DAMO: Deep Agile Mask Optimization for Full Chip Scale

Guojin Chen, Wanli Chen, Yuzhe Ma, Haoyu Yang, Bei Yu

The Chinese University of Hong Kong

Short bio of Guojin Chen

Guojin Chen

The Chinese University of Hong Kong.

I am an MSc student at The Chinese University of Hong Kong (CUHK) and broadly study foundational topics and applications in machine (sometimes deep) learning in VLSI and optimization, including reinforcement learning, computer vision. I am advised by Prof. Bei Yu. I received my Bachelor Degree of Computer Science from Huazhong University of Science and Technology.

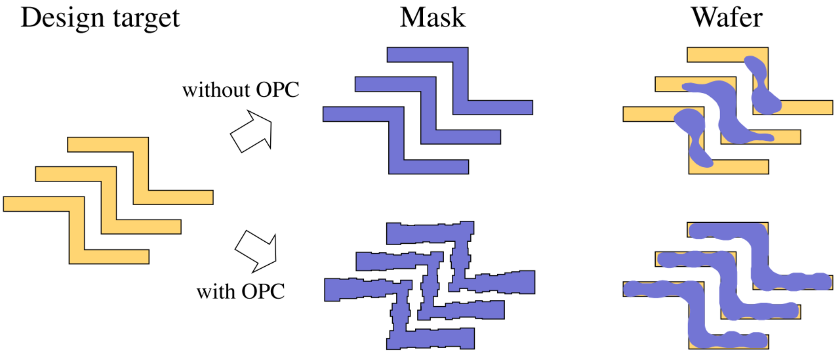
Outline

Outline

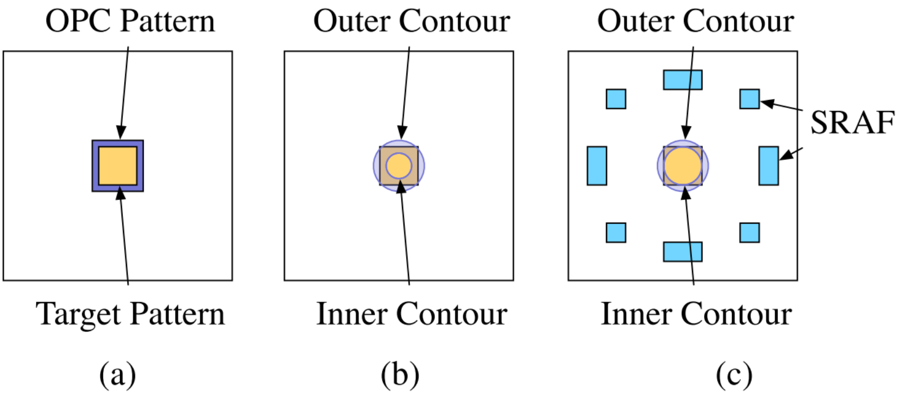
Background and problem formulation

Project backgroud

Optical proximity correction (OPC) is a photolithography enhancement technique commonly used to compensate for image errors due to diffraction or process effects.



PRELIMINARIES of OPC: DESIGN, SRAF, MASK, WAFER

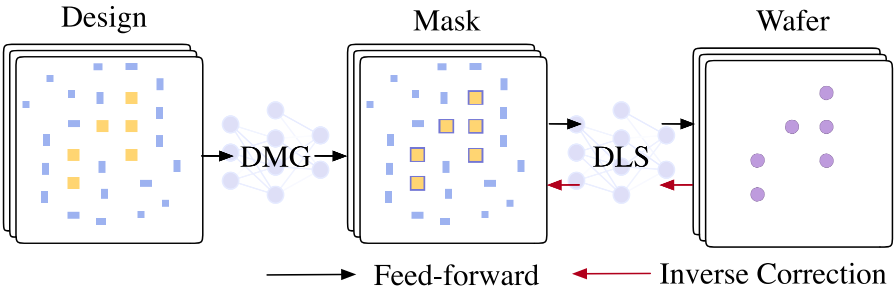


Problem definition: Given a design image $\vec{w}$, the objective of mask optimization is generating the corresponding mask $\vec{x}$ such that remaining patterns $\vec{y}$ after lithography process is as close as $\vec{w}$ or, in other words, minimizing PV Band and squared L_2 error of lithography images.

