WHERE **INNOVATION** BEGINS

DESIGN AUTOMATION CONFERENCE

JULY 9–13, 2023

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

WHERE INNOVATION BEGINS
DESIGN AUTOMATION CONFERENCE

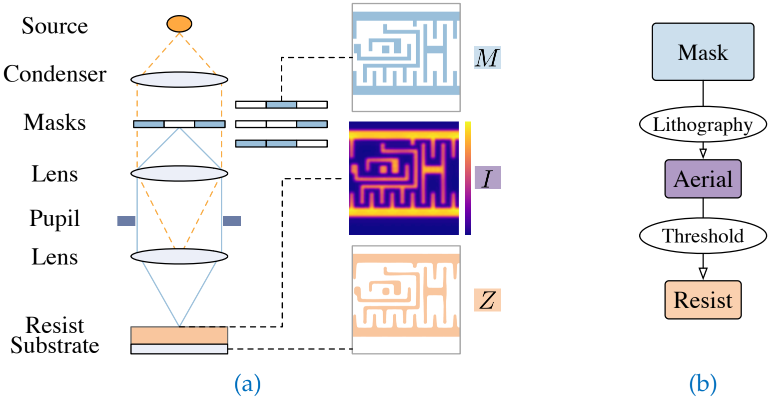
Physics-Informed Optical Kernel Regression Using Complex-valued Neural Fields

Guojin Chen[1], Zehua Pei[1], Haoyu Yang[2], Yuzhe Ma[3], Bei Yu[1], Martin D. F. Wong[1]

[1]CUHK [2]nVIDIA [3]HKUSZ(GZ)

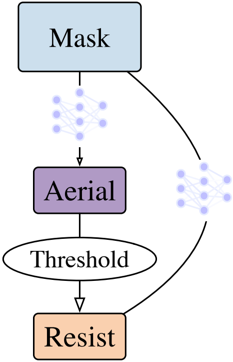
Outline

Micro- and Nanolithography

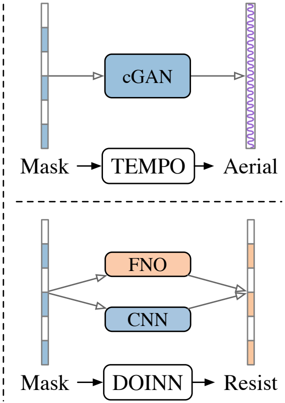


(a) Components of the lithography imaging system: illumination source, lenses, and pupil. (b) Lithography simulation flow using source- and pupil-dependent optical kernels.

Summary of previous works

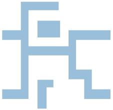
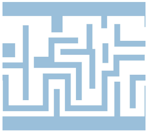
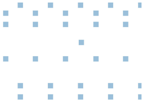


General flow of previous SOTA.

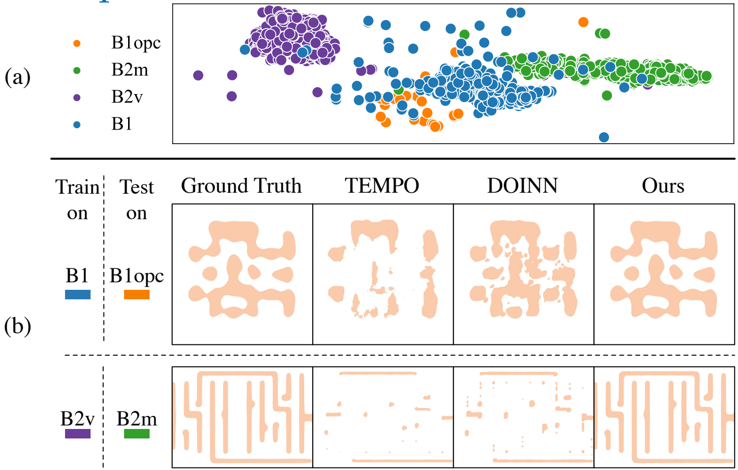


Previous SOTA work on Aerial stage (TEMPO [Ye+20]) and Resist stage (DOINN [Yan+22]).

