Enabling Scalable AI Computational Lithography with Physics-Inspired Models

Haoyu Yang
NVIDIA
haoyuy@nvidia.com

Haoxing Ren
NVIDIA
haoxingr@nvidia.com

Abstract

Computational lithography is a critical research area for the continued scaling of semiconductor manufacturing process technology by enhancing silicon printability via numerical computing methods. Today's solutions for these problems are primarily CPU-based and require many thousands of CPUs running for days to tape out a modern chip. We seek AI/GPU-assisted solutions for the two problems, aiming at improving both runtime and quality. Prior academic research has proposed using machine learning for lithography modeling and mask optimization, typically represented as image-to-image mapping problems, where convolution layer backboned UNets and ResNets are applied. However, due to the lack of domain knowledge integrated into the framework designs, these solutions have been limited by their application scenarios or performance. Our method aims to tackle the limitations of such previous CNN-based solutions by introducing lithography bias into the neural network design, yielding a much more efficient model design and significant performance improvements.

CCS Concepts

• **Hardware → VLSI design manufacturing considerations**.

Keywords

Computational Lithography, Physics-Inspired Machine Learning

ACM Reference Format:
Haoyu Yang and Haoxing Ren. 2023. Enabling Scalable AI Computational Lithography with Physics-Inspired Models . In *28th Asia and South Pacific Design Automation Conference (ASPDAC '23), January 16–19, 2023, Tokyo, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3566097.3568361

1 Introduction

Computational lithography is a critical research area for the continued scaling of semiconductor manufacturing process technology by enhancing silicon printability via numerical computing methods [1–3]. Detailed topics include lithography modeling, resolution enhancements, optical proximity correction (OPC), and source mask optimization (SMO). In this paper, we focus on 1) lithography modeling, which computes the post-lithograph shape on the silicon wafer given a mask design; and 2) mask optimization (inverse lithography), which optimizes a mask design such that the remaining pattern on the silicon wafer after the lithography process is as close as the desired shape. (See Figure 1)

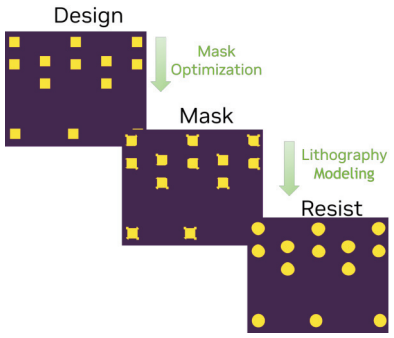

Figure 1: Lithography modeling and mask optimization.

Today's solutions for these problems are primarily CPU-based and require many thousands of CPUs running for days to tape out a modern chip. Limited by both CPU computing capabilities and legacy optimization algorithms, further QoR improvement becomes challenging. Developing AI-assisted computational lithography solutions will provide market opportunities for GPUs and avoid reliance on traditional algorithms in design for manufacturing flows, potentially helping to eventually improve yield in future chip products. Prior academic research has proposed using machine learning for lithography modeling and mask optimization, typically represented as image-to-image mapping problems [4–7]. Most solutions have reused existing computer vision-friendly neural network structures for the problem. Examples include LithoGAN [8], GAN-OPC [9], and DAMO [10], where convolution layer backboned UNets [11] and ResNets [12] are applied. The biggest challenge building AI computational lithography solution is lacking of data. A massive and well-distributed layout design data is usually hard to obtain, due to long chip design cycle and/or IP protection [13]. We seek physics-inspired AI solutions for the two problems, aiming at improving both runtime and quality.

Compensating Data with Physics. Physics-Informed Neural Networks (PINN) have been recently investigated to solve complicated physical and numerical problems, where there are limited volume of experimental data [14, 15]. Particularly, [14] introduces regimes of physical problems and their data availability, as shown in Figure 2. The nature of computational lithography makes it a problem with limited data and lots of physics, and that is why prior machine learning models can only be auxiliary components in traditional computational lithography solutions. Following the principles of physics-informed learning, there are three design

ASPDAC '23, January 16–19, 2023, Tokyo, Japan

ACM ISBN 978-1-4503-9783-4/23/01.
https://doi.org/10.1145/3566097.3568361

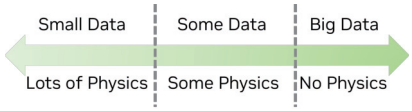

Figure 2: Regimes of physical problems and data availability, resketched from [15].

methodologies integrating physics into machine learning models: (1) *Observational bias* can be directly learned from data or datasets with underlying physical properties. An example is introducing shift/rotation invariance through data augmentation. (2) *Inductive bias* brings physics into the neural network architecture design that allows strict constraints in a machine learning system. (3) *Learning bias* focuses on specific loss functions and inference flow that reflects real physical constraints.

