MODiff: Layout-Guided Mask Optimization via Diffusion Model

Jiale Li[1*], Silin Chen[1*], Yiqing Wang[1], Kangjian Di[1], Li Du[1], Ningmu Zou[1,2†],
[1]School of Integrated Circuits, Nanjing University, Suzhou, China
[2]Interdisciplinary Research Center for Future Intelligent Chips (Chip-X), Nanjing University, Suzhou, China

***Abstract*—Inverse lithography technology (ILT) is one of the resolution enhancement techniques (RETs) that is very promising in modern chip manufacturing. However, traditional ILT methods need to consume a lot of computing resources, and the existing models based on deep learning are unstable in training due to the lack of a smooth prediction process. As a result, the generated mask is relatively discontinuous, which leads to a loss of mask accuracy and printability. In this paper, we proposed the MODiff based on the diffusion model. The model can make strong control over both the global layout map and each detailed object in the process of mask generation to obtain a high-quality mask. Firstly, we use a "noise to map" generative paradigm to make predictions by removing noise from a random Gaussian distribution step by step, guided by the target layout. Secondly, we propose a forward lithography simulation-based module to improve the mask accuracy. Finally, We introduce a cosine scheduling strategy specifically designed for mask optimization. Experimental results demonstrate that our method provides a clear and interpretable visualization of the mask generation process, and achieves superior performance compared with other state-of-the-art approaches.**

***Index Terms*—Inverse lithography, mask optimization, diffusion model.**

I. 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. Optical proximity correction (OPC), one of the critical resolution enhancement techniques (RETs), effectively mitigates these discrepancies by directly adjusting the mask features. The primary approaches for OPC include rule-based OPC [1], model-based OPC [2], [3], 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. ILT is a mathematically inverse approach that treats OPC as an inverse imaging problem with pixel-by-pixel mask optimization. Additionally, the level-set method [4], [5] provides an implicit alternative.

The traditional ILT [6] 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. [7] took the lead in applying conditional generation adversarial networks to realize the end-to-end OPC process. Yu et al. [5] used GPU to accelerate the iteration of the level-set function. CircleOpt [8], as a state-of-the-art method, integrates arc constraints into the inverse ILT flow to generate circle-based masks. High-fidelity masks are usually obtained through incremental boundary updates subject to printing and manufacturability constraints, whereas most existing models predict the final mask in a single step. Lacking a smooth refinement process, these models often train unstably and produce boundary discontinuities, degrading mask accuracy and printability. Although the method based on the level-set function [4], [5] adopts the method of iterating level-set function step by step to realize the continuous optimization of the mask, they face challenges in incorporating sub-resolution assist features (SRAFs), which are commonly used to enhance mask quality and achieve higher precision optimization [9]. As shown in Fig. 1, GAN-OPC and Neural-ILT are prone to violations at the edge of the mask (marked in black boxes). 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.

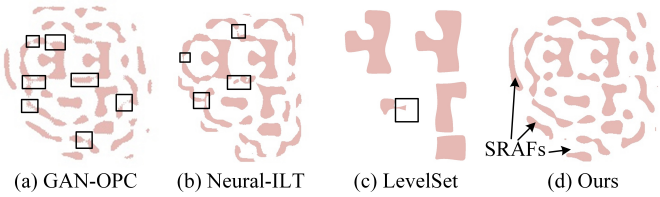

Fig. 1. Illustration of optimized mask results from (a) GAN-OPC. (b) Neural-ILT. (c) Level-Set. (d) Ours.

To address the above issues, we adopt a diffusion-based [10] scheme for iterative optimization and propose MODiff. MODiff follows a "noise-to-map" generative paradigm: at each step, the denoising process is driven by the objective function, naturally yielding a smooth refinement trajectory and more regular mask boundaries, thereby enabling high-fidelity mask optimization. In addition, we incorporate a forward lithography simulation-based module and a cosine noise schedule to further improve mask accuracy. Major contributions include:

- We propose MODiff, a diffusion-model-based framework for high-fidelity mask optimization.
- We introduce a forward lithography simulation–based

*Jiale Li and Silin Chen contributed equally to this work.
†Corresponding Author: nzou@nju.edu.cn. 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).

