A Novel Diffusion Model Based Approach for Mask Optimization

ABSTRACT

Mask optimization is a fundamental problem for lithography. In the past years, several machine learning based approaches (e.g., the GAN model) have been proposed to enhance the mask optimization quality and efficiency. Despite of these remarkable progresses, the current methods still suffer from several serious issues for designing mask, such as model collapse and diminished performance, particularly with high-resolution layouts. To tackle these challenges, we propose a novel diffusion model based approach, where the key techniques include a lithography guided loss function to train the model and an efficient denoising sampling algorithm to guide the mask generation process. The experimental results suggest that our method can achieve promising printability with lower computational time for mask optimization.

1 INTRODUCTION

As a fundamental topic in lithography, the *mask optimization (MO)* problem is to design an appropriate mask so that the difference between the generated pattern on the wafer and the target design is minimized. Figure 1 illustrates an example for the lithography process with mask optimization. As the technology nodes continually decrease, the minimum feature size also becomes substantially smaller, and thus the issues like optical diffraction and proximity effects are more and more non-negligible for lithography [1]. Due to these increasing uncertainties, the MO problem also becomes much more challenging in modern chip design. Existing MO methods can be roughly classified into two categories: *model-based* and *inverse lithography technique (ILT)-based*.

In model-based methods, the edges of the initial mask are usually divided into segments, which are moved iteratively under the guidance of the mathematical models. Kuang et al. [2] proposed a robust mask optimization algorithm to jointly optimize the process window and mask printability. Su et al. [3] proposed an effective process variation-aware mask optimization framework that optimizes EPE and PV bands simultaneously (EPE and PV bands are commonly used values to measure the MO quality and their formal definitions are shown in Section 2.1). Matsunawa et al. [4] provided an approach for solving the MO problem based on the Bayesian inference technology.

In inverse lithography technique-based methods, MO is regarded as an inverse imaging problem that is optimized pixel by pixel. Gao et al. [5] introduced the method called MOSAIC, which formalizes the EPE violation as a sigmoid function, and derived the closed form of its gradient for the optimization. Ma et al. [6] developed a gradient-based algorithm to optimize the layout decomposition and mask optimization simultaneously.

Despite of those remarkable progresses, the computational overhead of the MO problem is dramatically increasing as the feature sizes and technology nodes become more and more smaller. So it is urgent to seek other efficient methodologies for dealing with the MO problem beyond conventional methods. As the rapid developing of artificial intelligence, recently it attracts attention on using machine learning techniques for MO. For example, Yang et al.

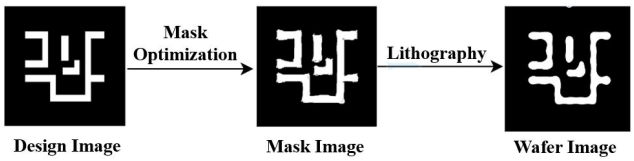


Figure 1: The process of mask optimization and lithography

[7] developed the GAN-OPC model to achieve better mask optimization performance and the training process can be simplified by pre-training. Jiang et al. [8] designed a machine learning-based mask optimization acceleration model, and also proposed a mask printability evaluation framework for lithography-related applications. Chen et al. [9] presented a MO system called DAMO with a high-resolution feature extractor DCGAN-HD. Jiang et al. [10] proposed Neural-ILT, which is an end-to-end neural network that attempts to completely replace the traditional ILT process.

Among those machine learning-based MO methods, a representative technical roadmap comes from the seminal *generative adversarial neteworks (GANs)*, such as [7, 9]. GAN [11] is a popular generative model for image generation, where the key idea is to train a generator and a discriminator by using the adversarial game. However, a major obstacle for applying GANs is called "mode collapse" [12], which is regarded as a central issue of GANs that seriously affects its pattern generation performance in practice. Mode collapse commonly happens when training GANs; roughly speaking, it refers to the phenomenon that the generator network can only produce limited sample diversity and fail to cover all the patterns of the data distribution. Though a number of training methods have been proposed to tackle mode collapse, it still cannot be well avoided in practice [13].