CFNO as Lithography Learner. Our method starts with the operator backbone design called Convolutional Fourier Neural Operator (CFNO), which inherits the advantages of Fourier Neural Operator (FNO) [16] and resembles the real physics in forward lithography modeling, adding inductive bias in neural network architecture designs. CFNO consists of three configurable components: (1) an shared FNO across input image tokens, (2) a token-wise convolution layer that aggregates intra-token dependencies and (3) a auxiliary convolution path that learns image local details. We will show later how these components bias the machine learning model for computational lithography followed by case studies on lithography modeling and mask optimization.

The remainder of the paper is organized as follows:Section 2 covers the details of the CFNO design; Section 3 presents two case studies using CFNO for lithography modeling and mask optimization; Section 4 concludes the paper with future researches.

2 Convolutional Fourier Neural Operator

In this section, we will show the analogy between FNO and mathematics in optical lithography process and detail the components of CFNO.

2.1 Analogy Between Lithography and FNO

The Hopkins diffraction model [17] is extensively studied in literature to represent lithography behavior. For a thin mask image $\boldsymbol{M} \in \mathbb{R}^{N\times N}$, the light intensity $\boldsymbol{I} \in \mathbb{R}^{N\times N}$ on the photo resist is given by

$$\boldsymbol{I}(m,n) = \tilde{\boldsymbol{s}}^{\mathsf{H}} \boldsymbol{A} \tilde{\boldsymbol{s}}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{N^2\times N^2}$ contains the transmission cross-coefficient, H denotes the conjugate transpose and $\tilde{\boldsymbol{s}} \in \mathbb{R}^{1\times N^2}$ is defined by

$$\tilde{\boldsymbol{s}} = \tilde{\boldsymbol{M}}(p,q) e^{j(2pm+2qn)\pi}, \tag{2}$$

where $\tilde{\boldsymbol{M}}(p,q) = \mathcal{F}(\boldsymbol{M}(m,n))$. To reduce the compute overhead, a singular value decomposition (SVD) approximation is typically adopted for lithography modeling. The basic idea is to take the SVD of the coefficient matrix $\boldsymbol{A}$ in the Hopkins model and formulate the lithography forward process as

$$\boldsymbol{I}(m,n) = \sum_{k=1}^{N^2} \alpha_k |\boldsymbol{h}_k(m,n) \otimes \boldsymbol{M}(m,n)|^2, \tag{3}$$

where $\boldsymbol{h}_k$ terms are lithography kernels and α_k terms are the associated eigenvalues. Refer to [17] on the details from Equation (1) to Equation (3). If we only keep the l largest α_k values for faster calculation [2, 18], eq. (3) becomes

$$\boldsymbol{I}(m,n) = \sum_{k=1}^{l} \alpha_k |\boldsymbol{h}_k(m,n) \otimes \boldsymbol{M}(m,n)|^2, l \ll N^2. \tag{4}$$

The computing cost can be further reduced if we move to Fourier space as

$$\boldsymbol{I} = \sum_{k=1}^{l} \alpha_k |\mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{h}_k) \odot \mathcal{F}(\boldsymbol{M}))|^2. \tag{5}$$

which is normally the equation used for forward simulation.

On the other hand, FNO [16] is an operator proposed to solve partial differential equations. FNO tries to find a parameterized mapping between two finite dimension spaces such that the mapping is close to the physical behavior. Numerically, FNO is given by

$$\boldsymbol{V}_{t+1} = \sigma(\mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{V}_t) \cdot \boldsymbol{W}_{\mathcal{R}})), \boldsymbol{W}_{\mathcal{R}} \in \mathbb{C}^{C\times C\times H\times W}, \tag{6}$$

where $\boldsymbol{V}_t$ and $\boldsymbol{V}_{t+1}$ represent the input and output space, respectively, $\boldsymbol{W}_{\mathcal{R}}$ corresponds to learnable parameters in FNO and $\sigma(\cdot)$ is some activation function.

Interestingly, Equation (6) has four major computing stages that resemble the aerial image computation in Equation (5), as summarized in Table 1. This hence motivates us to deploy FNO into neural network design to bias machine learning models.