Dataset Samples

Name	(a) B1	(b) B2m	(c) B2v
Mask			
Wafer			

Drawbacks of previous works



(a) t-SNE distribution of datasets.
(b) Comparison of generalization capability on out-of-distribution (OOD) datasets.

Drawbacks of previous works

Previous image-learning based simulator.

- ✘ ☹ Bias on image distribution.
  - Can not generalize on different layers.
  - Performance is sensitive to dataset distribution.
- ✘ ☹ Large models.
  - Needs more parameters to remember the different distribution on higher resolution images.
  - Needs to train new models on new datasets.
- ✓ ☺ Fast prediction.

Industrial rigorous lithography simulator.

- ✓ ☺ Can work on different layer types. Good generalization capability.
- ✓ ☺ The imaging models can be pre-calculated and stored as kernels and coefficients.
- ✘ ☹ Computationally expensive.

Recap on rigorous lithography model

Hopkins Model and Transmission Cross-Coefficient (TCC)

The imaging equation:

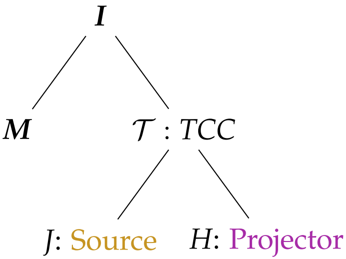
$$\mathcal{F}(\boldsymbol{I})(f,g) = \iint_{-\infty}^{\infty} \underline{\mathcal{T}\left((f'+f,g'+g),(f',g')\right)}\,\mathcal{F}(\boldsymbol{M})\left(f'+f,g'+g\right)\mathcal{F}(\boldsymbol{M})^{*}\left(f',g'\right)\mathrm{d}f'\mathrm{d}g', \tag{1}$$

where $\boldsymbol{M}$ is the mask, (f,g) is its frequencies. $\mathcal{T}$ is TCC given by:

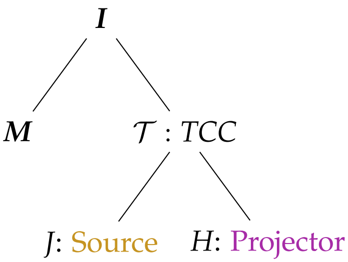
$$\mathcal{T}\left((f',g'),(f'',g'')\right) := \iint_{-\infty}^{\infty} \underline{\mathcal{F}(J)(f,g)}\,\mathcal{F}(H)\left(f+f',g+g'\right)\mathcal{F}(H)^{*}\left(f+f'',g+g''\right)\mathrm{d}f\,\mathrm{d}g, \tag{2}$$

where the weight factor J solely depends on effective source, H is projector transfer function.

Computation graph of aerial image



Computation graph of aerial image



When the projector and source are fixed,

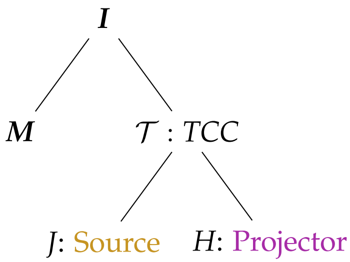
J: constant matrix

H: constant matrix

$\Downarrow$

$\mathcal{T}$: TCC is a constant matrix

Computation graph of aerial image



When the projector and source are fixed,

J: constant matrix	H: constant matrix

$\Downarrow$

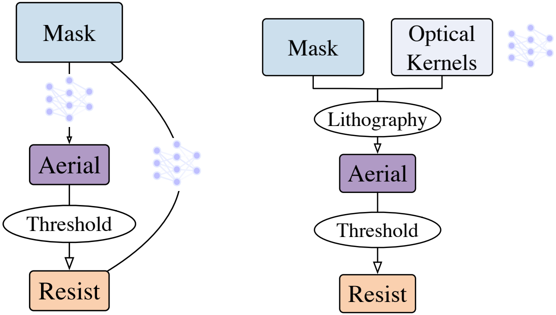
$\mathcal{T}$: TCC is a constant matrix

Instead of learning an image-to-image mapping,

Would it be Possible to learn the **TCC optical kernels**?

