MODiff: Layout-Guided Mask Optimization via Diffusion Model

Jiale Li[a*], Silin Chen[a*], Yiqing Wang[a], Kangjian Di[a], Li Du[a], and Ningmu Zou[a,b†]

[a]School of Integrated Circuits, Nanjing University, Suzhou, China
[b]Interdisciplinary Research Center for Future Intelligent Chips (Chip-X), Nanjing University, Suzhou, China
[*]These authors contributed equally to this work.
[†]Corresponding Author: nzou@nju.edu.cn.

ABSTRACT

Inverse lithography technology (ILT) faces challenges in computational efficiency and mask regularity. This paper proposes MODiff, a diffusion-based model that generates high-quality masks through a controlled "noise-to-mask" denoising process guided by the target layout. The method incorporates a forward lithography simulator for accuracy and a cosine scheduler for stable training. Evaluated on the ICCAD 2013 dataset, MODiff outperforms state-of-the-art techniques in mask precision and edge smoothness, including on via layers.

Keywords: Inverse lithography, mask optimization, diffusion model

1. INTRODUCTION

With the continuous shrinkage of technology nodes, the mismatch between wafer-printed images and designed graphics has become increasingly pronounced due to the inherent limitations of current lithography systems.[1,2] Optical proximity correction (OPC), one of the critical resolution enhancement techniques (RETs), effectively mitigates these discrepancies by directly adjusting the mask features, thereby reducing the likelihood of hotspot formation.[3] The primary approaches for OPC include rule-based OPC,[1,2] model-based OPC,[4–6] and inverse lithography technology (ILT). Rule-based OPC relies heavily on heuristic experience, often failing to achieve a global optimal solution. Model-based OPC suffers from limited flexibility, constraining the solution space. Inverse lithography technology is a mathematically inverse approach that treats OPC as an inverse imaging problem with pixel-by-pixel mask optimization. Additionally, the level-set method[7,8] provides an implicit alternative. The traditional ILT algorithm require substantial computational resources, and the mask generated is more complex. Fortunately, the explosion of deep learning in recent years has provided more possibilities for mask optimization. Among them, Yang et al.[9] took the lead in applying conditional generation adversarial networks to realize the end-to-end OPC process. Wang[10] explored the integration of spatial attention maps and reinforcement learning architecture to accelerate OPC. For the level-set-based method, Yu et al.[8] used GPU to accelerate the iteration of the level-set function. CircleOpt,[11] as a state-of-the-art method, integrates arc constraints into the inverse ILT flow to generate circle-based masks.

However, mask generation should be treated as a continuous optimization process, whereas most existing models directly produce final results without an iterative refinement stage. The absence of a smooth prediction process often results in unstable model training and discontinuities in the generated masks, which in turn reduce mask accuracy and printability. Although the method based on the level-set function[7,8] adopts the method of iterating level-set function step by step to realize the continuous optimization of the mask, they are highly dependent on the initial choice of the level-set function. Additionally, this model face challenges in incorporating sub-resolution assist features (SRAFs), which are commonly used to enhance mask quality and achieve higher precision optimization.[12] As shown in Fig. 1, We can find that GAN-OPC and Neural-ILT are prone to violations at the edge of the mask (marked in black boxes). GAN-OPC is more obvious, with a large number of burrs made up of discrete pixels at the edges. Although the edge of the mask generated by the level-set-based method is relatively smooth, it is difficult to introduce SRAFs and the mask accuracy is limited.

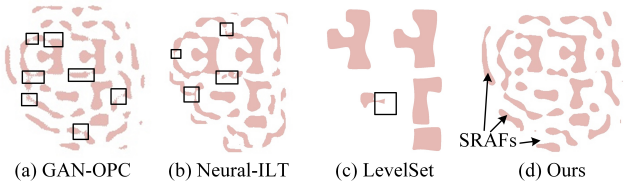

Figure 1. Illustration of optimized mask results from (a) GAN-OPC. (b) Neural-ILT. (c) Level-Set. (d) Ours.

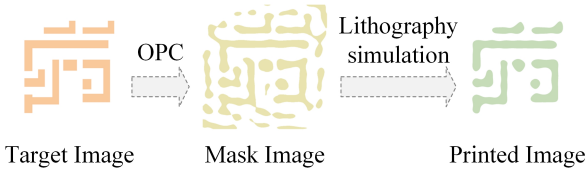

Figure 2. OPC and lithography simulation flow.

As a state-of-the-art image generation framework, the diffusion model[13] enables continuous mask synthesis with superior detail fidelity, an essential capability that is lacking in previous methods. Building on this foundation, we propose MODiff, a novel approach for mask optimization. MODiff adopts a "noise-to-map" generative paradigm, in which predictions are made by progressively denoising a random Gaussian distribution under the guidance of the target layout. Major contributions include:

- We propose MODiff, a mask optimization model based on the diffusion model. Due to the continuity of the model in the inference process and its ability to capture global information, the mask we generated has higher precision, and the mask is more continuous and regular at the edge.
- We introduce forward lithography simulation to better combine diffusion models with mask optimization. The mask optimization process is guided by the mismatch between printed image and target layout.
- We introduce a cosine schedule strategy specifically designed for mask optimization, which better balances the denoising process and preserves fine mask details.
- The experimental results show that our model reaches state-of-the-art performance.