2.2 CFNO Backbone

Motivated by previous discussion, we propose the operator design called CFNO [19] that targets at learning layout embeddings for down stream tasks. The basic architecture of CFNO is depicted in Figure 3, which shows three configurable components:

- *Shared FNO*: requires the input image/tensor to be divided into non-overlapped patches (tokens) and all patches will go through a shared FNO layer, as defined in Equation (6). This is the operation we designed to capture the lithography behavior in a local region and reduce the overhead of computing FFT over a large input.
- *TokenConv*: follows the output of Shared FNO. This is implemented by a convolution layer forcing all entries in a token to share the same parameter. This allows us to learn intra-token dependencies that is essential in lithography process and cannot be captured in shared FNO.
- *CNN Local Perception*: is a path containing regular convolution layers. It extracts layout features and aggragetes with TokenConv outputs.

We introduce the CNN path for two reasons: (1) convolution layers are talented for understanding image details which have significant impact on lithography results; (2) FNO plays an role of global convolution computed in frequency domain, which has

Table 1: Analogy between lithography simulation and FNO.

Step	Lithography Simulation	FNO
1	$\mathcal{F}(\boldsymbol{M})$: FFT on rasterized mask	$\mathcal{F}(\boldsymbol{V}_t)$: FFT on input space
2	$\mathcal{F}(\boldsymbol{h}_k)(\cdot)$: Linear transformation with lithography kernels	$\boldsymbol{W}_{\mathcal{R}}(\cdot)$: Linear channel mixing
3	$\mathcal{F}^{-1}(\cdot)$: Convert back to spatial domain	$\mathcal{F}^{-1}(\cdot)$: Convert back to spatial domain
4	$\alpha[\cdot]^2$: Weighted summation across intensity responses to all lithography kernels	σ: Some activation

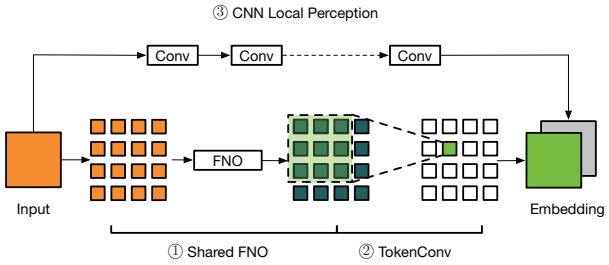

Figure 3: CFNO backbone architecture.

an assumption that the input tokens are periodic. However, the assumption does not necessarily hold for layout images, and we need additional layers to compensate the error.

For the token shared FNO, we used the same pipeline as in Equation (6). Given a layout image $\boldsymbol{Z}_t \in \mathbb{R}^{H\times W}$, we first divide it into non-overlapped patches, referred as tokens:

$$\boldsymbol{Z}_t = \begin{bmatrix} Z_{t,1,1} & Z_{t,1,2} & \dots & Z_{t,1,n} \\ Z_{t,2,1} & Z_{t,2,2} & \dots & Z_{t,2,n} \\ \dots & \dots & \dots & \dots \\ Z_{t,m,1} & Z_{t,m,2} & \dots & Z_{t,m,n} \end{bmatrix}, \tag{7}$$

where $\boldsymbol{Z}_{t,i,j} \in \mathbb{R}^{k\times k}$'s are layout tokens, $H = mk$ and $W = nk$. We define the shared FNO $f(\cdot;\boldsymbol{W}_1)$ to get the first level token embedding:

$$\tilde{\boldsymbol{T}}_{i,j} = f(\boldsymbol{Z}_{t,i,j};\boldsymbol{W}_1), i = 1, 2, ..., m, j = 1, 2, ..., n, \tag{8}$$

where $\boldsymbol{W} \in \mathbb{C}^{k\times k\times d}$ and d denotes the lifted channel number. Obviously, Equation (8) can be finished efficiently through batch processing and a smaller k indicates a shared FNO with fewer trainable parameters. However, this token-shared approach scarifies the ability of global information acquisition for model size.