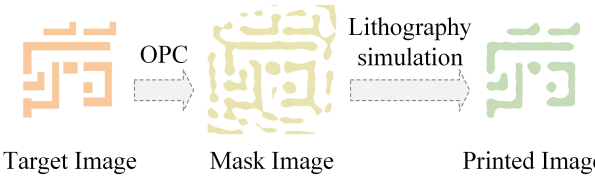

Fig. 2. OPC and lithography simulation flow.

module that drives the denoising process using the simulated lithography outcomes.
- We introduce a cosine schedule strategy specifically designed for mask optimization, which better balances the denoising process and preserves fine mask details.
- Experimental results demonstrate that our method achieves superior performance compared with other state-of-the-art approaches.

II. PRELIMINARIES

A. Lithography Simulation Model

Lithography simulation is designed to approximate the real lithography process in chip manufacturing. Lithography simulation includes optical projection and photoresist models [11]. 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 [12]:

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

where N_k=24 in our implementation, $\boldsymbol{h}_k$ and ω_k are the K^{th} kernel and its weight respectively.

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 I_{th} to generate wafer image $\boldsymbol{Z}$:

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

To enable a differentiable lithography simulation pipeline, the photoresist model is implemented as:

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

where τ_{resist} controls the sharpness of the transition.

B. 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.

III. ALGORITHMS AND FRAMEWORK

This section details the algorithm of MODiff. Section III.A introduces diffusion models for mask generation, Section III.B presents the forward lithography simulation module for lithography-loss-guided denoising. Section III.C describes the cosine noise schedule for finer mask refinement and in Section III.D, we present an overview of MODiff.

A. Diffusion Model for Mask Optimization

Diffusion model [13] 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 $\boldsymbol{M}^*$.

$$q(y_{1:T} \mid y_0) = q(y_1 \mid y_0) \prod_{t=2}^{T} q(y_t \mid y_{t-1}), \tag{4}$$

$$q(y_t \mid y_{t-1}) = \mathcal{N}\left(\sqrt{\alpha_t} y_{t-1}, (1-\alpha_t) I\right), \tag{5}$$

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 $t \to \infty, q(y_t) \sim \mathcal{N}(0,1)$. Combined with (4) and (5), the intermediate process can be simplified and diffused directly from y_0 to y_t:

$$q(y_t \mid y_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t} y_0, (1-\bar{\alpha}_t) I\right), \tag{6}$$

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i. \tag{7}$$

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

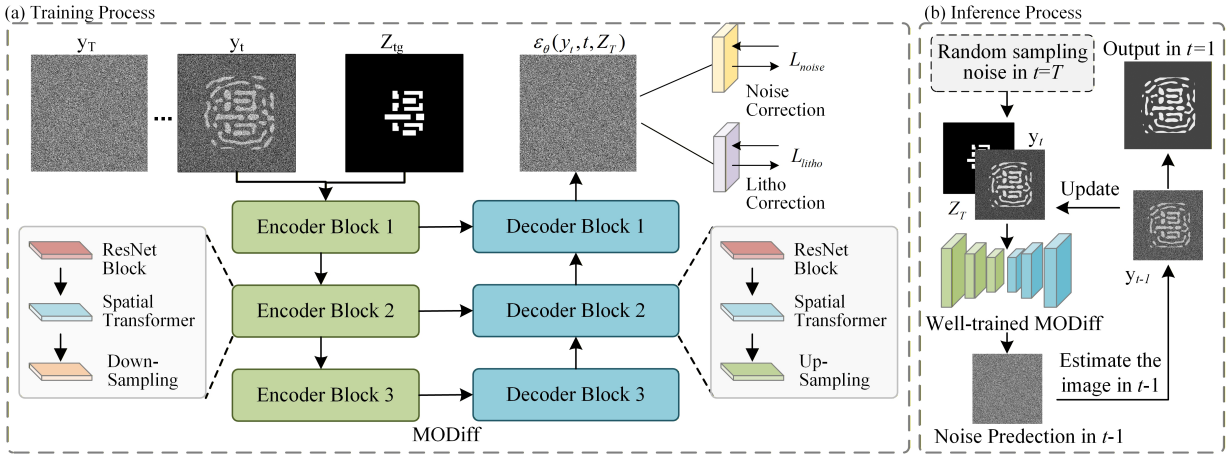

Fig. 3. Overview of MODiff: (a) Training process. (b) Inference process.

can be iterated continuously to the previous step. 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 Bayes' theorem [14]. The posterior distribution of y_{t-1} is given as:

$$q(y_{t-1} \mid y_t, y_0) = \mathcal{N}(\mu_q(y_t, y_0), \Sigma_q(t)), \tag{8}$$

$$\mu_q(y_t, y_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})y_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)y_0}{1-\bar{\alpha}_t}, \tag{9}$$

$$\Sigma_q(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} I = \sigma_q^2(t) I. \tag{10}$$