The benefits of learning optical kernels

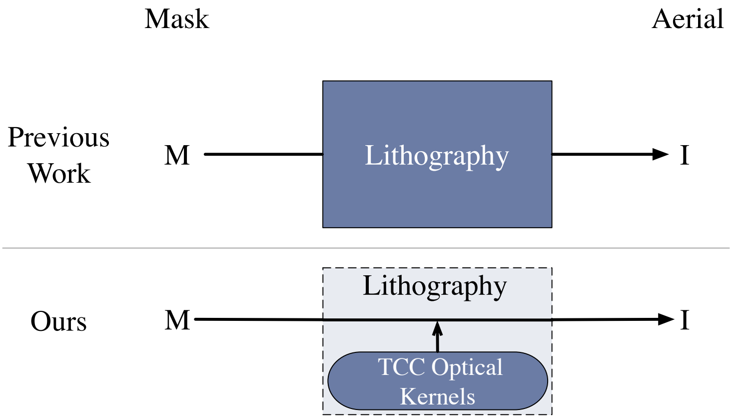
- Get rid of negative influence of layer types & dataset distribution.
- Less training data required & smaller model size.



(a) Previous (b) Ours

(a) Previous image-learning based methods. (b) Ours.

Our approach



The obstacles of learning optical kernels

Dataset : **no ground truth**, need to design the optical kernels.

TCC : are in **frequency domain**, need to support complex-valued computations.

We need to learn something with **no ground truth, no prior-knowledge** about the data structure and dimensions.

The solutions of learning optical kernels

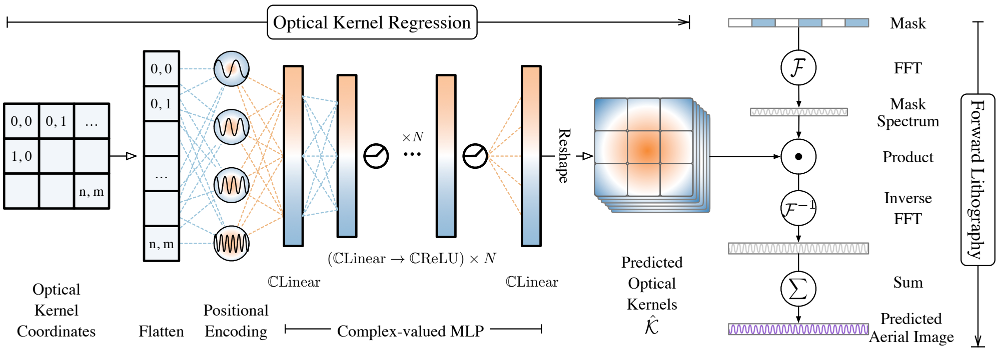
What to learn : Design the kernel dimension based on physical *"resolution limit"*.

How to learn :

Network : Implement a set of differentiable complex-valued neuron layers.

Training : A new training paradigm separates the influence of masks and optical kernels

Nitho Framework



The overall aerial image prediction pipeline of Nitho framework, which separates mask-related linear operations from optical kernel regression using coordinate-based ℂMLP.

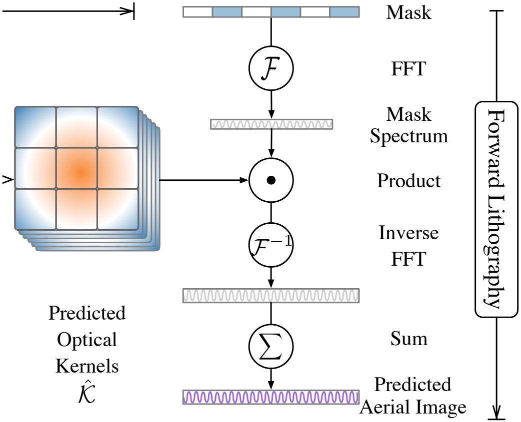
Sum of Coherent Sources Approach (SOCS)

SOCS

$$\boldsymbol{I} = \sum_{i=1}^{r} \alpha_i \left| \mathcal{F}^{-1} \left(\mathcal{F}\left(\boldsymbol{h}_i\right) \odot \mathcal{F}(\boldsymbol{M}) \right) \right|^2 .$$

$$\Downarrow$$

$$\boldsymbol{I} = \sum_{i}^{r} \left| \mathcal{F}^{-1} \left(\mathcal{K}_i \odot \mathcal{F}(\boldsymbol{M}) \right) \right|^2 , \qquad (3)$$



