GPU-Accelerated Inverse Lithography Towards High Quality Curvy Mask Generation

Haoyu Yang, Haoxing Ren
NVIDIA Corp.
{haoyuy,haoxingr}@nvidia.com

Abstract

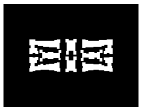
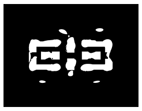
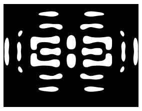
Inverse Lithography Technology (ILT) has emerged as a promising solution for photo mask design and optimization. Relying on multi-beam mask writers, ILT enables the creation of free-form curvilinear mask shapes that enhance printed wafer image quality and process window. However, a major challenge in implementing curvilinear ILT for large-scale production is mask rule checking, an area currently under development by foundries and EDA vendors. Although recent research has incorporated mask complexity into the optimization process, much of it focuses on reducing e-beam shots, which does not align with the goals of curvilinear ILT. In this paper, we introduce a GPU-accelerated ILT algorithm that improves not only contour quality and process window but also the precision of curvilinear mask shapes. Our experiments on open benchmarks demonstrate a significant advantage of our algorithm over leading academic ILT engines. Source code will be available at https://github.com/phdyang007/curvyILT.

1 Introduction

Lithography plays a crucial role in semiconductor manufacturing. However, a mismatch between lithography technology and the critical dimensions of chips leads to the optical proximity effect, posing challenges to technological advancement. To mitigate this issue, chip design photomasks must be optimized to correct for lithography proximity distortion, a process known as mask optimization.

Optical proximity correction (OPC) is the most widely used approach for mask optimization [1–3]. It involves dividing the edges of chip design polygons into segments, which are then adjusted using heuristic rules to counteract optical proximity effects. However, as chip feature sizes continue to shrink, the limited robustness of heuristic optimization demands extensive engineering effort, jeopardizing design turnaround time and production yield. **Inverse lithography technologies** (ILT), with their gradient-based free-form optimization, provide a broader solution space that can effectively address critical patterns where traditional OPC falls short. Despite this advantage, ILT has long faced a dilemma: while free-form optimization leads to better convergence, it also presents a significant manufacturing challenge, as mask shops struggle to produce these complex free-form masks efficiently. A common workaround is to approximate the ILT-generated mask with rectangles, aligning the ILT output with OPC shape rules. However, this approach sacrifices some of the optimality that ILT offers. Recently, a multi-beam mask writer was introduced for advanced lithography mask manufacturing [4]. This innovation maintains a consistent mask production time, enabling the direct fabrication of freeform or curvilinear masks. Though curvilinear ILT faces mask writing challenges to enforce clearance of mask rule check [5], it is now feasible to apply curvilinear inverse lithography technology (ILT) across a significant portion of chip design layers, leading to improved quality of results (QoR). The comparison among OPC, ILT, and Curvy ILT are listed in Table 1, and the development of Curvy ILT is our focus.

Table 1: Mask optimization solution. Our efforts focus on the direct generation of curvy ILT, with a specifically designed algorithm for better curvature and reduced mask artifacts.

Solution	OPC	ILT	**Curvy ILT**
Mask Writer	Variable Shaped Beam	Variable Shaped Beam	Multibeam
Mask Rule	Manhattan Geometry Constraints	Manhattan Geometry Constraints	Width, Area, Curvature
Efficiency	Fast	Slow	Slow
Solution Space	Small	Medium	Large
Optimizer	Heurestic	Gradient	Gradient
Example			

ILT has garnered significant attention in academic research due to its promising advantages. Much of this research has centered on enhancing algorithmic efficiency and optimizing the quality of the final simulated wafer. For example, Wang et al. [6] developed A2-ILT, introducing a spatial attention layer to regulate mask gradients, which however fails the growth of sub-resolution assist features (SRAFs)—a critical aspect for process window optimization. Additionally, Yu et al. [7] proposed using a level-set function to model the mask, improving smoothness during ILT procedures. More recently, an efficient ILT implementation was presented by Sun et al. [8], which has become the state-of-the-art by utilizing multi-level lithography simulations at different resolutions to achieve faster convergence. A similar implementation is also introduced in OpenILT [9]. However, these efforts primarily address the mask manufacturing challenges associated with VSB technology and are not directly applicable to curvilinear mask optimization. For instance, there has been little focus on eliminating isolated artifacts that breach shape area constraints, and the balance between quality of results (QoR) and mask smoothness has not been adequately managed [7, 8]. To overcome the limitations of previous work and

Table 2: Notations and symbols used throughout this paper.