2. PRELIMINARIES

2.1 Lithography Simulation Model

Lithography simulation is designed to approximate the real lithography process in chip manufacturing, simulating the mapping from the input mask $\boldsymbol{M}$ to the output wafer image $\boldsymbol{Z}$. Lithography simulation includes optical projection and photoresist models.[14] The flow of OPC and lithography simulation is shown in Fig. 2. In optical projection, the incident light passes through the mask, transmitting the spatial information of the mask patterns $\boldsymbol{M}$ to the optical projection system, and then obtains the lithographic intensity distribution $\boldsymbol{I}$ on the wafer plane. This process can be described by Hopskin's diffraction theory:[15]

$$\boldsymbol{I}(x,y) \approx \sum_{k=1}^{N_k} \omega_k \left| \boldsymbol{M}(x,y) \otimes \boldsymbol{h}_k(x,y) \right|^2, \tag{1}$$

Where $\boldsymbol{h}_k$ and ω_k are the K^{th} kernel and its weight respectively. We take N_k=24.[16] $\otimes$ denotes the convolution operation. $|\cdot|^2$ denotes computing the square modulus of each element.

The photoresist model transfers the aerial image $\boldsymbol{I}$ to the printed image $\boldsymbol{Z}$. The model checks whether the light intensity of the exposed area exceeds the threshold to generate wafer image $\boldsymbol{Z}$:

$$\boldsymbol{Z}(x,y) = \begin{cases} 1, & \text{if } I(x,y) \geq I_{th}, \\ 0, & \text{if } I(x,y) < I_{th}. \end{cases} \tag{2}$$

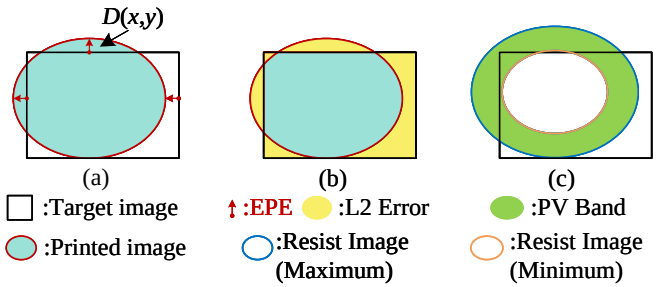

Figure 3. Illustration of the metrics: (a) Edge placement error. (b) Squared L_2 error. (c) Process variation band.

2.2 Inverse Lithography Technology

ILT attempts to solve the problem of mask patterns to find an optimized mask solution $\boldsymbol{M}^* = f^{-1}(\boldsymbol{Z_{tg}}, \boldsymbol{P}_{nom})$ for a given layout $\boldsymbol{Z_{tg}}$, where $f(\boldsymbol{M}, \boldsymbol{P})$ represents forward lithography simulation of mask $\boldsymbol{M}$ under process conditions $\boldsymbol{P}$. We can calculate the gradient of objective function f^{-1} to guide the adjustment of each pixel value, and the best mask can be obtained after iterations.

To achieve gradient descent in the iterative learning method, the lithography model is designed as:

$$\bar{\boldsymbol{Z}}(x,y) = \frac{1}{1+e^{-\alpha(\boldsymbol{I}(x,y)-I_{th})}}, \tag{3}$$

$$\bar{\boldsymbol{M}} = \frac{1}{1+e^{-\beta \boldsymbol{M}(x,y)}}. \tag{4}$$

Where $\boldsymbol{I}_{th}$ is the intensity threshold, α and β are used to control steepness, and (x, y) represents the coordinates on the photoresist image.

2.3 OPC Evaluation Metrics and Problem Formation

The following are commonly used evaluation metrics in OPC, which are also used in our article:

Definition 1 (Squared L_2 Error): Squared L_2 error is calculated by $\|\boldsymbol{Z_{tg}} - \boldsymbol{Z}\|_2^2$, where $\boldsymbol{Z_{tg}}$ is the target layout image and $\boldsymbol{Z}$ is the wafer image generated under normal process conditions $\boldsymbol{P}_{nom}$.

Definition 2 (Process Variation Band): Process variation band (PVB) is an XOR region of two wafer images $\boldsymbol{Z}_{in}$ and $\boldsymbol{Z}_{out}$ generated under minimum and maximum process conditions $\boldsymbol{P}_{min}$ and $\boldsymbol{P}_{max}$ (2% dose error).

Definition 3 (Edge Placement Error): Edge placement error (EPE) measures the horizontal or vertical distance $D(x, y)$ from a given point (EPE measuring point) on the edge of the target to the lithographic profile. If $D(x, y)$ exceeds the threshold distance D_{th}, EPE is considered to be violated.

