f_1, f_2)df_1df_2,$$

$$\mathcal{F}\{\text{TCC}(f_1, f_2)\} = \sum \alpha \, hh^*.$$

The quantitative results are reported in Table 4. We observe that all variants can be integrated into OptiCo without modification, and each kernel demonstrates competitive performance. Notably, the Green's function kernel shows strong results on the Metal task, while the Fresnel kernel achieves the lowest error on StdMetal. These results confirm that OptiCo is not restricted to a particular diffraction formulation and robustly adapts to different physics-based kernels.

Table 4: Ablation study using different diffraction kernels. Results are reported on the Metal and StdMetal subtasks.

Kernel	Metal ($\downarrow$)	StdMetal ($\downarrow$)
Helmholtz–Kirchhoff	2.1	0.12546
Hopkins (TCC)	1.7	0.07011
Green's function	**1.2**	0.06273
Fresnel	1.6	**0.04428**

B Pytorch-like Algorithm for QPF Kernel Construction

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

```
def create_Q_kernel(k, z, 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*z)
    Q = torch.exp(exponent)
    return Q
```

Algorithm 1: PyTorch implementation of the QPF kernel

C Additional Visualization of Mask Patterns and Intermediate Features

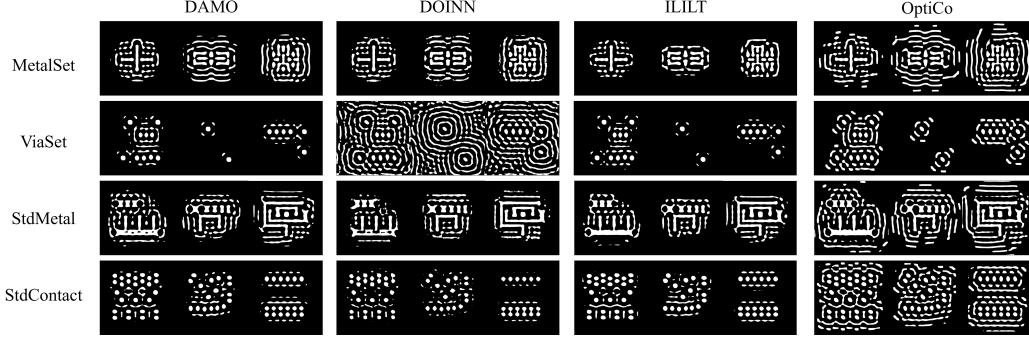## C.1 Additional Sample Mask Pattern Visualizations

Figure 6: Visualization of additional optimized mask patterns across all competing methods.

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 6 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.

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.
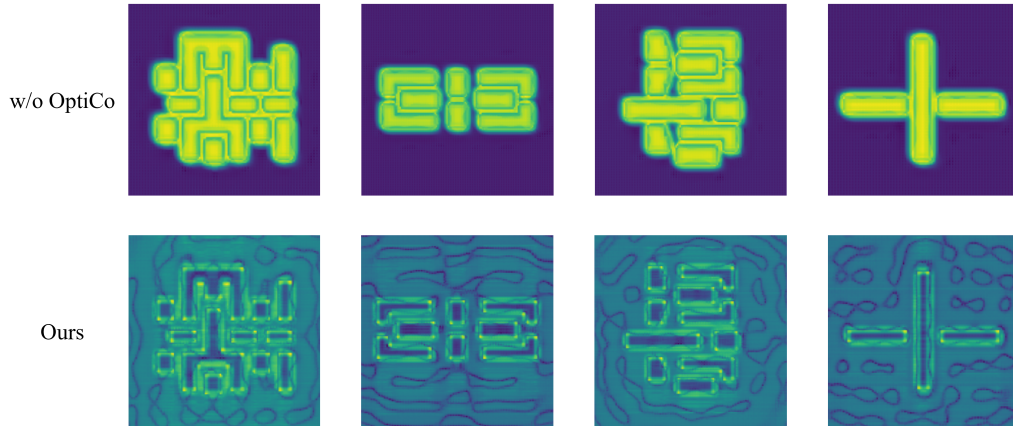
C.2 Intermediate Feature Visualization

Figure 7: 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 7. 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 E.2 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.

D   Experimental Details

D.1   Details of Evaluation Metrics in Semiconductor Computational Lithography
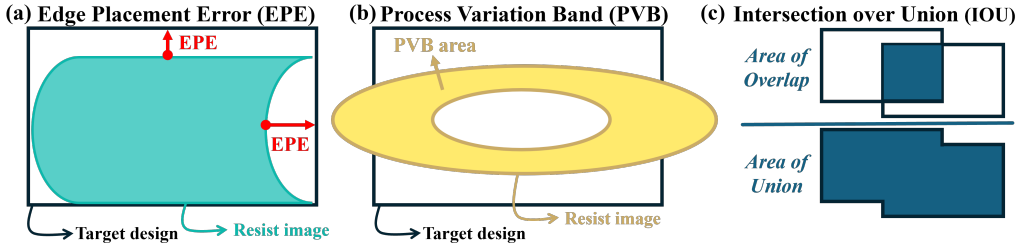

Figure 8: Illustrations of various evaluation metrics in computational lithography: (a) Edge Placement Error (EPE), which quantifies the deviation of edges from their expected positions; (b) Process Variation Band (PVB), showing the tolerance range of variations; and (c) Intersection over Union (IOU), 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 8 (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 8 (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 8 (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.

D.2   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 [17] is a fully data-driven model that uses a UNet++ [18] backbone. In addition, DOINN [20] and CFNO [21] incorporate optical physics priors by leveraging FNOs architectures.