Our DAMO: Deep Agile Mask Optimization forFull Chip Scale

Two main step

OPC and Litho : DMG and DLS

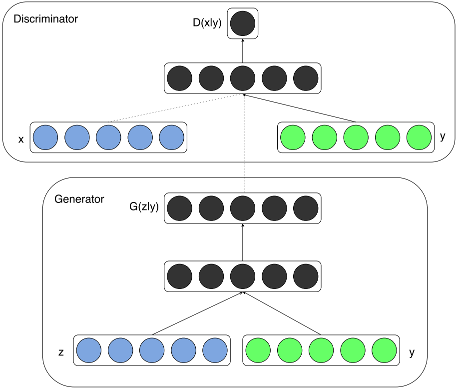


Outline

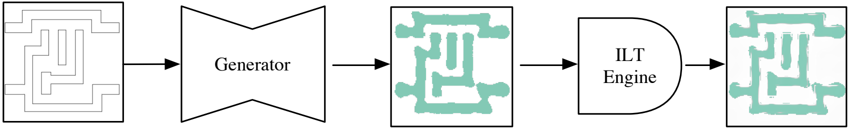
cGAN

Objective function

$$\begin{aligned}&\mathcal{L}_{cGAN}(G, D)\\ =&\mathbb{E}_{x,y}[\log D(x, y)]\\ +&\mathbb{E}_{x,z}[\log(1 - D(x, G(x, z))].\end{aligned} \quad (1)$$

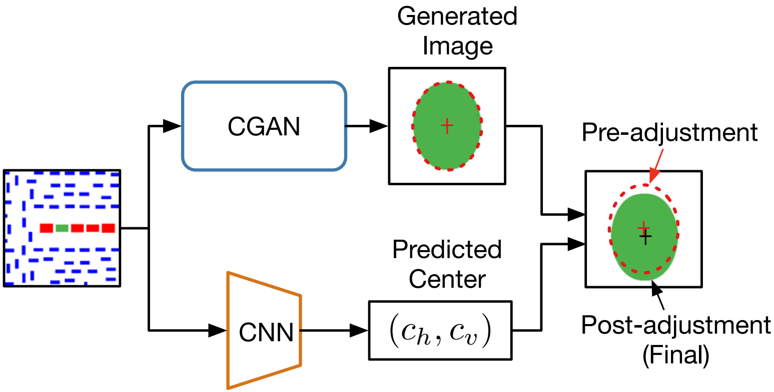


OPC stage previous work: GAN-OPC



GAN-OPC flow: generator inference and ILT refinement.

Litho stage previous work: LithoGAN



LithoGAN Framework.

Issues of previous work.

- Only initial solution, relie on the traditional ILT-model, time-consuming.
- Only targets a single shape within a clip, limited usage in general OPC tasks.
- Only small single clip, low resolution (256 $\times$ 256 pixels).

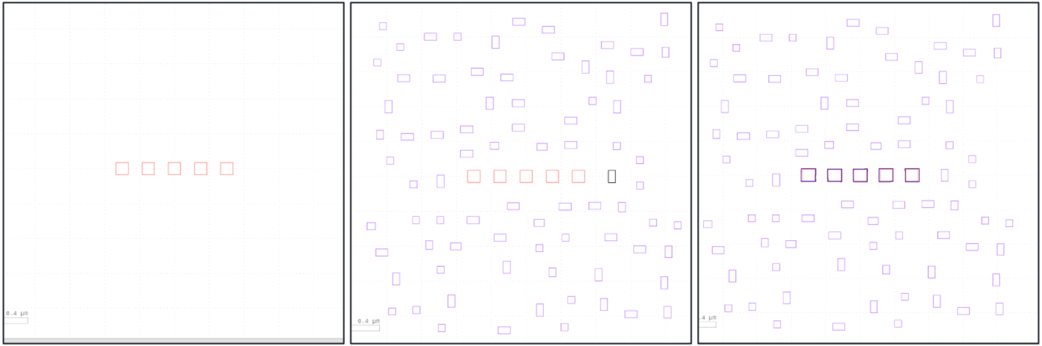
Outline

Solution

Solution: DAMO

- DAMO: End-to-end mask optimization framework without using traditional model.
- DCGAN-HD: High resolution cGAN model.
- Full-chip splitting algorithm for large layout.