Problem 1 (Mask Optimization): The goal of mask optimization is to generate the corresponding mask $\boldsymbol{M}$ according to the given target $\boldsymbol{Z_{tg}}$. After lithography simulation, the resisted image $\boldsymbol{Z}$ should be close to the target image $\boldsymbol{Z_{tg}}$ and should minimize L_2 loss, PVB, and EPE.

3. ALGORITHMS AND FRAMEWORK

The overall framework of MODiff is illustrated in Fig. 4. In section 3.1, we first tailor the fundamentals of diffusion models to the requirements of the mask optimization task, and formulate the specific optimization objective used in MODiff. In section 3.2, we design a forward lithography simulation module to explicitly incorporate lithography-related information, ensuring that the model continuously improves the printed resist images during optimization. In section 3.3, we present the detailed architecture of MODiff, together with the training and inference procedures. Finally, in section 3.4, we introduce a progressive mask refinement strategy with cosine diffusion scheduling, which enables higher-precision mask generation.

3.1 Diffusion Model

Diffusion model[17] builds a forward Markov chain that converts clean data into noise by gradually adding a small amount of Gaussian noise, so that a parameterized de-noising network can be learned to predict the noise added at each forward step.

Specifically, in the training stage, a forward Markov chain diffusion process q is defined, in which Gaussian noise is gradually added to the initial image y_0 in the diffusion of T step, where T is called the number of diffusion steps, and in ILT tasks, y_0 is the reference mask $\boldsymbol{M}^*$.

$$q(y_{1:T} \mid y_0) = q(y_1 \mid y_0) \prod_{t=2}^{T} q(y_t \mid y_{t-1}), \tag{5}$$

$$q(y_t \mid y_{t-1}) = \mathcal{N}\left(\sqrt{\alpha_t} y_{t-1}, (1-\alpha_t) I\right), \tag{6}$$

Where $\alpha_{1:T} \in (0,1)$ is the hyperparameter, which controls the variance of noise added at each diffusion, and I is the unit variance. At the same time, for each diffusion, the mean decreases with coefficient $\sqrt{\alpha_{1:T}}$. Therefore, when $t \to \infty, q(y_t) \sim \mathcal{N}(0,1)$. In other words, after sufficient diffusion of the reference mask $\boldsymbol{M}^*$, the image will gradually become pure Gaussian noise. Combined with (5) and (6), the intermediate process can be simplified and diffused directly from y_0 to y_t:

$$q(y_t \mid y_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t} y_0, (1-\bar{\alpha}_t) I\right), \tag{7}$$

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i. \tag{8}$$

The inverse process of the diffusion model starts from the pure noise image $y_T \sim \mathcal{N}(0,1)$. According to the learned conditional transfer distribution $p_\theta(y_{t-1} \mid y_t)$, and the image can be iterated continuously to the previous step ($y_T \to y_{T-1} \to \cdots \to y_2 \to y_1$). We expect $p_\theta(y_{t-1} \mid y_t)$ to approximate $q(y_{t-1} \mid y_t, y_0)$, which can be derived a closed form of Gaussian distribution according to (5)–(8) and Bayes' theorem.[18] The posterior distribution of y_{t-1} is given as:

$$q(y_{t-1} \mid y_t, y_0) = \mathcal{N}(\mu_q(y_t, y_0), \Sigma_q(t)), \tag{9}$$

$$\mu_q(y_t, y_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1}) y_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t) y_0}{1-\bar{\alpha}_t}, \tag{10}$$

$$\Sigma_q(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} I = \sigma_q^2(t) I. \tag{11}$$

The MODiff needs to learn how to reverse this process, the inverse diffusion process P_θ, θ represents the parameters of the neural network:

$$p_\theta(y_{t-1} \mid y_t) = \mathcal{N}(y_{t-1}; \mu_\theta, \Sigma_q(t)) \tag{12}$$

$$\mu_\theta(y_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(y_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \hat{\epsilon}_\theta \right) \tag{13}$$

The loss function at step t is:

$$D_{KL}(q(y_{t-1} \mid y_t, y_0) \parallel p_\theta(y_{t-1} \mid y_t)) = \frac{1}{2\sigma_q^2(t)} \|\mu_\theta - \mu_q\|_2^2 = \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t} \left[\|\epsilon - \hat{\epsilon}_\theta(y_t, t)\|_2^2 \right], \tag{14}$$

where ϵ represents the noise of the diffusion process, randomly sampled from the standard Gaussian distribution. $\hat{\epsilon}_\theta(y_t, t)$ is the predicted noise given by the model. As can be seen from (14), the model is essentially trained to predict the noise at each diffusion step, rather than directly regressing the initial clean mask. This formulation allows the error to be controlled at each reverse diffusion step, thereby improving the robustness of the model.