Another recently emerged cutting-edge generative model, called "*diffusion model*" [14], has revealed significantly better performance in many pattern generation tasks, especially for alleviating the mode collapse issue. This new methodology also inspires several new ideas to chip design. For example, Wang et al. [15] recently developed a diffusion model based method to generate layout pattern. However, the current research on the connection between diffusion model and MO is still quite limited, to the best of our knowledge. In fact, the realization of diffusion model for MO incurs several technical challenges, such as how to control the randomness in the generative process, especially for high-resolution mask design, and how to take account of the physical constraints in the training process. In this paper, we aim to tackle those challenges and take the advantage of diffusion model to solve the MO problem efficiently. We propose a novel framework called "DDiff_MO" and our main contributions can be summarized as follows:

- To fully accommodate the impact of physical constraints in lithography, we design a lithography-guided loss function combined with the original diffusion loss. Under such improved diffusion model, we can achieve substantially better training quality for mask generation.

- To further guarantee the computational efficiency of the diffusion model, we also propose a guided sampling method which uses low pass filter to guide the denoising process. Our experimental results suggest that our method can achieve promising printability with lower computational time for mask optimization.

2 PRELIMINARIES

We introduce several important definitions related to mask optimization and diffusion model. Throughout the paper, we denote $\mathbf{Z_t}$ as the target layout, $\mathbf{M}$ as the mask, $\mathbf{I}$ as the aerial image, and $\mathbf{Z}$ for the wafer image. Suppose the images are $N \times N$ with some integer $N > 0$; we use "m" and "n" to denote the indices of the column and row for an image.

2.1 Mask Optimization

Firstly, we introduce the following three metrics to characterize mask. The first two values measure the printability, and the third one assesses the robustness.

- **Edge Placement Error (EPE)** [16] is the distance between the target edge and the lithographic contour printed under nominal conditions in geometry. If a point has the EPE value exceeding a given tolerance value, we recognize it as an EPE violation.
- **Mean Square Error (MSE)** measures the pixel-wise difference between the wafer image and the target layout: $MSE = ||\mathbf{Z} - \mathbf{Z_t}||_F^2$, where $||\cdot||_F$ is the frobenius norm.
- **Process Variation Band (PVB)** [16] is the measure that denotes the total area encompassed by the lithography simulation contours.

Then we introduce the formal definition for MO. In real manufacturing flows, the "Hopkins theory" of the partially coherence imaging system [17] is a widely used simulation and analysis tool for the lithography process. However, the method is complex and usually takes high computational complexity. A more practical approach is introduced in [18], where each pixel of the aerial image $\mathbf{I}$ can be formulated as:

$$\mathbf{I}(m,n) = \sum_{k=1}^{N^2} \alpha_k \left|\mathbf{h}(m,n) \otimes \mathbf{M}(m,n)\right|^2, \tag{1}$$

where $\otimes$ denotes the convolution operation, $\mathbf{h}$ is the lithography kernel, and α_k is the kernel related coefficient. Then the wafer image can be obtained by the following function:

$$\mathbf{Z}(m,n) = \begin{cases} 0 & \text{if } \mathbf{I}(m,n) < I_0 \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where I_0 is a given fixed threshold.

In lithography, the goal of MO is to find a mask image $\mathbf{M}$ to produce a high-quality wafer image $\mathbf{Z}$ from the above (2); the quality is measured by the losses of MSE, EPE, and PVB between $\mathbf{Z}$ and $\mathbf{Z_t}$.

Remark 1. *Actually the process of MO can be viewed as an image-to-image translation problem. That is, given the image $\mathbf{Z_t}$ in design image domain, our task is to generate the corresponding image $\mathbf{M}$ in mask domain. So it is natural to consider utilizing the generative models, such as GANs and diffusion models, to handle this problem.*

2.2 Diffusion Model

In recent years, diffusion model has demonstrated promising performance for various image generation tasks. In this section, we provide a brief overview for the basic structure and training process.