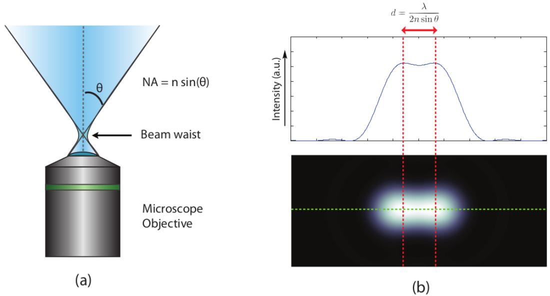
Optical kernel regression

Q1: How to design the kernel dimension

"Resolution limit"

Q1: How to design the kernel dimension

"Resolution limit"



(a) Illustration of the numerical aperture (NA) of a microscope objective, (b) Two points are blurred by diffraction, which results in a limited resolution. The smallest resolvable distance between two points with an optical technique is limited by $d = \lambda/(2n \sin\theta)$

Resolution limit

Smallest feature and resolution.

$$\mathrm{d} = \frac{\lambda}{2n\sin\theta} = \frac{\lambda}{2\mathrm{NA}}, R = \frac{1}{d} = \frac{2NA}{\lambda}$$

Using this description, the kernel width and height can be set as:

$$m = (W \times \frac{2\mathrm{NA}}{\lambda}) \times 2 + 1, n = (H \times \frac{2\mathrm{NA}}{\lambda}) \times 2 + 1, \quad (4)$$

where we use one-pixel width/height to represent $1nm$, the mask pitch can be replaced by mask image width W, height H.

Optical Kernel Dimension

$$\mathcal{K} \in \mathbb{C}^{r \times n \times m} \quad (5)$$

SOCS approximation

Since the eigenvalues α_i in Equation (3) rapidly decay in magnitude, truncating the summation at order r can be a decent approximation with error bounds proven in [Pat+94].
So the SOCS can be approximated as:

$$\boldsymbol{I} = \sum_{i}^{r} \left| \mathcal{F}^{-1} \left(\mathcal{K}_i \odot \mathcal{F}(\boldsymbol{M}) \right) \right|^2, \tag{6}$$

where $\mathcal{K}_i$ is the i-th optical kernel, r is the total number of kernels.

Discussion about kernel size

$$\mathcal{K} \in \mathbb{C}^{r \times n \times m} \tag{7}$$

Given commonly used:

$$\lambda = 193nm, NA = 1.35, \tag{8}$$

We have:

$$m \approx 0.028 * W, n \approx 0.028 * H \tag{9}$$

In our settings: $r < 60$.
Previous image-learning based space $\mathbb{R}^{C \times W \times H}$. VS. Our space $\mathbb{C}^{r \times n \times m}$.

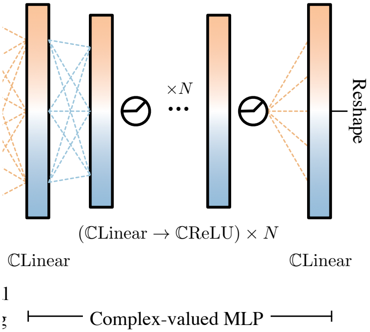
$$\mathbb{R}^{C \times W \times H} \gg \mathbb{C}^{r \times n \times m} \tag{10}$$

ℂMLP

The ℂMLP is further constructed as,

$$\mathbb{C}\text{MLP} : \mathbb{C}\text{Linear} \rightarrow (\mathbb{C}\text{Linear} \rightarrow \mathbb{C}\text{ReLU}) \times N \ldots \rightarrow \mathbb{C}\text{Linear}, \quad (11)$$

where $\times N$ means there are N hidden blocks ($\mathbb{C}\text{Linear} \rightarrow \mathbb{C}\text{ReLU}$).



The solutions of learning optical kernels

What to learn ✓: Design the kernel dimension based on physical "*resolution limit*".

Network ✓: Implement a set of differentiable complex-valued neuron layers.