In MODiff, to ensure the strict correspondence between the target and mask, an additional image target $\boldsymbol{Z}_{tg}$ needs to be introduced to guide the prediction of noise. Therefore, the noise predicted loss at step t is:

$$L_{noise\_t} = \left\| \epsilon - \hat{\epsilon}_\theta \left(y_t, t, \boldsymbol{Z_{tg}} \right) \right\|_2^2. \tag{15}$$

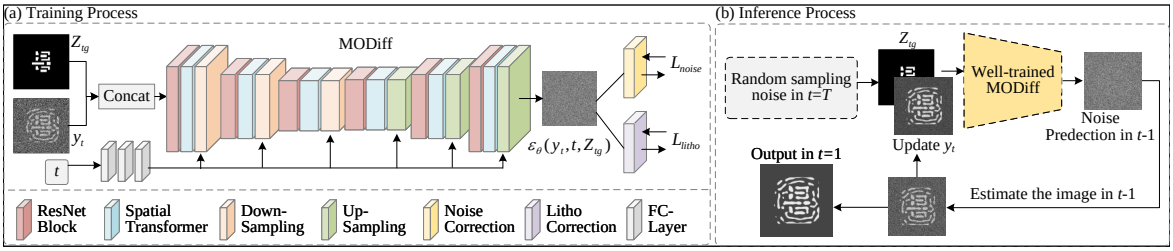

Figure 4. Overview of MODiff: (a) Training process. (b) Inference process.

3.2 Forward Lithography Simulation Module

After the model completes its learning of noise patterns, the mask $\boldsymbol{M}$ can be reconstructed through reverse diffusion. However, since the reference mask $\boldsymbol{M^*}$ itself contains inherent errors, strictly replicating it would limit the model's ultimate accuracy. We treat the model's noise-learning phase as a pretraining step, after which the formal training objective shifts to ensuring that the printed image $\boldsymbol{Z}$ matches the original target image $\boldsymbol{Z_{tg}}$ as closely as possible.

To perform forward lithography simulation, we first need to obtain the initial mask generated by the model. Unfortunately, the generation of the mask by our MODiff requires step-by-step reasoning. If full-step inference is adopted, the computational resource consumption and training time will be extremely high.

Inspired by the forward and backward symmetry of diffusion processes, we propose an approximate method to introduce lithography losses. In the training stage, for the image y_t of step t, we do not gradually deduce from the reverse diffusion process, but directly use the result of forward diffusion to approximate according to (7). The image y_{t-1} is obtained by one-step inverse diffusion according to (12), and the lithography loss of y_{t-1} can guide the noise prediction of step t. While significantly conserving computational resources, the arbitrary selection of t allows us to guide each step of the reverse diffusion process using the lithography simulation module, thereby steering the inverse diffusion toward minimizing the L_2 error. The lithography error of step t is:

$$L_{litho\_t} = \|\boldsymbol{Z_{tg}} - \boldsymbol{Z}_{t-1}\|_2^2, \tag{16}$$

where $\boldsymbol{Z}_{t-1}$ represents the result of the mask at step $(t-1)$ after lithography simulation. Combining (1) – (4), (16) can be further derived:

$$L_{litho\_t} = \|\boldsymbol{Z_{tg}} - f(\mu_\theta(y_t, t) + \sigma_q(t)\epsilon, \boldsymbol{P}_{nom})\|_2^2. \tag{17}$$

The gradient of lithography loss can be deduced by the chain rule:

$$\frac{\partial Z_{t-1}}{\partial \theta} = \frac{\partial Z_{t-1}}{\partial f} \cdot \frac{\partial f}{\partial y_{t-1}} \cdot \frac{\partial y_{t-1}}{\partial \mu_\theta} \cdot \frac{\partial \mu_\theta}{\partial \hat{\epsilon}_\theta} \cdot \frac{\partial \hat{\epsilon}_\theta}{\partial \theta}, \tag{18}$$

$$\frac{\partial L_{litho\_t}}{\partial \theta} = -2(\boldsymbol{Z_{tg}} - Z_{t-1}) \cdot \frac{\partial f}{\partial y_{t-1}} \cdot \frac{1-\alpha_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}} \cdot \frac{\partial \hat{\epsilon}_\theta}{\partial \theta}. \tag{19}$$

It can be found that the gradient of lithography loss can be converted to the gradient of noise prediction loss. The total loss function at step t is:

$$L_t = (1-w) * L_{noise\_t} + w * L_{litho\_t}, \tag{20}$$

where w represents the weight of lithography loss. Considering that the more steps of inference, the closer the intermediate image is to the reference mask, we let w gradually increase as t decreases.