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

$$p_\theta(y_{t-1} \mid y_t) = \mathcal{N}(y_{t-1}; \mu_\theta, \Sigma_q(t)) \tag{11}$$

$$\mu_\theta(y_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(y_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \hat{\epsilon}_\theta \right) \tag{12}$$

The loss function at step t is:

$$\begin{aligned} D_{KL}(q(y_{t-1} \mid y_t, y_0) \parallel p_\theta(y_{t-1} \mid y_t)) &= \frac{1}{2\sigma_q^2(t)} \|\mu_\theta - \mu_q\|_2^2 \\ &= \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t} \left[\|\epsilon - \hat{\epsilon}_\theta(y_t, t)\|_2^2 \right], \end{aligned} \tag{13}$$

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.

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:

$$L_{noise\_t} = \|\epsilon - \hat{\epsilon}_\theta(y_t, t, \boldsymbol{Z_{tg}})\|_2^2. \tag{14}$$

B. 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. Therefore, we treat this as a pretraining stage to ensure that MODiff acquires the basic denoising capability. In training stage, our goal is for printed image $\boldsymbol{Z}$ matches the original target image $\boldsymbol{Z_{tg}}$ as closely as possible. We detail the training pipeline in Algorithm 1.

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 (6). The image y_{t-1} is obtained by one-step inverse diffusion according to (11), 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:

$$L_{litho\_t} = \|\boldsymbol{Z_{tg}} - \boldsymbol{Z}_{t-1}\|_2^2, \tag{15}$$

where $\boldsymbol{Z}_{t-1}$ represents the result of the mask at step $(t-1)$ after lithography simulation. Combining (1) – (3), L_{litho_t} can be further derived:

$$L_{litho\_t} = \|\boldsymbol{Z_{tg}} - f(\mu_\theta(y_t, t) + \sigma_q(t)\epsilon, \boldsymbol{P}_{nom})\|_2^2. \tag{16}$$

where $f(\boldsymbol{M}, \boldsymbol{P_{nom}})$ represents forward lithography simulation of mask $\boldsymbol{M}$ under normal process conditions $\boldsymbol{P_{nom}}$.

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

$$\frac{\partial Z_{t-1}}{\partial \theta} = \frac{\partial Z_{t-1}}{\partial f} \cdot \frac{\partial f}{\partial y_{t-1}} \cdot \frac{\partial y_{t-1}}{\partial \mu_\theta} \cdot \frac{\partial \mu_\theta}{\partial \hat{\epsilon}_\theta} \cdot \frac{\partial \hat{\epsilon}_\theta}{\partial \theta}, \tag{17}$$

$$\frac{\partial L_{litho\_t}}{\partial \theta} = -2(\boldsymbol{Z_{tg}} - Z_{t-1}) \cdot \frac{\partial f}{\partial y_{t-1}} \cdot \frac{1-\alpha_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}} \cdot \frac{\partial \hat{\epsilon}_\theta}{\partial \theta}. \tag{18}$$

The total loss function at step t is:

$$L_t = (1 - w_t) * L_{noise\_t} + w_t * L_{litho\_t}, \tag{19}$$

where w_t represents the weight of lithography loss, since $\bar{\alpha}_t$ increases as t decreases, the denoised mask becomes progressively more reliable, so we scale up the lithography guidance toward later denoising steps:

$$w_t = \lambda_{\text{litho}} \cdot (\bar{\alpha}_t)^2 \tag{20}$$

C. Progressive Mask Refinement via Cosine Diffusion Scheduling

The noise scheduling strategy plays a pivotal role in the quality of generation in diffusion models. 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 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 schedule preserves gradual noise decay at the beginning and end, while accelerating denoising in the middle. Early steps capture global structure, middle steps quickly refine internal details, and late low-noise steps fine-tune boundary precision. This progression is clearly illustrated in Fig. 4.

D. Framework of MODiff

The structure of MODiff is shown in Fig. 3 (a). 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 inference process of the model is shown in Fig. 3 (b) and a detailed procedure is provided in Algorithm 2. With continuous iteration, the desired mask will eventually be obtained.