The Framework of Diffusion Model. "Denoising Diffusion Probabilistic Models(DDPM)" [14] is the pioneering work of diffusion models, and we take it as the example to illustrate. The diffusion model is inspired by physical diffusion processes, where it contains a forward process (diffusion process) and a reverse process (denoising process). Both of them are defined as a Markov chain. In the forward process, we gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, ..., \beta_\mathbf{T}$ until the image becomes a pure Gaussian noise; "T" denotes the maximum diffusion step number. The Gaussian noise of each single step can be calculate by:

$$q(\mathbf{x_t}|\mathbf{x_{t-1}}) := \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t \mathbf{E})$$

where $\mathbf{x_t}$ is the set of the latent variables of the t-th step in the same sample space with $\mathbf{x_0}$ (the original image), and $\mathbf{E}$ is the identity matrix. In the reverse process, we gradually subtract the Gaussian noise from a pure Gaussian noise until data is recovered in the original data space. The Gaussian noise of each single step can be calculate by:

$$p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t}) := \mathcal{N}(\mathbf{x_{t-1}}; \mu_\theta(\mathbf{x_t}, \mathbf{t}), \Sigma_\theta(\mathbf{x_t}, \mathbf{t}))$$

where the parameter set θ is learned by the diffusion model to fit the data distribution $q(x0)$. The model predicts the noise at each step in the reverse process, so as to enable the reverse process to generate an image in the source domain from a pure Gaussian noise.

The conditional diffusion model DDIB. The original diffusion model is not suitable for solving mask optimization, because we need to generate the corresponding mask from a given design image, rather than generating a random mask from random noise. The recently proposed "Dual diffusion implicit bridges (DDIB)" [19] is a conditional diffusion model that in particular can be applied to image-to-image translation problem. It consists of two diffusion models D_1 and D_2 that are trained independently in the source image domain $\mathcal{X}$ and the target image domain $\mathcal{Y}$. In the test process, DDIB reverses the denoising process of the source diffusion model D_1, so that it receives the source image $X \in \mathcal{X}$ and outputs the latent encoding Z of the source image X. Then the target diffusion model D_2 receives the latent encoding Z and generates the target images $Y \in \mathcal{Y}$ corresponding to the source image X by the reverse process. The high-level idea for the process is shown in figure 2.

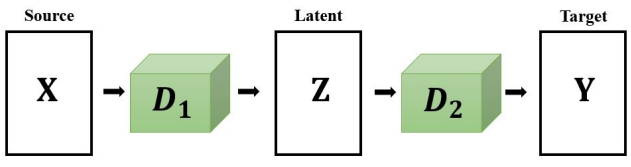


Figure 2: Test process of DDIB. X, Z, and Y represent the source image, the latent encoding, and the target image respectively.

3 OUR DDIFF_MO FRAMEWORK

In this section, we introduce our main result, the MO framework "DDiff_MO", based on the dual diffusion model DDIB. Firstly, we provide an overview of our framework, and then elaborate on the technical parts in Section 3.1 and Section 3.2, respectively.

We notice that it is not an easy job to apply the DDIB framework to the MO problem. The major challenge comes from the special requirements on mask image. According the lithography process and the losses introduced in Section 2.1, the obtained image should be stylistically close to the target layout, and meanwhile it needs to satisfy the physical constraints. For example, when the design image is densely arranged, the mask image should tend to be thinner. The original DDIB was designed for generating general images in our daily life rather than the specialized mask images for lithography, and therefore those constraints are not embedded to its framework and training algorithm.

Our high-level idea. To resolve the aforementioned challenge, we focus on modifying the model design for training, and then propose a new denoising sampling algorithm for the mask generation process. A key idea in our model design is to introduce a lithography-guided loss function, which enables the model to not only learn the mask style but also grasp the underlying physical constraints. By incorporating this loss function, we can enhance the overall quality of mask generation to a great extent. In the denoising sampling algorithm, we leverage a low pass filter, which is a particularly designed two-layer network, to guide the sampling process. The guided sampling algorithm can effectively retain the crucial positional information that is essential for accurate mask representation. In Figure 3 we show the overall denoising process of DDiff_MO.