Algorithm 1 MODiff Training Process

Input: Training data D, Diffusion time step T, Model to be trained *model*.

for Target layout Z_{tg}, Reference mask M^* in D **do**
  $t \leftarrow$ Take a random sample from T;
  $y_0 \leftarrow M^*$;
  $\epsilon \leftarrow$ sample random noise;
  $y_t \leftarrow$ from y_0 to step t according to (7);
  $\hat{\epsilon}_\theta \leftarrow$noise predicted by $model(y_t, t, \boldsymbol{Z}_{tg})$
  $L_{noise\_t} = \|\epsilon - \hat{\epsilon}_\theta\|_2^2$;
  if *pretrain* **then**
    $L_t = L_{noise\_t}$;
  else
    $y_{t-1} \leftarrow$ reverse diffusion according to (12);
    $Z_{t-1} \leftarrow f(y_{t-1}, \boldsymbol{P}_{nom})$;
    $L_{litho\_t} \leftarrow \|\boldsymbol{Z_{tg}} - \boldsymbol{Z}_{t-1}\|_2^2$;
    $L_t \leftarrow (1-w) * L_{noise\_t} + w * L_{litho\_t}$;
  end if
  $\nabla_\theta \leftarrow \frac{\partial \mathcal{L}_t}{\partial \theta}$;
  $\theta \leftarrow \theta - \eta \nabla_\theta$; ▷ η is the learning rate
end for

Algorithm 2 MODiff Inference Process

Input: Target layout Z_{tg}, Diffusion time step T, Well-trained model *model*.
Output: Optimized mask M.

$y_T \leftarrow$ sample random noise image;
for t in $T \sim 1$ **do**
  $\hat{\epsilon}_\theta \leftarrow$ noise predicted by $model(y_t, t, \boldsymbol{Z}_{tg})$;
  $y_{t-1} \leftarrow$ perform reverse diffusion according to (12);
end for
$M \leftarrow y_0$;

3.3 Framework of MODiff

The structure of MODiff is shown in Fig. 4. It should be noted in particular that "ResNet Block" represents the residual module, and "Spatial Transformer" refers to the self-attention module. "Noise Correction" and "Litho Correction" represent the correction of model parameters by backpropagation of noise prediction loss and lithography loss, respectively.

The training process of the model is shown in Fig. 4 (a), and a more detailed description is provided in Algorithm 1. In MODiff, diffusion begins with the reference mask M^*. As shown in the second line: during the training progress, for each training image, we only select a time step t with equal probability between 1 and T, and only calculate the loss in this time step. Over many iterations, the model is exposed to all timesteps, which preserves training effectiveness while significantly reducing computational cost. We consider only the noise loss during the pretraining phase, while incorporating the lithography loss in the training phase.

The inference process of the model is shown in Fig. 4 (b) and a detailed procedure is provided in Algorithm 2. With continuous iteration, the desired mask will eventually be obtained.

3.4 Progressive Mask Refinement via Cosine Diffusion Scheduling

In conventional diffusion models, the noise scheduling strategy plays a pivotal role in the quality of generation. Traditional linear scheduling often leads to an imbalance between global and local generation, as well as edge-blending artifacts. For our mask optimization task which demands precise control of mask features, we introduce

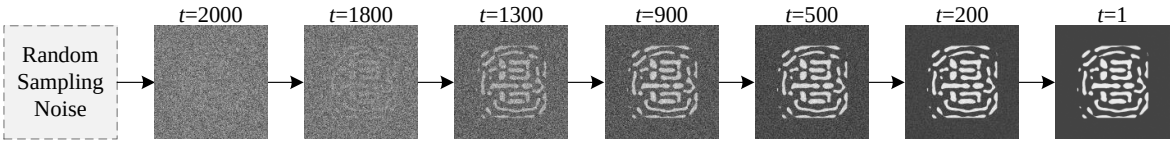

Figure 5. Mask generation process.

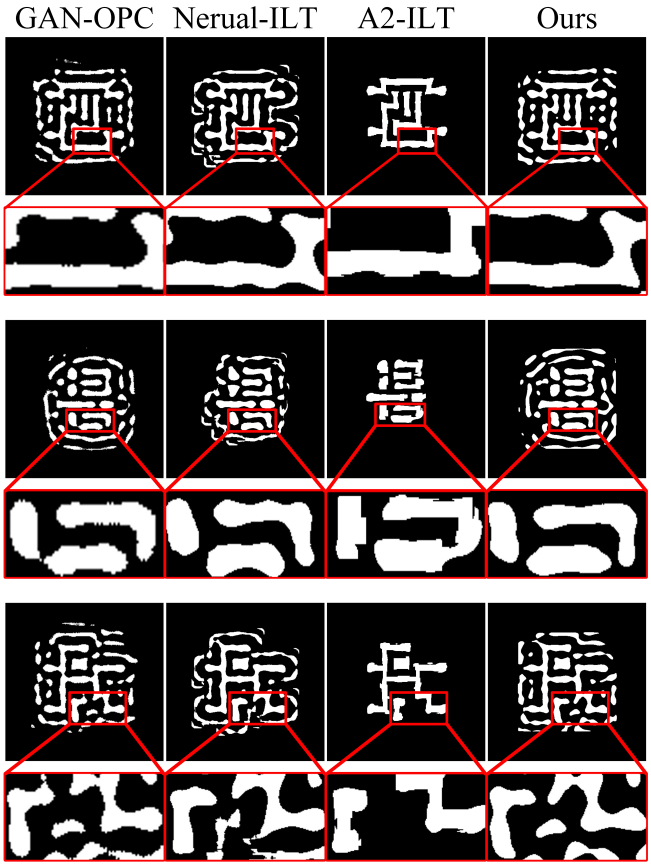

Figure 6. Masks generated by several different methods

a cosine schedule strategy to address these limitations:

$$\bar{\alpha}_t = \frac{\cos^2\left(\frac{\frac{t}{T}+s}{1+s}\cdot\frac{\pi}{2}\right)}{\cos^2\left(\frac{s}{1+s}\cdot\frac{\pi}{2}\right)}, \quad s = 0.006 \tag{21}$$