To tackle this concern, we further introduce the second level token embedding via a token-wise convolution parametered with $\boldsymbol{W}_2 \in \mathbb{R}^{(2s+1)\times(2s+1)}$:

$$\boldsymbol{T}_{i,j} = \sum_{t_x=-s}^{s}\sum_{t_y=-s}^{s} \boldsymbol{W}_2[i+t_x, j+t_y]\cdot\tilde{\boldsymbol{T}}_{i+t_x,j+t_y}, \tag{9}$$

which finally formulates the layout global embedding:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_{1,1} & \boldsymbol{T}_{1,2} & \dots & \boldsymbol{T}_{1,n} \\ \boldsymbol{T}_{2,1} & \boldsymbol{T}_{2,2} & \dots & \boldsymbol{T}_{2,n} \\ \dots & \dots & \dots & \dots \\ \boldsymbol{T}_{m,1} & \boldsymbol{T}_{m,2} & \dots & \boldsymbol{T}_{m,n} \end{bmatrix}. \tag{10}$$

Equation (9) defines how tokens at different spatial locations are mixed and hence addresses token boundary inconsistency issue and long-range dependency requirements. *It should be also noted that both shared FNO and token-wise convolution are configurable according to different receptive field and model capacity demands.*

3 Case Studies

3.1 AI for Lithography Modeling

For the first case study, we investigate the possibility of CFNO backbone for learning lithography modeling process. Particularly, we focus on the process of mask to resist image mapping with dual band optics-inspired neural network (DOINN) [20].

3.1.1 Building DOINN with CFNO Backbone The architecture of DOINN is depicted in Figure 4, where we adopt a very simple CFNO instantiation in the network architecture. In the training phase, the network is designed to take $2\mu m \times 2\mu$ mask input rasterized with $1nm$ pixel size. In the forward lithography modeling task, we set the number of patches to be one. Therefore, the token-wise convolution is no longer necessary, yielding two embedding learning path: (1) global perception (GP) and (2) local perception (LP). The learned embedding will be feed into stacked convolution/transposed convolution layers for resist image reconstruction. It should be noticed that a downsample pooling is applied as FNO is not designed for high frequency details. This will also reduce the computing in FNO significantly.

In the inference phase, we want the model to be efficient in practical application scenario, where the input mask images could be with any random sizes. We hence propose the large tile simulation scheme. This is not trivial due to the existence of Fourier components, because if we feed larger tile in the FNO unit, the frequency coefficients and the trained parameters will mismatch and cause random artifacts in the generated resist images. Therefore, we proposed the large tile simulation scheme for the local perception path as shown in Figure 5.

Suppose the DOINN is trained with $H \times W$ mask-contour pairs. The global perception $\mathcal{G} : \mathbb{R}^{H\times W} \rightarrow \mathbb{R}^{C\times\frac{H}{8}\times\frac{W}{8}}$ can be expressed as

$$\boldsymbol{F}_{\text{gp}} = \mathcal{G}(\boldsymbol{M};\boldsymbol{W}_{\mathcal{R}}), \tag{11}$$

where $\boldsymbol{F}_{\text{gp}} \in \mathbb{R}^{C\times\frac{H}{8}\times\frac{W}{8}}$ is the feature map output of GP path, $\boldsymbol{M} \in \mathbb{R}^{H\times W}$ denotes the after-pooling mask and $\boldsymbol{W}_{\mathcal{R}}$ are trained parameters of the Fourier Unit. Let $\mathcal{G}_{\text{s}} : \mathbb{R}^{sH\times sW} \rightarrow \mathbb{R}^{C\times\frac{sH}{8}\times\frac{sW}{8}}$ be the global perception path which processes a mask $\boldsymbol{M}_{\text{s}} \in \mathbb{R}^{sH\times sW}$ that is $s\times$ larger than tiles used for training. Then each entry of output feature map $\boldsymbol{F}_{\text{s,gp}}$ is defined as

$$\begin{aligned}&\boldsymbol{F}_{\text{s,gp}}[:,i,j] = \mathcal{G}_{\text{s}}(\boldsymbol{M}_{\text{s}};\boldsymbol{W}_{\mathcal{R}})[:,i,j] \\ =&\mathcal{G}(\boldsymbol{M}_{\text{s}}[\frac{mH}{2}:\frac{(m+2)H}{2},\frac{nW}{2}:\frac{(n+2)W}{2}];\boldsymbol{W}_{\mathcal{R}})[:,p,q],\end{aligned} \tag{12}$$