3.1 Model Design

As introduced in Section 2.2, the model contains the "source diffusion" and "target diffustion" two parts. Because we do not need to generate image in the design image domain, we just directly use the original DDIB [20] model as the source diffusion model. So we only focus on the design for the target diffusion model.

The classical diffusion models usually utilize the l_2-norm loss function:

$$L_0 = \mathbb{E}_{\mathbf{x},\epsilon,t}||\epsilon_\theta(\mathbf{x_t}, t) - \epsilon||_2^2 \text{ or } L_0' = \mathbb{E}_{\mathbf{x},\epsilon,t}||x_\theta(\mathbf{x_t}, t) - \mathbf{x_0}||_2^2$$

for training, where ϵ is a Gaussian noise, $\mathbf{x_0}$ is the training data, ϵ_θ is the module for predicting the noise, and x_θ is the module for predicting $\mathbf{x_0}$. However, the physical constraint also plays a key role for the mask optimization problem. Thus we design a lithography-guided loss function that incorporates lithography results with l_2-norm error to enhance the generation quality for mask image.

The design image and mask image pair is represented by $(\mathbf{y_0}, \mathbf{x_0})$ in our loss. We denote the lithography simulator as S, where a lithography simulator is a simulation model that can output the corresponding design image for a given mask image using the available lithography engine from ICCAD 2013 CAD Contest benchmark [16]. Also let $\lambda > 0$ be the coefficient of the lithography error term. The new loss function can be expressed as Equation (3):

$$L_{\texttt{litho}} = \mathbb{E}_{\mathbf{x},\epsilon,t}||x_\theta(\mathbf{x_t}, t) - \mathbf{x_0}||_2^2 + \lambda||S\big(x_\theta(\mathbf{x_t}, t)\big) - \mathbf{y_0}||_2^2. \quad (3)$$

The detailed training process is described in algorithm 1. In each iteration, we first randomly sample a pair of data points $(\mathbf{y_0}, \mathbf{x_0})$ from the data distribution. Then we sample a step t from $[1, T]$, sample the noise from $\mathcal{N}(0, 1)$, and compute the t-th step latent variable $\mathbf{x_t}$. Finally, we use our proposed loss function (3) to compute the loss and update the model parameters with gradient descent method.

Algorithm 1: Source diffusion model training

Input: dataset $\mathcal{X} \times \mathcal{Y}$, iterating times n, maximum diffusion step T, lithography engine S
Output: diffusion model θ

for $i = 1, 2, \cdots, n$ **do**
  $x_0, y_0 \sim \mathcal{X} \times \mathcal{Y}$
  $t \sim Uniform[1, T]$
  $\epsilon \sim N(0, 1)$
  update θ with the gradient:
    $-\nabla_\theta\big(||x_\theta(x_t, t) - x_0||^2 + \lambda||S(x_\theta(x_t, t)) - y_0||^2\big)$
end

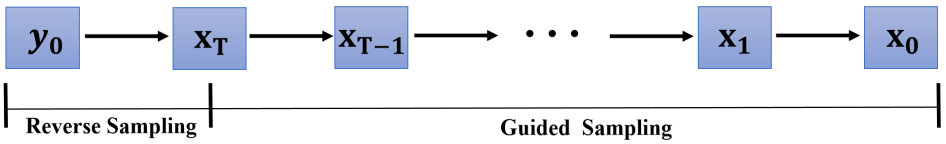
3.2 Mask Denoising Method

Now we turn our attention to the denoising process which directly determines our mask generation result. We first revisit the classical diffusion method [20], where the denoising process is typically iterated based on Equation (4):

$$\begin{aligned}\mathbf{x}_{t-1} =& \sqrt{\alpha_{t-1}}\underbrace{\Big(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{\alpha_t}}\Big)}_{\text{predicted } \mathbf{x}_0} \\ &+ \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta^{(t)}(\mathbf{x}_t)}_{\text{direction pointing to } \mathbf{x}_t} + \underbrace{\sigma_t\epsilon_t,}_{\text{random noise}}\end{aligned} \quad (4)$$