This scheduling approach enables the model to maintain gradual noise reduction during the initial and final denoising phases while achieving accelerated noise removal in the intermediate stage. Specifically, during the early reverse diffusion steps, the model can generate global structures under high-noise conditions. The middle steps rapidly eliminate noise to refine internal details within the masked regions. Finally, low-noise fine-tuning further enhances the precision of the mask. This progression is clearly illustrated in Fig. 5.

4. EXPERIMENTAL RESULTS

We use Pytorch to implement the entire framework of MODiff and test it on a Linux system. All the framework is equipped with a single Nvidia H800 GPU. Our training dataset is obtained from an open-source OPC dataset,[13] which contains target images along with their corresponding reference masks. Our evaluation dataset is provided

Table 1. Comparison with ILT methods based on generative neural network on ICCAD 2013 dataset

Bench	ILT[16]			PGAN-OPC[9]			Neural-ILT[20]			Ours		
	EPE	L_2 (nm^2)	PVB (nm^2)	EPE	L_2 (nm^2)	PVB (nm^2)	EPE	L_2 (nm^2)	PVB (nm^2)	EPE	L_2 (nm^2)	PVB (nm^2)
case1	6	49863	65534	8	52570	56267	8	50795	63695	3	39925	50475
case2	10	50369	48230	13	42253	56267	3	36969	60232	0	31241	38670
case3	59	81007	108608	51	83663	94498	52	94447	85358	20	62283	82785
case4	1	20044	28285	2	19965	28957	2	1742	32287	1	11381	23482
case5	6	44656	58835	8	44733	59328	3	42337	65536	0	33376	52293
case6	1	57375	48739	12	46062	52845	5	39601	59247	0	31188	47542
case7	0	37221	43490	7	26438	47981	0	25424	50109	0	17733	39307
case8	2	19782	22846	0	17690	23564	0	15588	25826	0	14366	20541
case9	6	55399	66331	12	56125	65417	2	52304	68650	0	37170	61683
case10	0	24381	18097	0	9990	19893	0	10153	22443	0	11426	16552
Average	9.1	44012.7	50899.5	11.3	39948.9	49957.2	7.5	38503.8	53338.3	**2.4**	**29008.9**	**43335.0**
Ratio	3.79	1.52	1.17	4.71	1.38	1.15	3.12	1.33	1.23	**1**	**1**	**1**

Table 2. Comparison with ILT methods based on other models on ICCAD 2013 dataset

Bench	GLS-ILT[8]			A2-ILT[10]			CircleOpt[11]			Ours		
	EPE	L_2 (nm^2)	PVB (nm^2)	EPE	L_2 (nm^2)	PVB (nm^2)	EPE	L_2 (nm^2)	PVB (nm^2)	EPE	L_2 (nm^2)	PVB (nm^2)
case1	10	46032	62693	7	45824	59136	3	43358	46905	3	39925	50475
case2	1	36177	50642	3	33976	52054	1	35496	37920	0	31241	38670
case3	64	71178	100945	62	94634	82661	32	75206	66241	20	62283	82785
case4	2	16345	29831	2	20405	29435	1	13205	23234	1	11381	23482
case5	1	47103	56328	1	37038	62068	1	34938	53110	0	33376	52293
case6	2	46205	51033	2	40701	54842	0	36797	44269	0	31188	47542
case7	0	28609	44953	0	21840	48474	0	21036	41118	0	17733	39307
case8	0	19477	22541	0	14912	24598	0	13906	19859	0	14366	20541
case9	0	52613	62568	2	47489	68056	1	47844	54624	0	37170	61683
case10	0	22415	18769	0	9399	20243	0	9107	16969	0	11426	16552
Average	8.0	38645.4	50030.3	7.9	36621.8	50156.7	3,9	33089.3	**40451.5**	**2.4**	**29008.9**	43335.0
Ratio	3.33	1.33	1.15	3.30	1.26	1.16	1.625	1.14	**0.93**	**1**	**1**	1

by the ICCAD 2013 CAD Contest.[19] We use "L_2", "PVB", and "EPE" to evaluate the results. The diffusion time step T is 2000. $(1 - \alpha_{1:T})$ increases from 0.0001 to 0.02.