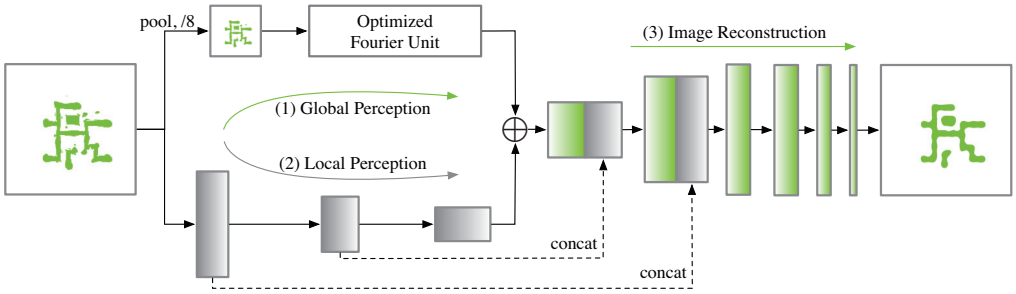

Figure 4: The overall resist image prediction pipeline of the DOINN [20].

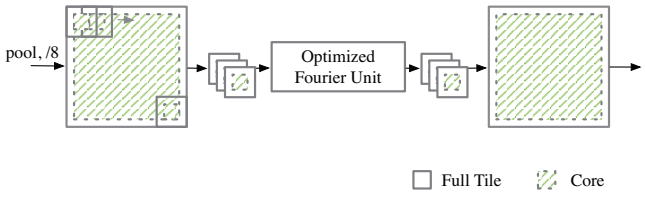

Figure 5: Large tile global perception [20].

where $\frac{d}{2} \leq i < sH - \frac{d}{2}$, $\frac{d}{2} \leq j < sW - \frac{d}{2}$ and

$$m = \lfloor \frac{2(i - \frac{d}{2})}{H} \rfloor, n = \lfloor \frac{2(j - \frac{d}{2})}{W} \rfloor,$$
$$p = ((i - \frac{d}{2}) \mod \frac{H}{2}) + \frac{d}{2},$$
$$q = ((j - \frac{d}{2}) \mod \frac{W}{2}) + \frac{d}{2}. \quad (13)$$

3.1.2 Results To verify the effectiveness of the DOINN model, we show experiments on academic and industrial designs from 14nm to 32nm technology nodes. Statistics of the benchmarks are listed in Table 2 with total number of mask-resist image pairs. We adopt mean pixel accuracy and mean intersection over union to measure the resist image quality.

Definition 1 (Mean Intersection Over Union (mIOU))**.** Given k classes of predicted shapes P_i and their ground truth G_i, $i = 1, 2, ..., k$. The mIOU is defined as

$$\text{mIOU}(P, G) = \frac{1}{k} \sum_{i=1}^{k} \frac{P_i \cap G_i}{P_i \cup G_i}. \quad (14)$$

Definition 2 (Mean Pixel Accuracy (mPA))**.** Given k classes of predicted shapes P_i and their ground truth G_i, $i = 1, 2, ..., k$. The mPA is defined as

$$\text{mPA}(P, G) = \frac{1}{k} \sum_{i=1}^{k} \frac{P_i \cap G_i}{G_i}. \quad (15)$$

Table 3 lists the performance of DOINN on small mask tiles, where the (L) and (H) correspond to 2nm and 1nm pixel size during mask rasterization, respectively. Our approach achieves the best mPA and mIOU compared to two state-of-the-art models.

Table 4 shows the effectiveness of large tile simulation scheme, which attains DOINN good quality when dealing with large tiles, while the original model exhibits great performance degradation.

Table 2: Details of the Dataset.

Dataset	Train	Test	Tile Size	Litho Engine
ICCAD-2013	4875	10	$4\mu m^2$	Lithosim [18]
ISPD-2019	10300	11641	$4\mu m^2$	Calibre [21]
ISPD-2019-LT	-	10	$64\mu m^2$	Calibre [21]
N14	1630	137	$4\mu m^2$	-

Table 3: Result Comparison with State-of-the-Art.

Benchmark	UNet [11]		DAMO-DLS [10]		Ours	
	mPA (%)	mIOU (%)	mPA (%)	mIOU (%)	mPA (%)	mIOU (%)
ISPD-2019 (L)	99.40	98.03	99.25	98.11	**99.43**	**98.27**
ISPD-2019 (H)	99.08	97.97	-	-	**99.21**	**98.45**
ICCAD-2013 (L)	97.30	95.38	98.94	96.97	**98.98**	**97.79**
ICCAD-2013 (H)	95.16	93.04	-	-	**99.12**	**97.77**
N14	94.39	91.64	-	-	**98.68**	**96.49**

Table 4: Large Tile Simulation Scheme.