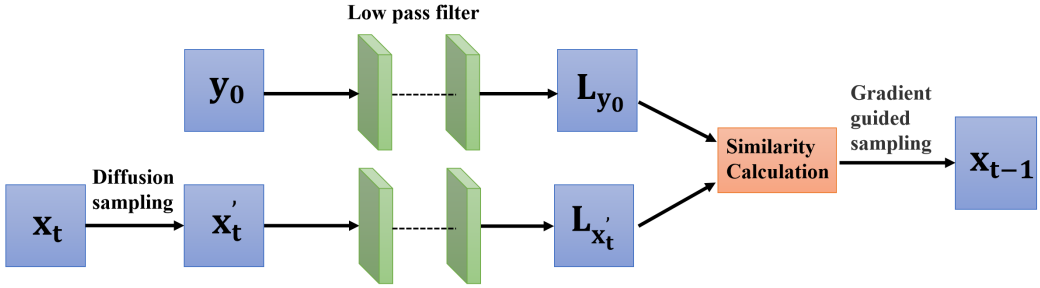
where α_t and σ_t are prespecified constants, $\epsilon_\theta^{(t)}$ is the module for predicting the noise in the next step, and ϵ_t is a Gaussian noise. In this equation, the first two terms are both deterministic and the third term is the only random part. So if σ_t is set to 0, the entire iteration process becomes deterministic. That is, we can obtain the design image $\mathbf{y_0}$ from Gaussian noise through this denoising process by the source diffusion model. Similarly, we can also obtain the Gaussian noise associated with the design image $\mathbf{y_0}$ through the reverse process of the denoising procedure. We refer to this inverse procedure derived by (4) as the "reverse sampling" process. After the reverse sampling, which relies on the source diffusion model D_1 trained on the design image domain, we obtain the latent encoding $\mathbf{z_0}$ of the design image $\mathbf{y_0}$.

However, despite obtaining the latent encoding of $\mathbf{y_0}$, it is still challenging to compute the corresponding data $\mathbf{x_0}$ in mask image domain. The reason is that the diffusion model introduces too much randomness in the denoising process, where each sampling step needs to add a random Gaussian noise to the data. As a consequence, if without effective control, the model often fails to generate high-quality masks $\mathbf{x_0}$ from a given target design image $\mathbf{y_0}$. This



Denoising Process

(a)



Ilustration for one step of the guided sampling: from x_t to x_{t-1}

(b)

Figure 3: Figure (a) shows the overall denoising process of DDiff_MO. Figure (b) elucidates the guided sampling module for the step from x_t to x_{t-1} in the denoising process.

motivates us to design a guided sampling method to constrain the randomness in the denoising process.

Guided sampling method. Our idea is based on the simple observation that the design image and mask image pair $(\mathbf{y_0}, \mathbf{x_0})$ have the same position information. That is, the patterns in $\mathbf{x_0}$ and their corresponding patterns in $\mathbf{y_0}$ should be located at the same positions in these two images. Thus we expect to maintain the positional information unchanged during the generation process. According to the analysis in the recent work [21], positional information of an image in diffusion model often reveals low frequency feature, which means that it changes slowly when adding noise. So we employ a low pass filter, which is a two-layer network for capturing low frequency features, to extract the positional information and consequently guide the denoising process. During the process, we use the gradient descent algorithm to optimize the difference between the positional informations of the newly generated mask image and the design image. The difference can be calculated by the cosine similarity.

The algorithm is described in algorithm 2 and the data flow is shown in Figure 3(b). We use $\mathbf{x_t}$ to represent the latent variable at t-th step and $\mathbf{y_0}$ to represent the design image. At the t-th step of sampling, the algorithm begins by obtaining $\mathbf{x'_{t-1}}$ from $\mathbf{x_t}$ through the diffusion sampling process. Then, it uses a low-pass filter to extract the positional information $\mathbf{P_1}$ and $\mathbf{P_2}$ from $\mathbf{y_0}$ and $\mathbf{x'_{t-1}}$, respectively. Next, the algorithm computes the similarity between $\mathbf{P_1}$ and $\mathbf{P_2}$, and updates $\mathbf{x'_{t-1}}$ by using the gradient ascent method to obtain $\mathbf{x_{t-1}}$. We repeat this process until the final result $\mathbf{x_0}$ is achieved.