Training ☐ : A new training paradigm separates the influence of masks and optical kernels

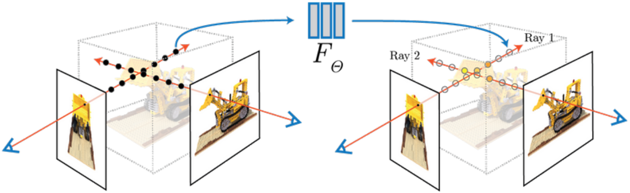
Unresolved challenges:

- No-ground truth for optical kernels
  - How to define the input, output.
  - What kind of network to use.
- How to train the network.

NeRF

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis [Mil+20]

Ben Mildenhall*	Pratul Srinivasan*	Matt Tancik*	Jon Barron	Ravi Ramamoorthi	Ren Ng
					
UC Berkeley	UC Berkeley	UC Berkeley	Google Research	UC San Diego	UC Berkeley



NeRF: problem settings

Problem defination

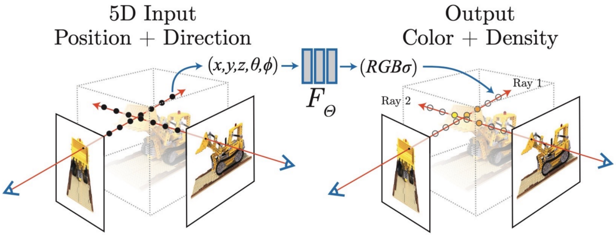
- Given a dataset containing RGB images of a static scene, their corresponding camera poses, and intrinsic parameters,
- Predict the color and volume density for every viewing location and direction.

Inputs:

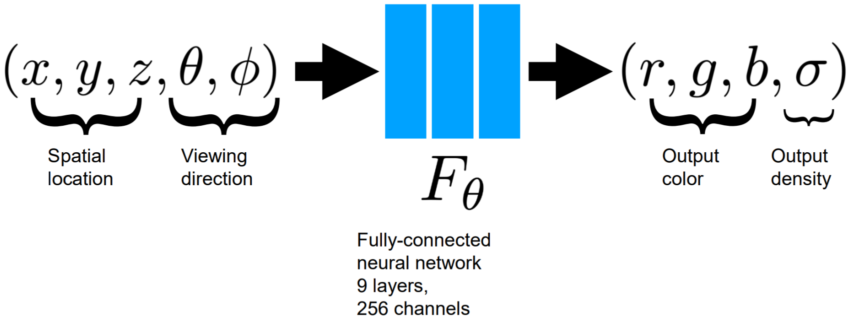
- x, y, z: Target position.
- θ, ϕ: Target orientation.

Outputs:

- $c = (r, g, b)$: Color.
- σ: Volume density.



NeRF: coordinates-based networks



Nitho: NeRF inspired lithography simulator.

The lithography conditions are location dependent.

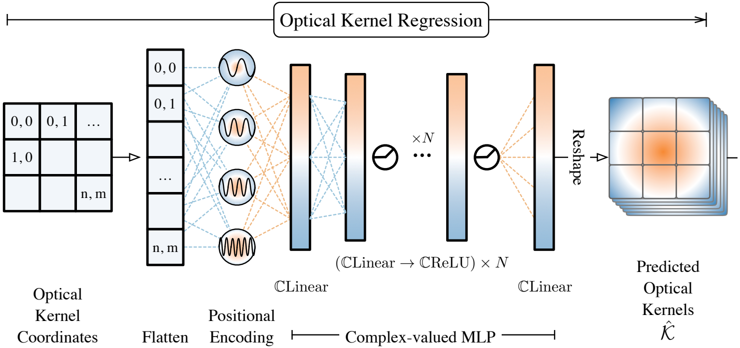
TCC is given by:

$$\mathcal{T}\left((f',g'),(f'',g'')\right) := \iint_{-\infty}^{\infty} \underline{\mathcal{F}(J)(f,g)}\,\underset{\sim}{\mathcal{F}(H)\left(f+f',g+g'\right)\mathcal{F}(H)^{*}\left(f+f'',g+g''\right)}\mathrm{d}f\,\mathrm{d}g, \quad (12)$$