Generate Training set



DCGAN-HD: solution for higher resolution

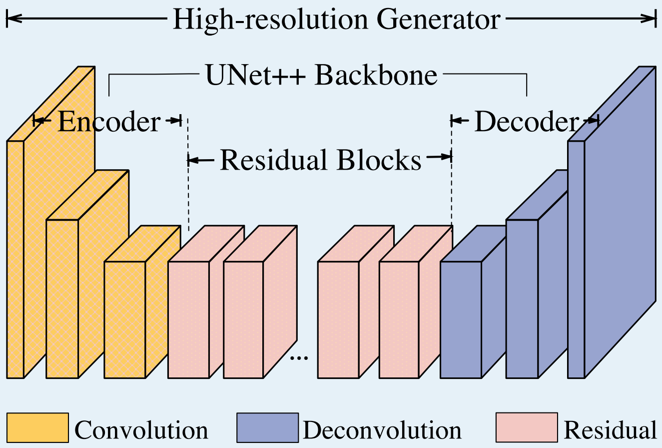
- High-resolution Generator
- Multi-scale discriminator
- Perceptual Loss

High-resolution Generator of DCGAN-HD

Arch.

- UNet++ with Residual blocks.
- High resolution of 1024×1024.

DCUNet++

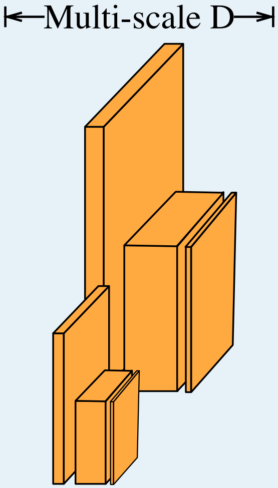


Multi Scale discriminator

Arch.

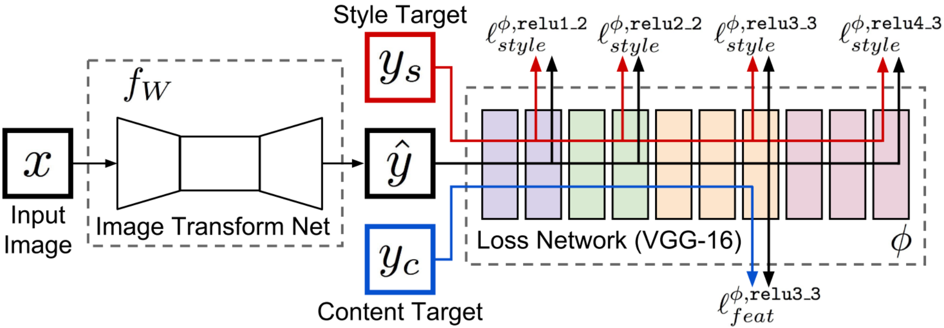
- Two discriminators at different input size, D_1, D_2.
- High resolution of 1024×1024 and 512×512.
- Helps the training of high-resolution model easier.