ISPD-2019-LT	mPA (%)	mIOU (%)
DOINN	96.30	92.03
DOINN-LT	**99.25**	**98.23**

The large tile simulation scheme also successfully cleaned the random artifacts caused by frequency mismatching (Figure 6).

3.2 AI for Mask Optimization

In the second study, we focus on the mask optimization problem, covering the carefully designed CFNO-backboned network and the litho-guided self training algorithm [19]. Mask optimization (MO) is a problem to find a proper mask $\boldsymbol{M}$ associated with a design $\boldsymbol{Z_t}$, such that the difference between the resist image $\boldsymbol{Z}$ after the forward lithography modeling and the design is minimized. In literature, there are many evaluation metrics used to estimate the quality of mask optimization solutions. Well accepted ones are edge-displacement-error (EPE) violations, mean square error (MSE) and process variation band (PVB) area. EPE and PVB are depicted in Figure 7.

Definition 3 (EPE Violation[18])**.** EPE is measured as the geometric distance between the target edge and the lithographic contour

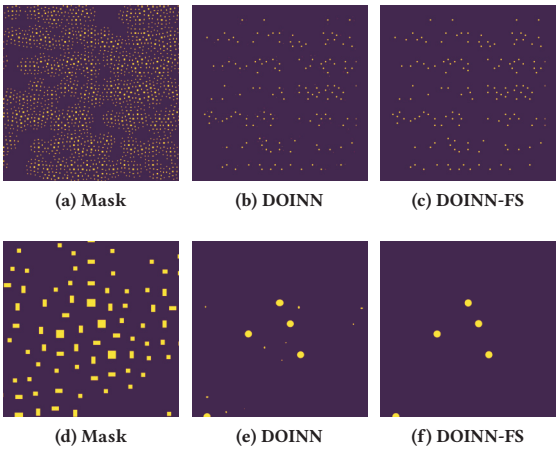

Figure 6: Visualization of large tile simulation. (a)-(c) are input mask, contour prediction with default DOINN and contour prediction with large tile simulation scheme. (d)-(f) are partial zoom-in view at the same location of (a)-(c) respectively [20].

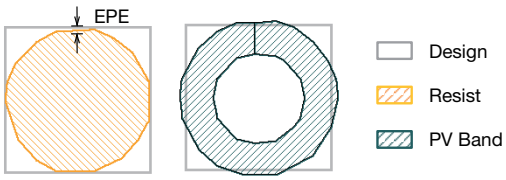

Figure 7: Mask quality measurements [19].

printed at the nominal condition. If the EPE measured at a point is greater than certain tolerance value, we call it an EPE violation.

Definition 4 (MSE)**.** MSE measures the pixel-wise difference between the design and the resist image as in:

$$\text{MSE} = ||Z - Z_t||_F^2. \tag{16}$$

Definition 5 (PVB Area[18])**.** This is evaluated by running lithography simulation at different corners on the final mask solution. Once run, a process variation band metric will be defined as the XOR of all the contours. The total area of the process variation band is defined as PVB Area.

3.3 CFNO for Mask Optimization

The network architecture for mask optimization is shown in Figure 8, with three CFNO units with different token size for multi-scale token embedding plus a stacked convolution path. The embedding of four learning paths will then be aggregated and feed into a series of convolution and transposed convolution layers to generate masks. Compared to DOINN (1.3M), such structure comes with smaller patch size, smaller model size (0.4M) and hence more efficient computing.

3.4 Litho-Guided Self Training

In some early experiments, we observe that our model can produce even better masks than the original training data in terms of EPE

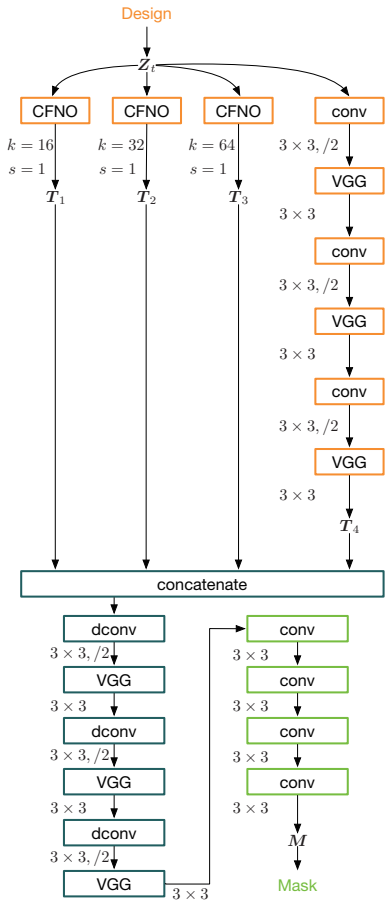

Figure 8: The structure of the CFNO-backboned mask optimizer [19]. conv and dconv represent convolution and transposed convolution layers. VGG denotes a stacked convolution block as proposed in [22]. 3×3 indicates the convolution kernel size and $/2$ represents a stride of 2. k, s define the layout token size and the token-wise convolution kernel size, respectively.