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 (6);
  $\hat{\epsilon}_\theta \leftarrow$noise predicted by $model(y_t, t, \boldsymbol{Z}_{tg})$
  $L_{noise\_t} = \|\epsilon - \hat{\epsilon}_\theta\|_2^2$;
  if $pretraining$ **then**
    $L_t = L_{noise\_t}$;
  else
    $y_{t-1} \leftarrow$ reverse diffusion according to (11);
    $Z_{t-1} \leftarrow f(y_{t-1}, \boldsymbol{P}_{nom})$;
    $L_{litho\_t} \leftarrow \|\boldsymbol{Z_{tg}} - \boldsymbol{Z}_{t-1}\|_2^2$;
    $L_t \leftarrow (1 - w_t) * L_{noise\_t} + w_t * L_{litho\_t}$;
  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 (11);
end for
$M \leftarrow y_0$;

IV. EXPERIMENTAL RESULTS

We implement MODiff in PyTorch, and all experiments are conducted on a single NVIDIA H800 GPU. Our training dataset is obtained from [11]. Our evaluation dataset is provided by the ICCAD 2013 CAD Contest [15], which also provide a lithography simulator. We use "L_2", "PVB", and "EPE" to evaluate the results. Among them, "EPE" denotes the number of EPE violations. The diffusion time step T is 2000.

A. Comparison With State-of-the-Art Methods

We compare the proposed MODiff with the conventional ILT [6] method and other state-of-the-art ILT methods. We present the results of the model based on generative neural network in Table I, and the results based on other methods in Table II. The "ILT" shown in the first column of Table I refers to the conventional ILT algorithm. Compared to the conventional ILT [6], MODiff achieves 1.52×lower L_2, 1.17×lower PVB, and 3.79×fewer EPE violations on average. Meanwhile, MODiff achieves 1.33×lower L_2, 1.15×lower PVB, and 3.33×fewer EPE violations. In comparison to the state-of-the-art CircleOpt [8], MODiff achieves 1.14×lower L_2 and 1.63×fewer EPE violations.

We show the masks generated by several different sets of models in Fig. 5. Compared with models [7] [16] [17] that predict masks directly, our model can generate more regular and continuous masks, significantly enhancing both pattern fidelity and printability.

B. Runtime Comparison With Other Methods

The average runtime of each model on the ICCAD 2013 dataset is illustrated in Fig. 6. The results show that our

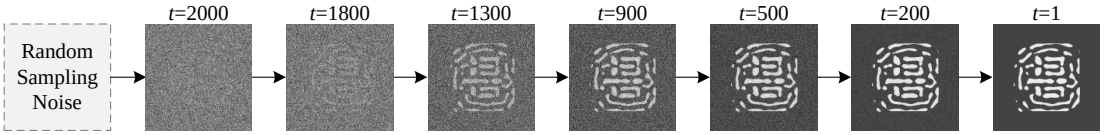

Fig. 4. Mask generation process.

TABLE I
COMPARISON WITH ILT METHODS BASED ON GENERATIVE NEURAL NETWORK ON ICCAD 2013 DATASET

Bench	ILT [6] EPE	ILT [6] L_2 (nm^2)	ILT [6] PVB (nm^2)	PGAN-OPC [7] EPE	PGAN-OPC [7] L_2 (nm^2)	PGAN-OPC [7] PVB (nm^2)	Neural-ILT [16] EPE	Neural-ILT [16] L_2 (nm^2)	Neural-ILT [16] PVB (nm^2)	Ours EPE	Ours L_2 (nm^2)	Ours 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	17420	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 II
COMPARISON WITH ILT METHODS BASED ON OTHER MODELS ON ICCAD 2013 DATASET

Bench	GLS-ILT [5] EPE	GLS-ILT [5] L_2 (nm^2)	GLS-ILT [5] PVB (nm^2)	A2-ILT [17] EPE	A2-ILT [17] L_2 (nm^2)	A2-ILT [17] PVB (nm^2)	CircleOpt [8] EPE	CircleOpt [8] L_2 (nm^2)	CircleOpt [8] PVB (nm^2)	Ours EPE	Ours L_2 (nm^2)	Ours 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

TABLE III
ABLATION STUDY ON MODEL ENHANCEMENTS

	Linear schedule	Cosine schedule	Ours
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

TABLE IV
RESULTS COMPARISON ON VIA LAYER

	PGAN-OPC [7]	Neural-ILT [16]	Ours
EPE	8.3	6.2	**4.4**
$L_2(nm^2)$	14767	12723	**10477**
PVB (nm^2)	6686	8537	5644

approach achieves a 33.77×faster over conventional ILT [6]. Compared to level-set-based methods [5] that also employ continuous prediction, our method delivers a 4.29×faster. Although our runtime is slightly longer than that of direct-prediction models [16], [17], the trade-off is justified by our masks' higher accuracy and more geometrically regular shapes.