Algorithm 2: Guided sampling

Input: dataset $\mathbf{y_0}$, target diffusion model $\mathbf{D_2}$, low pass filter L, the maximum sampling step T , an initial noise $\mathbf{n_0}$, a hyperparameter λ

Output: mask image $\mathbf{x_0}$

$x_T = \mathbf{n_0}$
for $t = T, T-1, \cdots, 1$ **do**
  $\mathbf{x'_{t-1}}$ is derived from $\mathbf{x_t}$ based on the function (4)
  $P_1 = L(\mathbf{x'_{t-1}})$
  $P_2 = L(\mathbf{y_0})$
  $loss$ = the cosine similarity between P_1 and P_2
  $\mathbf{x_{t-1}} = \mathbf{x'_{t-1}} + \lambda \nabla_{\mathbf{x'_{t-1}}}(loss)$
end

4 EXPERIMENTAL RESULTS

We conduct a set of experiments to test our proposed DDiff_MO framework for the MO problem. We evaluate the masks generated by our method using the measures introduced in Section 2.1. We also investigate the computational efficiency of our method.

Datasets. In our experiments, we consider two commonly used datasets for testing MO methods. The first dataset was synthesized by Yang et al. [7] based on the design specifications from 32nm M1

layout topologies provided by ICCAD 2013 CAD Contest benchmark [16]. The wire sizes were adjusted to guarantee that the shapes in synthesized layouts are similar to those in the given benchmark. It contains 4000 training instances.

The second dataset was synthesized by Jiang et al. [22] with the approach proposed in [23], which is a generative learning-based pattern generation framework for pattern topology generation and legalization. It also contains 4000 training instances.

Implementation details. We conduct all of our experiments on a single NVIDIA GeForce RTX 3090 with Pytorch library [24]. We use the improved-DDPM [20] as our basic model for both the source and target diffusion models. We first resize the original images to be 512 × 512 uniformly before the training process. During the training, we set the diffusion step to be 1000 and the iteration number to be 12000. The batchsize is 1, the learning rate is 0.0001, and the number of ResNet block is 3. In the denoising process, we configure the reverse sample step of the source diffusion model to be 500 to obtain the latent encoding of the source image. And we set the sample step of the target diffusion model to be 500 to obtain the target image.

4.1 MO Performance

In figure 4, we show the data flow of the whole method. After obtaining the initial mask image from diffusion model, we use an ILT engine to refine the performance of the mask image. To verify the performance of our method, we test ten industrial 32nm M1 layouts provided by [16] and compare the performance with the conventional ILT [5], the generative model based mask optimization method PGAN-OPC [7] and the learning-based method Neural-ILT [10]. Figure 5 illustrates the design image, mask images and wafer images that are generated by our method for case 1 in ICCAD 2013 contest benchmark suite [16]. The detailed results on MO performance are shown in table 1, where the columns “L2”, “PVB” and "EPE" denote the mean square error, process variation band and the edge placement error, respectively. The results of PGAN-OPC and Neural-ILT are reported from their original articles [7, 10]. The results of ILT are the experimental outcomes obtained by using the Mosai algorithm implemented in a public GitHub repository [25]. And we set the number of iterations to be 40. The model we used is trained on the combination of the datasets from [7, 22]. The results show that our method can achieve the lowest average values for L_2 and PVB, and in particular outperforms the previous machine learning based approaches PGAN-OPC and Neural-ILT.

Figure 4: The data flow of our method .