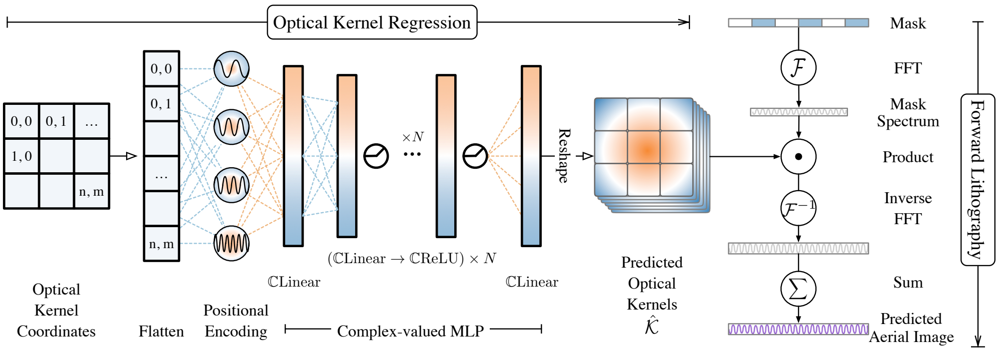
Nitho: design

Inputs : the coordinates of TCC spectrum.

Outputs : TCC values



Forward training paradigm



The overall aerial image prediction pipeline of Nitho framework, which separates mask-related linear operations from optical kernel regression using coordinate-based ℂMLP.

Comparison with SOTA

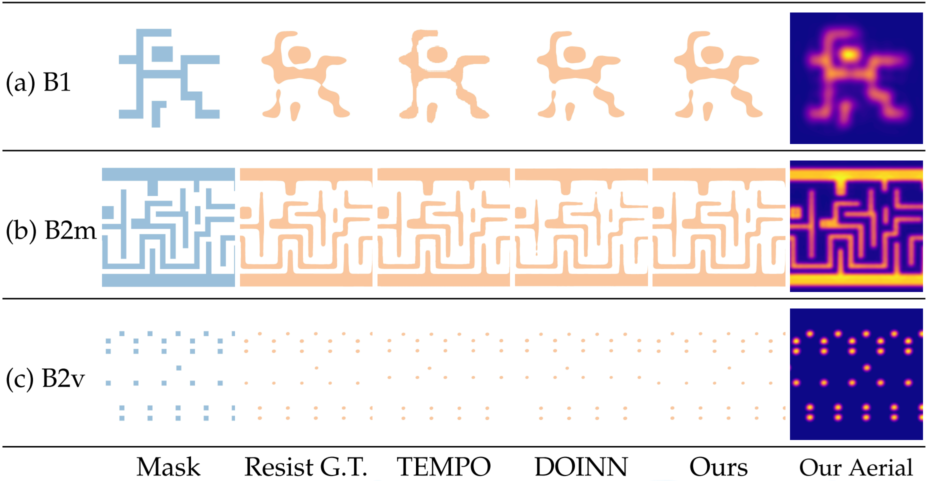
Table: Result Comparison with State-of-the-Art.

Bench	Aerial Image									Resist Image					
	TEMPO [Ye+20]			DOINN [Yan+22]			Nitho			TEMPO* [Ye+20]		DOINN* [Yan+22]		Nitho	
	MSE $\times 10^{-5}$	ME $\times 10^{-2}$	PSNR dB	MSE $\times 10^{-5}$	ME $\times 10^{-2}$	PSNR dB	MSE $\times 10^{-5}$	ME $\times 10^{-2}$	PSNR dB	mPA (%)	mIOU (%)	mPA (%)	mIOU (%)	mPA (%)	mIOU (%)
B1	108.29	10.49	32.01	5.55	1.94	47.10	**1.32**	**0.51**	**50.75**	94.60	88.70	99.19	98.32	**99.45**	**99.21**
B2m	1899.04	13.96	30.77	1202.39	6.11	31.64	**25.48**	**0.82**	**49.06**	98.24	96.55	98.79	97.10	**99.15**	**99.02**
B2v	6.54	3.86	42.76	2.26	2.75	46.37	**2.01**	**0.68**	**48.06**	99.06	93.28	99.21	98.41	**99.59**	**99.34**
B2m + B2v	4352.25	15.21	27.10	3114.24	12.35	29.92	**33.13**	**0.78**	**47.88**	98.63	95.84	98.71	96.68	**99.61**	**99.36**
Average	1591.53	10.88	33.16	1081.11	5.79	39.26	**15.49**	**0.70**	**48.94**	97.63	93.59	98.98	97.63	**99.45**	**99.23**
Ratio	102.77	15.55	0.68	69.81	8.27	0.80	**1.00**	**1.00**	**1.00**	0.98	0.94	0.99	0.98	**1.00**	**1.00**