4.1 Comparison With State-of-the-Art Methods

We compare the proposed MODiff with the conventional ILT[16] method and other state-of-the-art ILT methods, including level-set-based methods[8] as well as other machine learning-based methods.[9,10,20,21] We present the results of the model based on generative neural network in Table 1, and the results based on other methods in Table 2. The "ILT" shown in the first column of Table 1 refers to the conventional ILT algorithm. Compared to the conventional ILT,[16] the L_2 and PVB of our MODiff decreased by 52% and 17%, respectively. We also achieved a 279% reduction in EPE violations. Meanwhile, in comparison with the level-set-based method GLS-ILT,[8] Our L_2 decreased by 33%, PVB decreased by 15%, and EPE decreased by 233%. In comparison to the state-of-the-art CircleOpt,[11] our model achieved a 14% reduction in L_2 and an 62.5% reduction in EPE.

We show the masks generated by several different sets of models in Fig. 6. Compared with models[9,10,20] that predict masks directly, our model can generate more regular and continuous masks, significantly enhancing both pattern fidelity and printability.

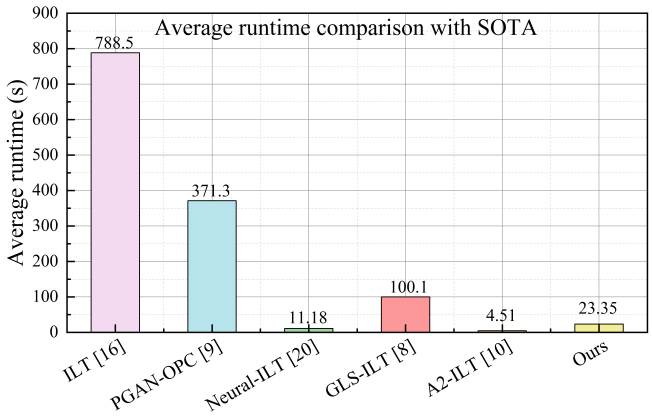

Figure 7. Average runtime comparison with SOTA.

Table 3. Ablation Study on Model Enhancements

	Linear schedule	Cosine schedule	MODiff
EPE	4.2	2.6	**2.4**
$L_2(nm^2)$	30931.0	29906.3	**29008.9**
PVB (nm^2)	**42722.2**	43211.0	43335.0

4.2 Runtime Comparison With SOTA

The average runtime of each model on the ICCAD 2013 dataset is illustrated in Fig. 7. The results show that our approach achieves a 33.77×speedup over conventional ILT.[16] Compared to level-set-based methods[8] that also employ continuous prediction, our method delivers a 4.29×faster performance. Although our runtime is slightly longer than that of direct-prediction models,[10,20] the trade-off is justified by our masks' higher accuracy and more geometrically regular shapes.

4.3 Ablation Study on Model Enhancements

We evaluated models with different configurations on the ICCAD 2013 dataset, with average results presented in Table 3. The first two columns show the pretraining-only performance using linear and cosine scheduling respectively, while the last column incorporates the lithography simulation module with cosine scheduling. Our analysis reveals that cosine scheduling reduces the mask L2 loss by 3.4% and achieves a remarkable 61.5% decrease in EPE. Through careful examination, we found that this significant EPE improvement under cosine scheduling primarily stems from case 3, which is the most complex scenario in our evaluation, demonstrating the superior capability of our scheduler in preserving mask details.

With the introduction of lithography loss, we observe additional reductions of 3.1% in L_2 and 8.4% in EPE, confirming that our lithography simulation module effectively guides the diffusion process toward minimizing pattern discrepancies. While the model's emphasis on L_2 optimization leads to a marginal increase in PVB , this trade-off is justified by the substantial improvement in mask accuracy.

4.4 Comparison on the Via Layer

We also tested our model on the via layer,[13] which contains ten 2 µm × 2 µm clips with varying numbers of 70 nm × 70 nm via patterns.

The average test results for these 10 patterns are shown in Table 4. Our model also performs well on the via layer. Compared with PGAN-OPC,[9] although we are slightly inferior in PVB, we achieved a 41.0% reduction in L_2. Compared to Neural-ILT,[20] our model achieved a 21.4% reduction in L_2 and an 11.7% reduction in PVB. Our model also had fewer EPE violations, achieving reductions of 88.6% and 41.0%, respectively.

Table 4. Results comparison on via layer

	PGAN-OPC[9]	Neural-ILT[20]	Ours
EPE	8.3	6.2	**4.4**
$L_2(nm^2)$	14767	12723	**10477**
PVB (nm^2)	**6686**	8537	7644

5. CONCLUSION

In this paper, we propose the MODiff, a model based on diffusion model. We use a "noise to map" generative paradigm to make predictions by progressively removing noise from a random Gaussian distribution, guided by the target layout. A forward lithography simulation-based module is integrated to improve the mask accuracy. We also introduce a cosine scheduling strategy specifically designed for mask optimization.We tested on the data set given by ICCAD 2013 CAD Contest. Compared to state-of-the-art ILT models, we achieved better results, more regular edges of the mask, and we also showed the mask generation process. Our model shows strong generalization ability, delivering promising results on the via layer.