DCUNet++



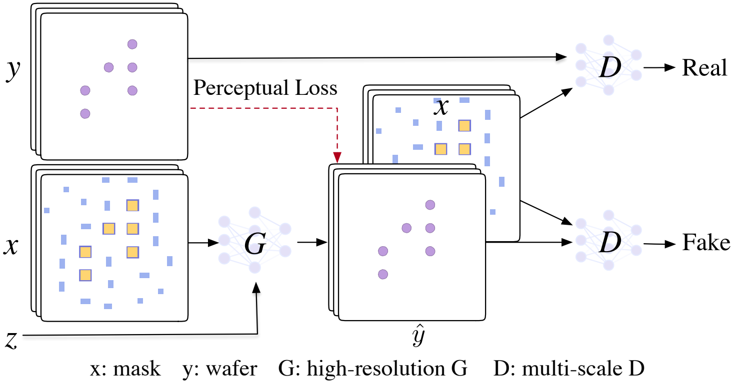
Perceptual Loss

$$\mathcal{L}_{L_P}^{G,\Phi}(\vec{x},\hat{\vec{x}}) = \mathcal{L}_{L_1}(\Phi(\vec{x}),\Phi(\hat{\vec{x}})) = \mathbb{E}_{\vec{x},\hat{\vec{x}}}\left[\|\Phi(\vec{x}) - \Phi(\hat{\vec{x}})\|_1\right], \quad (2)$$



DLS training

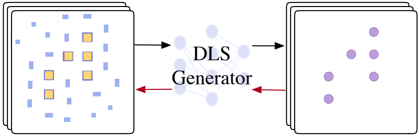
$$\mathcal{L}_{DLS} = \sum_{k=1,2} \mathcal{L}_{cGAN}(G, D_k) + \lambda_0 \mathcal{L}_{L_P}^{G,\Phi}(y, \hat{y}). \quad (3)$$



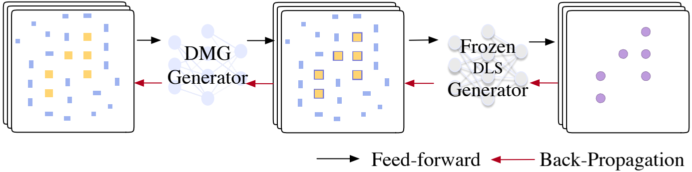
DMG training

$$\mathcal{L}_{DMG} = \sum_{k=1,2} \mathcal{L}_{cGAN}(G_{DMG}, (D_{DMG})_k) + \lambda_1 \mathcal{L}_{L_P}^{G_{DMG},\Phi}(x, \hat{x}). \tag{4}$$

$$\mathcal{L}_{DAMO} = \mathcal{L}_{DMG} + \mathcal{L}_{DLS} + \lambda_2 \mathcal{L}_{L_1}(\hat{y}, w_r). \tag{5}$$



(a)



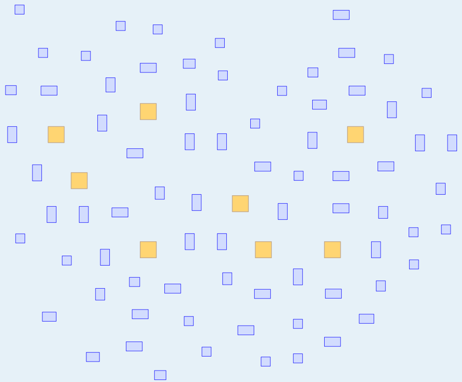
(b)

Full-chip Splitting Algo: Coarse to Fine, DBSCAN to KMeans

Algo. detail

1. DBSCAN then KMeans++
2. Initialize the number of centroids from 1 to V to run KMeans++.
3. Every cluster contains no more than K via patterns.
4. Every via pattern must be contained in a window.
5. If (3) or (4) is not satisfied, increase the centroid number.

Algo. figure

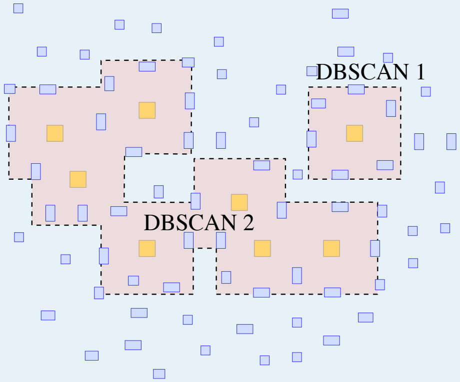


Coarse step, DBSCAN

Algo. detail

DBSCAN algorithm is used to initially detect the clusters of via patterns. After the coarse step, the via patterns in a large layout will be assigned into different DBSCAN clusters.