violation and PVB Area. We refer readers to [19] for more detailed analysis and discussion. Here we directly introduce the *litho-guided self training* (LGST) as in Algorithm 1. The first step is to train the machine learning model with the initial training set (line 1), where masks are generated from the ILT engine. Following steps are T rounds LGST (lines 2–12). In each LGST round, we perform model inference on the training set and obtain the model generated masks (line 4). Both the ML-Mask and ILT-Mask will be fed into the lithography simulation engine to measure the resist quality (lines 5–6). Here we use MSE as a example (see definition 4). If the machine learning generated mask has better resist quality than the ILT created mask, we will replace it in the training set (lines 7–9). At the end of T rounds LGST, we will retrain the model with latest training set.

3.5 Results

We evaluate the mask optimization solution on both via and metal layer designs, where the EPE violation count is significantly reduced by our approach compared to two state-of-the-art numerical

Table 5: Result comparison with state-of-the-art.

Metal	levelsetGPU [1]				A2-ILT [23]				Ours			
	MSE	EPE #	PVB	Score	MSE	EPE #	PVB	Score	MSE	EPE #	PVB	Score
Metal	709293.8	139.6	1020105.9	4778423.6	589664.8	128.8	1147425.6	5233702.4	591714.8	**45.6**	1126395.6	4733582.4
Via	645658.6	165.2	401970	2433880	624754.8	288.5	491023.9	3406595.6	335335.2	**2.7**	455712.8	1836351.2

Algorithm 1 Litho-Guided Self Training.

Input: Training dataset $\{\mathcal{Z}_{tr}, \mathcal{M}_{tr}\}$, LGST max iteration T, a random initialized machine learning mode $f(\cdot; \boldsymbol{w})$ and a lithography simulator $l(\cdot)$;

Output: Trained model $f(\cdot; \boldsymbol{w})$ and updated training set $\{\mathcal{Z}_{tr}, \mathcal{M}_{tr}\}$.

$\boldsymbol{w} \leftarrow$ Train f with $\{\mathcal{Z}_{tr}, \mathcal{M}_{tr}\}$;
for $t = 1, 2, ..., T$ **do**
  for each $Z^*_{tr,i} \in \mathcal{Z}_{tr}$ **do**
    $\tilde{M}_{tr,i} \leftarrow f(Z^*_{tr,i}; \boldsymbol{w})$;
    $\mathbf{MSE}_{ml} \leftarrow l(\tilde{M}_{tr,i}, Z^*_{tr,i})$;
    $\mathbf{MSE}_{ilt} \leftarrow l(M_{tr,i}, Z^*_{tr,i})$;
    if $\mathbf{MSE}_{ml} < \mathbf{MSE}_{ilt}$ **then**
      $\mathcal{M}_{tr} \leftarrow$ Replace $M_{tr,i}$ with $\tilde{M}_{tr,i}$;
  $\boldsymbol{w} \leftarrow$ Train f with $\{\mathcal{Z}_{tr}, \mathcal{M}_{tr}\}$;

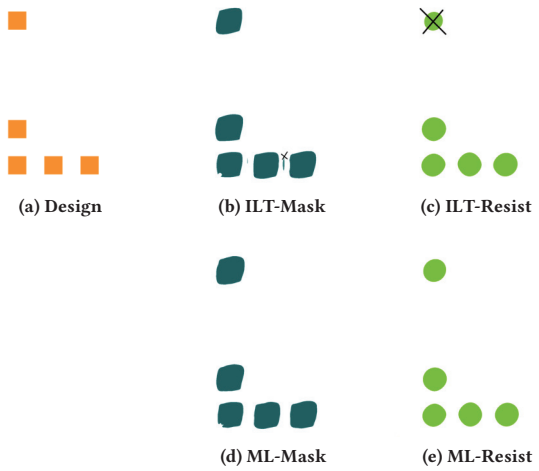

Figure 9: Machine learning can do better on mask optimization tasks [19]. (a) Part of a design containing via arrays. (b) Mask generated by the levelset ILT engine. (c) Nominal resist image from the ILT-Mask. (d) Mask generated by the machine learning model. (e) Nominal resist image from the ML-Mask.

solutions. Particularly, the results are generated by single inference without legacy tool finetuning (Table 5).