4.2 Computational Efficiency

In this part, we conduct the experiment to evaluate the computational efficiency of our proposed method. We still take the ten layouts in [16] as the test data and count the whole time of the mask optimization process including generation time and refinement time. And we compared the result with ILT [5]. To illustrate the convergence curve accurately, we take “case 1” as an example for plotting. The curves in figure 6 show the convergence trajectories of ILT and ILT with DDiff_MO generation as the initial input. The results shown in the figure reveal that using the mask image generated by DDiff_MO as the initial input can accelerate the convergence of the conventional ILT method, and meanwhile leads to slightly lower L2 loss. Moreover, we test the whole runtime of the mask optimization process until convergence. For ILT, we test the entire iteration time. For DDiff_MO, we evaluate both the generation time and the iteration time. Due to its faster convergence speed, even though DDiff_MO requires extra time to generate the initial image, it still takes a shorter overall runtime. The average time to generate an image using the diffusion model is approximately 26 seconds, while ILT takes around 3 seconds per iteration. The detailed total times spent on mask optimization are shown in Table 2. The experimental results demonstrate that our approach can accelerate the ILT method by 10% around.

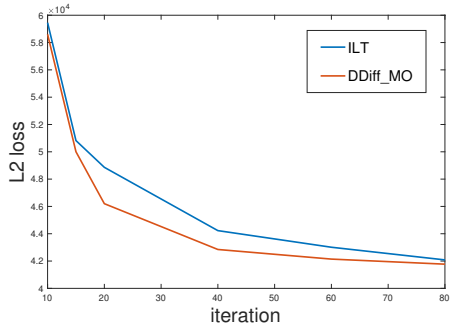


Figure 5: (a) is the design image , (b) is the mask generated by our method and (c) is the wafer image.

Figure 6: The figure shows the convergence trajectories. The horizontal axis represents the iteration number, and the vertical axis represents the L2 loss.

5 CONCLUSION

In this paper, we focus on the generative model for solving the challenging mask optimization problem. We propose a diffusion model based framework DDiff_MO that combines several novel

Table 1: Mask printability preformance (L_2 and PVB unit: nm^2)

Bench	Area(nm^2)	ILT			PGAN-OPC			Neural-TLT			OURS		
		L_2	PVB	EPE	L_2	PVB	EPE	L_2	PVB	EPE	L_2	PVB	EPE
case1	215433	44228	56794	5	52570	56267	–	50795	63695	–	42847	56688	6
case2	169280	33187	47708	0	42253	50822	–	36969	60232	–	31995	47440	3
case3	213504	67852	84936	24	83663	94498	–	94447	85358	–	69272	82513	25
case4	82560	14207	24849	1	19965	28957	–	17420	32287	–	14916	24880	0
case5	281958	37679	57815	0	44733	59328	–	42337	65536	–	37711	57659	0
case6	286234	36760	52376	0	46062	52845	–	39601	59247	–	37043	52216	0
case7	229149	29292	47775	0	26438	47981	–	25424	50109	–	29159	47727	0
case8	128544	14123	24756	0	17690	23564	–	15588	25826	–	14311	24711	0
case9	317581	41097	64496	1	56125	65417	–	52304	68650	–	41445	64632	0
case10	102400	10360	21757	0	9990	19893	–	10153	22443	–	9237	21664	0
Average		32878	48326	**3.1**	39948.9	49957.2	–	38504	53338	–	**32793**	**48013**	3.4

Table 2: Running time (unit: s)

	case1	case2	case3	case4	case5	case6	case7	case8	case9	case10
ILT	128.13	128.09	126.37	125.56	128.13	127.88	128.24	126.77	126.52	124.74
OURS	110.4	108.83	108.39	108.38	111.79	111.55	112.55	109.71	113.04	91.88

techniques for training and denoising. The experimental results suggest that our method can achieve promising printability with lower computational time. As for the future work, it is deserved to consider large scale design image by taking the advantage of diffusion model with further acceleration algorithms.

REFERENCES

[1] David Z Pan, Bei Yu, and Jhih-Rong Gao. Design for manufacturing with emerging nanolithography. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(10):1453–1472, 2013.