ACKNOWLEDGMENTS

This work was supported in part by Postgraduate Research Practice Innovation Program of Jiangsu Province (Grant Number: KYCX25_0394) and the National Natural Science Foundation of China (62341408).

REFERENCES

[1] Park, J.-S., Park, C.-H., Rhie, S.-U., Kim, Y.-H., Yoo, M.-H., Kong, J.-T., Kim, H.-W., and Yoo, S.-I., "An efficient rule-based opc approach using a drc tool for 0.18 /spl mu/m asic," in [*Proceedings IEEE 2000 First International Symposium on Quality Electronic Design (Cat. No. PR00525)*], 81–85 (2000).

[2] Dong, X. and Zhang, L., "Process-variation-aware rule-based optical proximity correction for analog layout migration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **36**(8), 1395–1405 (2017).

[3] Chen, S., Di, K., Wang, G., Zhao, W., Du, L., and Zou, N., "Delving into topology representation for layout pattern: A novel contrastive learning framework for hotspot detection," in [*2025 62nd ACM/IEEE Design Automation Conference (DAC)*], 1–6, IEEE (2025).

[4] Awad, A., Takahashi, A., Tanaka, S., and Kodama, C., "A fast process variation and pattern fidelity aware mask optimization algorithm," in [*2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*], 238–245 (2014).

[5] Kuang, J., Chow, W.-K., and Young, E. F. Y., "A robust approach for process variation aware mask optimization," in [*2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*], 1591–1594 (2015).

[6] Su, Y.-H., Huang, Y.-C., Tsai, L.-C., Chang, Y.-W., and Banerjee, S., "Fast lithographic mask optimization considering process variation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **35**(8), 1345–1357 (2016).

[7] Chen, G., Yu, Z., Liu, H., Ma, Y., and Yu, B., "Develset: Deep neural level set for instant mask optimization," in [*2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*], 1–9 (2021).

[8] Yu, Z., Chen, G., Ma, Y., and Yu, B., "A gpu-enabled level-set method for mask optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **42**(2), 594–605 (2023).

[9] Yang, H., Li, S., Deng, Z., Ma, Y., Yu, B., and Young, E. F. Y., "Gan-opc: Mask optimization with lithography-guided generative adversarial nets," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **39**(10), 2822–2834 (2020).

[10] Wang, Q., Jiang, B., Wong, M. D. F., and Young, E. F. Y., "A2-ilt: Gpu accelerated ilt with spatial attention mechanism," in [*Proceedings of the 59th ACM/IEEE Design Automation Conference*], *DAC '22*, 967–972, Association for Computing Machinery, New York, NY, USA (2022).

[11] Zhang, X., Zheng, S., Chen, G., Zhu, B., Xu, H., and Yu, B., "Fracturing-aware curvilinear ilt via circular e-beam mask writer," in [*Proceedings of the 61st ACM/IEEE Design Automation Conference*], *DAC '24*, Association for Computing Machinery, New York, NY, USA (2024).

[12] Xu, X., Lin, Y., Li, M., Matsunawa, T., Nojima, S., Kodama, C., Kotani, T., and Pan, D. Z., "Subresolution assist feature generation with supervised data learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **37**(6), 1225–1236 (2018).

[13] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B., "Score-based generative modeling through stochastic differential equations," (2021).

[14] Zheng, S., Yang, H., Zhu, B., Yu, B., and Wong, M. D., "Lithobench: benchmarking ai computational lithography for semiconductor manufacturing," in [*Proceedings of the 37th International Conference on Neural Information Processing Systems*], *NIPS '23*, Curran Associates Inc., Red Hook, NY, USA (2023).

[15] Hopkins, H. H., "The concept of partial coherence in optics," (Aug 1951).

[16] Gao, J.-R., Xu, X., Yu, B., and Pan, D. Z., "Mosaic: Mask optimizing solution with process window aware inverse correction," in [*2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*], 1–6 (2014).

[17] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S., "Deep unsupervised learning using nonequilibrium thermodynamics," (2015).

[18] Ho, J., Jain, A., and Abbeel, P., "Denoising diffusion probabilistic models," (2020).

[19] Banerjee, S., Li, Z., and Nassif, S. R., "Iccad-2013 cad contest in mask optimization and benchmark suite," in [*2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*], 271–274 (2013).

[20] Jiang, B., Liu, L., Ma, Y., Yu, B., and Young, E. F. Y., "Neural-ilt 2.0: Migrating ilt to domain-specific and multitask-enabled neural network," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **41**(8), 2671–2684 (2022).

[21] Zhu, B., Zheng, S., Yu, Z., Chen, G., Ma, Y., Yang, F., Yu, B., and Wong, M. D. F., "L2o-ilt: Learning to optimize inverse lithography techniques," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **43**(3), 944–955 (2024).