4 Conclusion

In this paper, we discuss recent advances of machine solution in computational lithography problems. We argue that these solutions are challenged by lacking in data problem, which can be addressed by bringing more physics to bias the machine learning model. Therefore, we propose a physics-inspired backbone structure called CFNO, which can be plugged into most neural network designs. We show CFNO can significantly outperform state-of-the-art literature on two case studies. We hope this paper can motivate continuous research on physics-inspired machine learning solutions to various design for manufacturability problems.

References

[1] Z. Yu, G. Chen, Y. Ma, and B. Yu, "A GPU-enabled level set method for mask optimization," in *Proc. DATE*, 2021.
[2] J.-R. Gao, X. Xu, B. Yu, and D. Z. Pan, "MOSAIC: Mask optimizing solution with process window aware inverse correction," in *Proc. DAC*, 2014, pp. 52:1–52:6.
[3] "ITRS," http://www.itrs.net.
[4] H. Yang, J. Su, Y. Zou, Y. Ma, B. Yu, and E. F. Y. Young, "Layout hotspot detection with feature tensor generation and deep biased learning," *IEEE TCAD*, vol. 38, no. 6, pp. 1175–1187, 2019.
[5] H. Yang, S. Li, C. Tabery, B. Lin, and B. Yu, "Bridging the gap between layout pattern sampling and hotspot detection via batch active learning," *IEEE TCAD*, 2021.
[6] H. Geng, H. Yang, L. Zhang, J. Miao, F. Yang, X. Zeng, and B. Yu, "Hotspot detection via attention-based deep layout metric learning," in *Proc. ICCAD*, 2020.
[7] R. Chen, W. Zhong, H. Yang, H. Geng, X. Zeng, and B. Yu, "Faster region-based hotspot detection," in *Proc. DAC*, 2019, pp. 146:1–146:6.
[8] W. Ye, M. B. Alawieh, Y. Lin, and D. Z. Pan, "LithoGAN: End-to-end lithography modeling with generative adversarial networks," in *Proc. DAC*, 2019, pp. 107:1–107:6.
[9] H. Yang, S. Li, Z. Deng, Y. Ma, B. Yu, and E. F. Y. Young, "GAN-OPC: Mask optimization with lithography-guided generative adversarial nets," *IEEE TCAD*, 2020.
[10] G. Chen, W. Chen, Y. Ma, H. Yang, and B. Yu, "DAMO: Deep agile mask optimization for full chip scale," in *Proc. ICCAD*, 2020.
[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241.
[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.
[13] H. Yang, P. Pathak, F. Gennari, Y.-C. Lai, and B. Yu, "DeePattern: Layout pattern generation with transforming convolutional auto-encoder," in *Proc. DAC*, 2019, pp. 148:1–148:6.
[14] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.
[15] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.
[16] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier neural operator for parametric partial differential equations," in *Proc. ICLR*, 2021.
[17] X. Ma and G. R. Arce, *Computational lithography*. John Wiley & Sons, 2011, vol. 77.
[18] S. Banerjee, Z. Li, and S. R. Nassif, "ICCAD-2013 CAD contest in mask optimization and benchmark suite," in *Proc. ICCAD*, 2013, pp. 271–274.
[19] H. Yang, Z. Li, K. Sastry, S. Mukhopadhyay, A. Anandkumar, B. Khailany, V. Singh, and H. Ren, "Large scale mask optimization via convolutional fourier neural operator and litho-guided self training," *arXiv preprint arXiv:2207.04056*, 2022.
[20] H. Yang, Z. Li, K. Sastry, S. Mukhopadhyay, M. Kilgard, A. Anandkumar, B. Khailany, V. Singh, and H. Ren, "Generic lithography modeling with dual-band optics-inspired neural networks," in *Proc. DAC*, 2022.
[21] "Calibre," https://eda.sw.siemens.com/en-US/ic/calibre-design/.
[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[23] Q. Wang, B. Jiang, M. D. Wong, and E. F. Young, "A2-ILT: GPU accelerated ILT with spatial attention mechanism," in *Proc. DAC*, 2022.