[2] Jian Kuang, Wing-Kai Chow, and Evangeline FY Young. A robust approach for process variation aware mask optimization. In *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1591–1594. IEEE, 2015.

[3] Yu-Hsuan Su, Yu-Chen Huang, Liang-Chun Tsai, Yao-Wen Chang, and Shayak Banerjee. Fast lithographic mask optimization considering process variation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(8):1345–1357, 2016.

[4] Tetsuaki Matsunawa, Bei Yu, and David Z Pan. Optical proximity correction with hierarchical bayes model. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 15(2):021009–021009, 2016.

[5] Jhih-Rong Gao, Xiaoqing Xu, Bei Yu, and David Z Pan. Mosaic: Mask optimizing solution with process window aware inverse correction. In *Proceedings of the 51st Annual Design Automation Conference*, pages 1–6, 2014.

[6] Yuzhe Ma, Wei Zhong, Shuxiang Hu, Jhih-Rong Gao, Jian Kuang, Jin Miao, and Bei Yu. A unified framework for simultaneous layout decomposition and mask optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(12):5069–5082, 2020.

[7] Haoyu Yang, Shuhe Li, Yuzhe Ma, Bei Yu, and Evangeline FY Young. Gan-opc: Mask optimization with lithography-guided generative adversarial nets. In *Proceedings of the 55th Annual Design Automation Conference*, pages 1–6, 2018.

[8] Bentian Jiang, Hang Zhang, Jinglei Yang, and Evangeline FY Young. A fast machine learning-based mask printability predictor for opc acceleration. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pages 412–419, 2019.

[9] Guojin Chen, Wanli Chen, Yuzhe Ma, Haoyu Yang, and Bei Yu. Damo: Deep agile mask optimization for full chip scale. In *Proceedings of the 39th International Conference on Computer-Aided Design*, pages 1–9, 2020.

[10] Bentian Jiang, Lixin Liu, Yuzhe Ma, Hang Zhang, Bei Yu, and Evangeline FY Young. Neural-ilt: Migrating ilt to neural networks for mask printability and complexity co-optimization. In *Proceedings of the 39th International Conference on Computer-Aided Design*, pages 1–9, 2020.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[12] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in neural information processing systems*, 30, 2017.

[13] Minguk Kang, Woohyeon Shim, Minsu Cho, and Jaesik Park. Rebooting ACGAN: auxiliary classifier gans with stable training. In *NeurIPS*, pages 23505–23518, 2021.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[15] Zixiao Wang, Yunheng Shen, Wenqian Zhao, Yang Bai, Guojin Chen, Farzan Farnia, and Bei Yu. Diffpattern: Layout pattern generation via discrete diffusion. In *60th ACM/IEEE Design Automation Conference, DAC 2023, San Francisco, CA, USA, July 9-13, 2023*, pages 1–6. IEEE, 2023.

[16] Shayak Banerjee, Zhuo Li, and Sani R Nassif. Iccad-2013 cad contest in mask optimization and benchmark suite. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 271–274. IEEE, 2013.

[17] Harold Horace Hopkins. The concept of partial coherence in optics. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 208 (1093):263–277, 1951.

[18] Xu Ma and Gonzalo R Arce. *Computational lithography*. John Wiley & Sons, 2011.

[19] Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[20] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[21] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations, 2022.

[22] Bentian Jiang, Xiaopeng Zhang, Lixin Liu, and Evangeline FY Young. Building up end-to-end mask optimization framework with self-training. In *Proceedings of the 2021 International Symposium on Physical Design*, pages 63–70, 2021.

[23] Xiaopeng Zhang, James Shiely, and Evangeline FY Young. Layout pattern generation and legalization with generative learning models. In *Proceedings of the 39th International Conference on Computer-Aided Design*, pages 1–9, 2020.

[24] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[25] Su Zheng, Yuzhe Ma, Binwu Zhu, Guojin Chen, Wenqian Zhao, Shuo Yin, Ziyang Yu, and Bei Yu. Openilt: An open-source platform for inverse lithography technique research. https://github.com/OpenOPC/OpenILT/, 2023.