DBSCAN clusters

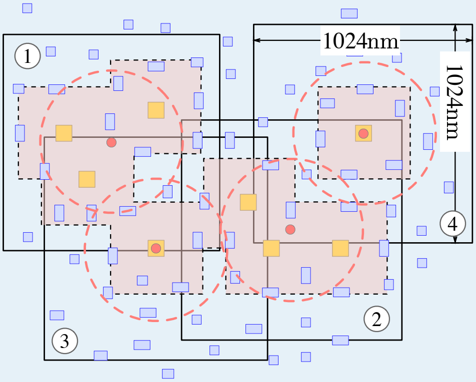


Fine step, KMeans++

Algo. detail

Then we search every coarse cluster and run KMeans++ algorithm to find the best splitting windows.

KMeans clusters

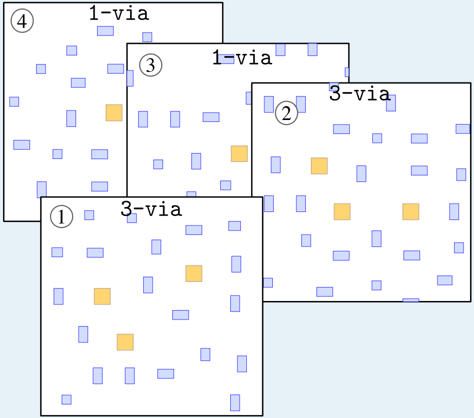


The split chips.

Algo. detail

Every KMeans cluster belongs to a $1024 \times 1024nm^2$ chip, whose center locates at the centroid of the KMeans cluster.

The split chips

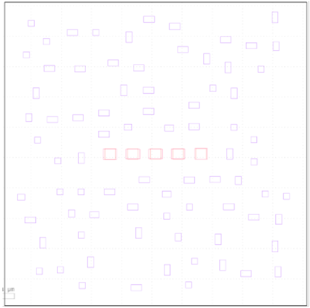


Main Contribution

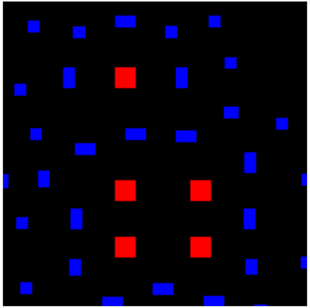
- **DCGAN-HD**: we extend cGANs model by redesign the generator and discriminator for high resolution input (1024*1024), combined with a novel window-splitting algorithm, our model can handle input layout of any size with high accuracy.
- **DLS**: We build up a deep lithography simulator (DLS) based on our DCGAN-HD. Thanks to the express power of stack convolution layers, DLS is expected to conduct lithography simulation faster with similar contour quality compared to legacy lithography simulation process.
- **DAMO**: We present DAMO, a unified end-to-end trainable OPC engine that employs both DLS and DMG to conduct full-chip scale mask optimization without further fine-tune with legacy OPC engines.
- Experimental results show that the proposed DAMO framework is able to output high quality lithography contours more efficiently than Calibre, which also derives $\sim 4\times$ speed-up in OPC tasks while generating masks with even better printability.