C. Ablation Study on Model Enhancements

We evaluated models with different configurations with average results presented in Table III. 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

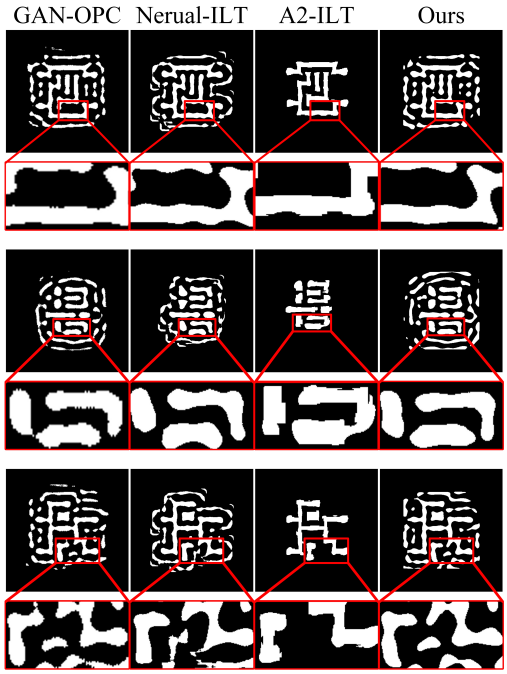

Fig. 5. Masks generated by several different methods

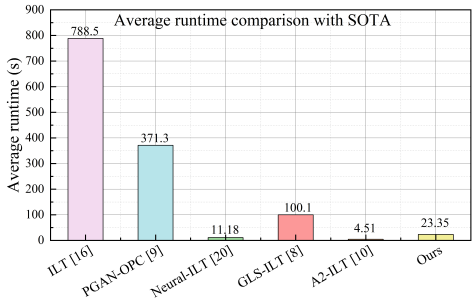

Fig. 6. Average runtime comparison with SOTA.

loss by 3.4% and achieves a remarkable 61.5% decrease in EPE. 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.

D. Comparison on Via Layer

We also tested our model on the via layer according to [11], which contains ten $2\,\mu\mathrm{m} \times 2\,\mu\mathrm{m}$ clips with varying numbers of $70\,\mathrm{nm} \times 70\,\mathrm{nm}$ via patterns. The average test results for these 10 patterns are shown in Table IV. Our model achieves superior performance across all metrics on the via layer, demonstrating its strong generalization capability.

V. CONCLUSION

We propose MODiff, a diffusion-based model for mask optimization. MODiff generates masks via a layout-guided "noise-to-map" denoising process, enhanced with a forward lithography simulation module and a cosine noise schedule for improved accuracy. Experiments show that MODiff offers an interpretable generation trajectory and outperforms state-of-the-art methods.

REFERENCES

[1] J.-S. Park, C.-H. Park, S.-U. Rhie, Y.-H. Kim, M.-H. Yoo, J.-T. Kong, H.-W. Kim, and S.-I. Yoo, "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)*, 2000, pp. 81–85.

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

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

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

[5] Z. Yu, G. Chen, Y. Ma, and B. Yu, "A gpu-enabled level-set method for mask optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 2, pp. 594–605, 2023.

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

[7] 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 Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2822–2834, 2020.

[8] X. Zhang, S. Zheng, G. Chen, B. Zhu, H. Xu, and B. Yu, "Fracturing-aware curvilinear ilt via circular e-beam mask writer," in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, ser. DAC '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3649329.3655926

[9] X. Xu, Y. Lin, M. Li, T. Matsunawa, S. Nojima, C. Kodama, T. Kotani, and D. Z. Pan, "Subresolution assist feature generation with supervised data learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 6, pp. 1225–1236, 2018.

[10] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," 2021. [Online]. Available: https://arxiv.org/abs/2011.13456

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

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

[13] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," 2015. [Online]. Available: https://arxiv.org/abs/1503.03585

[14] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

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

[16] B. Jiang, L. Liu, Y. Ma, B. Yu, and E. F. Y. Young, "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*, vol. 41, no. 8, pp. 2671–2684, 2022.

[17] Q. Wang, B. Jiang, M. D. F. Wong, and E. F. Y. Young, "A2-ilt: Gpu accelerated ilt with spatial attention mechanism," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, ser. DAC '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 967–972. [Online]. Available: https://doi.org/10.1145/3489517.3530579