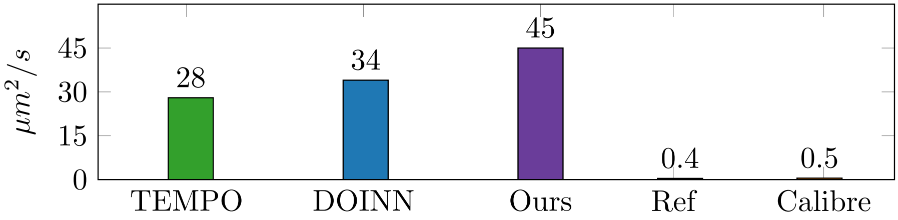
* Models are re-trained using resist image dataset with an amendment to the final activation layer.

Visualization

Visualization of the results of Nitho in aerial and resist stage.



Runtime comparison



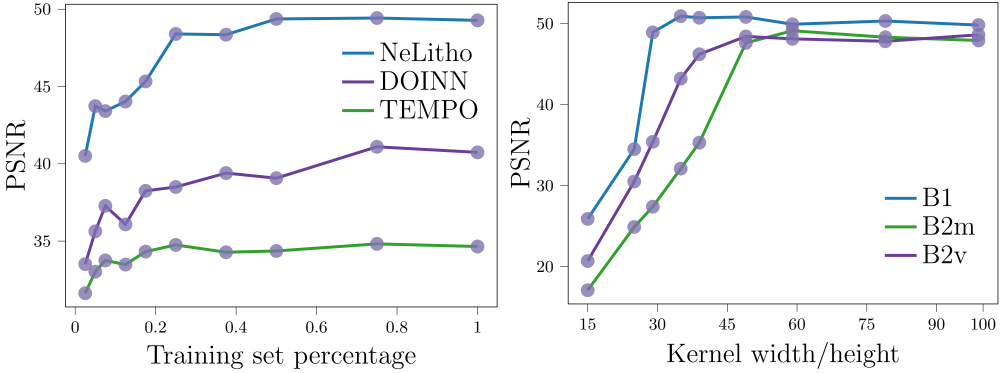
Runtime comparison with SOTA.

Comparison on out-of-distribution (OOD) datasets.

Table: Comparison with SOTA on out-of-distribution dataset.

Benchmark Train on	Benchmark Test on	TEMPO [Ye+20] mPA %	TEMPO [Ye+20] mIOU %	DOINN [Yan+22] mPA %	DOINN [Yan+22] mIOU %	Nitho mPA %	Nitho mIOU %
B1	B1opc	90.25	86.15	98.03	94.76	99.43	99.17
Drop		↓ 4.35	↓ 2.55	↓ 1.16	↓ 3.56	↓ 0.02	↓ 0.04
B2m	B2v	99.40	71.86	99.64	78.31	99.58	97.33
Drop		↑ 0.34	↓ 21.42	↑ 0.43	↓ 20.10	↓ 0.01	↓ 2.01
B2v	B2m	66.06	55.82	76.43	68.73	98.08	97.18
Drop		↓ 32.18	↓ 40.73	↓ 22.36	↓ 28.37	↓ 1.07	↓ 1.84
Average		85.24	71.28	91.36	80.60	**99.03**	**97.90**
Avg. Drop		↓ 12.06	↓ 21.57	↓ 7.70	↓ 17.34	↓ **0.37**	↓ **1.29**

Ablation study on smaller training sets and kernels sizes



(a) (b)

(a) Comparison with SOTA on smaller training sets. (b) Ablation study on kernel size on different datasets.

WHERE INNOVATION BEGINS
DESIGN AUTOMATION CONFERENCE

THANK YOU!

NVIDIA