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 [33] is a recent state-of-the-art mask optimization model that iteratively inputs simulated resist images along with target patterns.

D.3 Hyperparameters and Training Strategies

D.3.1 Overall Training Procedure of Mask Optimization

Our method follows the standard training methodology provided by LithoBench [31] 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_{ref}. During the pretraining stage, the mask optimization model is trained to generate a mask M that follows this reference mask M_{ref} using the MSE loss:

$$\mathcal{L}_{\mathrm{pretrain}} = \mathcal{L}_{\mathrm{mse}}(M, M_{\mathrm{ref}}). \tag{8}$$

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^*.

$$\mathcal{L}_{\mathrm{training}} = \mathcal{L}_{\mathrm{mse}}(g(M), R^*). \tag{9}$$

D.3.2 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 5: 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

D.3.3 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 $T = 2$.

D.3.4 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 [34] with an NVIDIA RTX 3090 GPU, except for ILILT, which was tested on an NVIDIA A6000.

Table 6: Training and inference time comparison across models.

Model	Training (s / epoch)	Inference (s / img)
DOINN	1793	0.107
CFNO	2246	0.129
DAMO	5930	0.149
ILILT	21237	0.441
Backbone	6954	0.138
Backbone+FNO	8412	0.162
Backbone+OptiCo (Ours)	7880	0.151

E Related Work

E.1 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 [35] pioneering use of GAN for lithography simulation, GAN-OPC [16] emerged as one of the early DL methods that employ a conditional GAN architecture to improve mask optimization. Following this, Neural-ILT [36] proposed a fine-grained, end-to-end correction framework based on the UNet architecture, which significantly improved the mask optimization process. Subsequently, L2O-ILT [37] 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 [39] and CAMO [40] 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.

E.2 Diffraction-Inspired Mask Optimization Techniques

As technology nodes continue to shrink, diffraction effects become increasingly significant in lithography, motivating the widespread use of Sub-Resolution Assist Features (SRAFs) to capture and compensate for these effects [41]. SRAF refers to assist blocks placed around the main pattern to mitigate diffraction-induced errors and improve pattern fidelity. In parallel, the Concentric Circle Area Sampling (CCAS) layout [42] also draws inspiration from diffraction, injecting prior knowledge into the mask optimization process via a circular, pre-defined feature extractor. This diffraction-inspired feature extractor is designed to improve SRAF placement by explicitly accounting for wave-based propagation characteristics [3]. Notably, SRAF corresponds to the diffraction-inspired rings discussed in the main text.

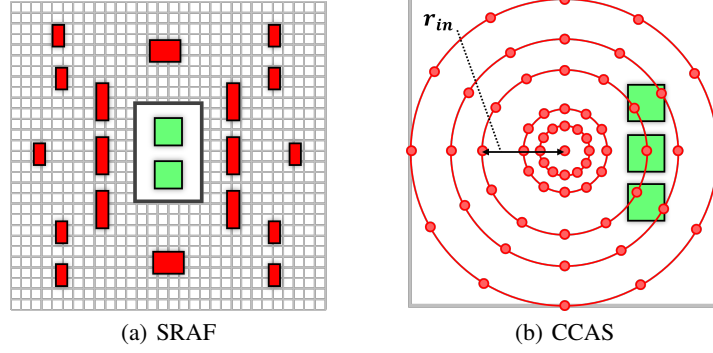|	
(a) SRAF	(b) CCAS

Figure 9: Visualization of diffraction-inspired mask optimization techniques. (a) Sub-Resolution Assist Features (SRAF) and (b) Concentric Circle Area Sampling (CCAS). Green regions indicate the target wafer design patterns.

E.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 [31]. One of the pioneering frameworks in this domain is LithoGAN [35], 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 [17] introduced enhancements by incorporating a UNet++ backbone [18, 43, 44] 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 [19] 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 [20] 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 [21] further refined and advanced the approach by integrating inductive lithography biases into the model architecture inspired by ViT [45].

F 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.

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 [46] 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
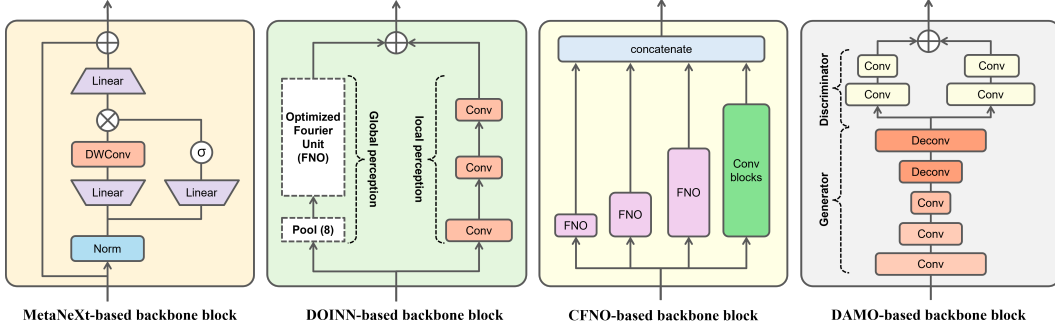
17

Figure 10: Visualizations of the backbone neural network used in our study, including MetaNeXT, DOINN, CFNO, and DAMO.

the input data dynamically. This capability allows the transformer to capture global dependencies and intricate relationships within the data [47]. Recently, Mamba architectures [48], 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 10, 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 [49, 50], 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.

G 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 [51], inverse optimization [52, 53], Pareto optimization [54], reducing mini-pixel errors [55], and data augmentation [56] 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 [57, 58], which could further improve fidelity under real optical conditions and interferograms [59].