Outline

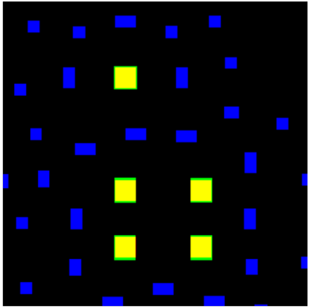
Results on self-generated datasets



Layouts



Design



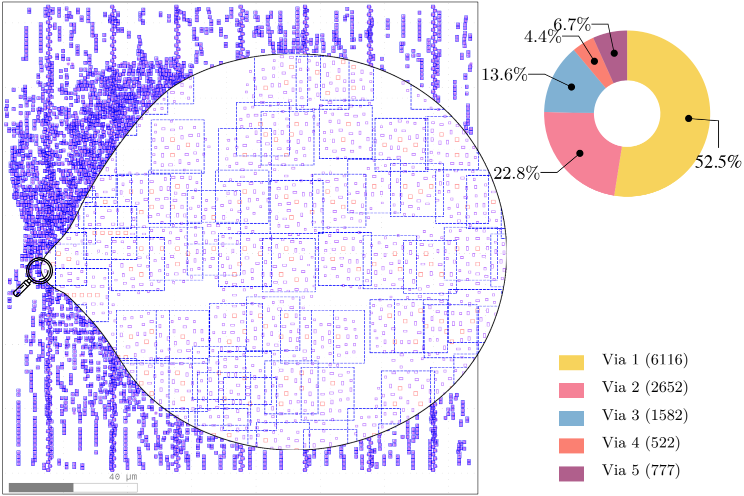
Mask

Results on self-generated datasets

Table 2: Comparison with State-of-the-art on validation set

Bench	case#	GAN-OPC			Calibre			DAMO		
		L_2 (nm^2)	PV Band (nm^2)	runtime (s)	L_2 (nm^2)	PV Band (nm^2)	runtime (s)	L_2 (nm^2)	PV Band (nm^2)	runtime (s)
1-via	500	1464	3064	321	1084	2918	1417	**1080**	**2917**	**284**
2-via	500	4447	6964	336	2161	5595	1406	**2129**	**5576**	**281**
3-via	500	8171	11426	317	3350	8286	1435	**3244**	**8271**	**285**
4-via	500	11659	14958	327	4331	10975	1477	**4263**	**10946**	**291**
5-via	500	15773	18976	318	5410	13663	1423	**5396**	**13640**	**279**
6-via	500	18904	22371	320	6647	15572	1419	**5981**	**15543**	**284**
Average		10069	12960	323	3831	9502	1430	**3682**	**9482**	**284**
Ratio		2.735	1.367	1.138	1.040	1.002	4.427	**1.00**	**1.00**	**1.00**

Results on ISPD 2019 datasets

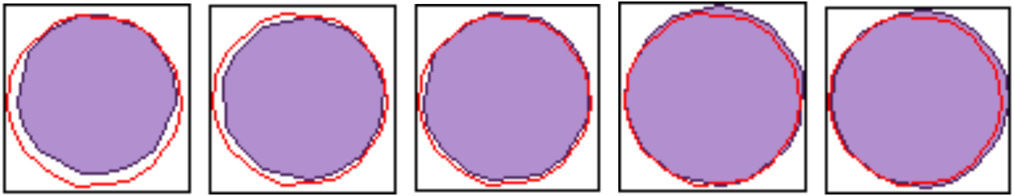


Results on ISPD 2019 datasets

Table 3: Comparison on ISPD 2019 full-chip splitting windows

Bench	case#	GAN-OPC			Calibre			DAMO		
		L_2 (nm^2)	PV Band (nm^2)	runtime (s)	L_2 (nm^2)	PV Band (nm^2)	runtime (s)	L_2 (nm^2)	PV Band (nm^2)	runtime (s)
ISPD-1-via	6116	2367	3492	3963	1073	2857	18959	**1056**	**2848**	**3669**
ISPD-2-via	2652	5412	7126	1742	2232	5670	7537	**2172**	**5654**	**1591**
ISPD-3-via	1582	8792	13047	1021	3602	8276	4494	**3196**	**8127**	**949**
ISPD-4-via	522	12395	15015	341	4395	11051	1692	**4361**	**10987**	**313**
ISPD-5-via	777	16526	19147	495	5526	12305	2537	**4542**	**12251**	**466**
Average		9098	11565	1512	3365	8031	7043	**3065**	**7973**	**1397**
Ratio		2.968	1.451	1.082	1.098	1.007	5.041	**1.00**	**1.00**	**1.00**

Results Visualization



(c) (d) (e) (f) (g)

Visualization of DAMO model advancement on via layer:
(c) Epoch 20; (d) Epoch 40; (e) Epoch 60; (f) Epoch 80; (g) Epoch 100.

Thanks

Thank you.