Notation	Description
$\boldsymbol{M}$	Mask image
$\boldsymbol{M}_c$	Continuous tone mask image
$\boldsymbol{Z}^*$	Manhattan design target
$\boldsymbol{Z}_r^*$	Retargeted design target
$\boldsymbol{X}(i, j)$	The entry of $\boldsymbol{X}$ given i, j index
$\boldsymbol{X}(i_1 : i_2, j_1 : j_2)$	A sub-block of $\boldsymbol{X}$ given i_1, i_2, j_1, j_2 index
$\boldsymbol{A} \otimes \boldsymbol{B}$	Convolution of $\boldsymbol{A}$ by $\boldsymbol{B}$
$\boldsymbol{A} \odot \boldsymbol{B}$	Element-wise product between $\boldsymbol{A}$ and $\boldsymbol{B}$
$\boldsymbol{A} \oplus \boldsymbol{B}$	Dilation of $\boldsymbol{A}$ by the structuring element $\boldsymbol{B}$
$\boldsymbol{A} \ominus \boldsymbol{B}$	Erosion of $\boldsymbol{A}$ by the structuring element $\boldsymbol{B}$
$\mathcal{F}(\boldsymbol{A})$	The Fourier transform of $\boldsymbol{A}$

encourage further research into curvilinear ILT solutions, we introduce a new GPU-accelerated ILT algorithm that: 1) improves upon existing algorithms to achieve better optimality, and 2) addresses the challenges of curvilinear mask writing using differentiable morphological operators. Our major contributions include:

- We thoroughly analyze the limitations of existing academic ILT algorithms and have developed a new algorithm that improves optimization convergence and enhances mask quality.
- We develope the idea of curvilinear design retargeting to allow ILT solvers to optimize toward corner-smoothed targets leading to faster and better convergence.
- We introduce a differentiable morphological operator that can be seamlessly integrated into legacy ILT algorithms to control mask curvature and shape without compromising the final quality of results (QoR).
- We conduct experiments on layers from both real-world and synthetic designs, demonstrating the superior performance of our algorithm.

Reminder of the manuscript is organized as follows: Section 2 introduces related works and fundamental terminologies associated with mask optimization and ILT.; Section 3 provides a detailed description of the proposed ILT algorithm.; Section 4 presents a comprehensive analysis of the experimental results for our algorithm across various design layers; and Section 5 discusses future work and concludes the paper.

2 Preliminaries

2.1 Notations

Throughout the paper, we use lowercase letters for scalar (e.g. x), bold lowercase letters for vector (e.g. $\boldsymbol{x}$) and bold uppercase letters for matrix (e.g. $\boldsymbol{X}$). Specifically, we use $\boldsymbol{X}(i, j)$ for single entry index and $\boldsymbol{X}(i_1 : i_2, j_1 : j_2)$ for block index. A full list of notations and symbols is shown in Table 2.

2.2 Forward and Inverse Lithography

Forward lithography simulation encompasses the fundamental mathematics of mask optimization, modeling the lithographic process through a series of approximation equations. In this context, we utilize the widely adopted Hopkin's Diffraction model along with a constant resist threshold, as described below:

$$\boldsymbol{I} = \sum_{i=1}^{k} \alpha_i ||\mathcal{F}^{-1}(\mathcal{M} \odot \mathcal{H}_i)||_2^2, \tag{1}$$

where $\mathcal{M} = \mathcal{F}(\boldsymbol{M})$ represents the rasterized mask image $\boldsymbol{M}$ in the Fourier domain, $\mathcal{H}_i$'s are lithography transmission kernels which are pre-computed given lithography system configurations, and $\boldsymbol{I}$ is the aerial image which contains the light intensity projected on the resist material. Through constant resist threshold modeling, we can derive the final resist image as follows:

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

Thus, the ILT problem can be defined as finding the mask $\boldsymbol{M}$ such that the resulting resist pattern $\boldsymbol{Z}$ closely matches the intended design $\boldsymbol{Z}^*$. In practical manufacturing, the lithographic system may not operate under ideal conditions, leading to process variations where the resist pattern is either larger (Z_{outer}) or smaller (Z_{inner}) than theoretically predicted. A robust ILT algorithm aims to generate a mask that minimizes these process variations.

Under our assumption, we have $\boldsymbol{M}, \boldsymbol{Z} \in \{0, 1\}$. However, to make the mask optimization end-to-end differentiable during ILT, a practice is to convert the binary mask to a continuous tone mask through the sigmoid function,

$$\boldsymbol{M}_c = \frac{1}{1 + \exp[-\beta_1(\boldsymbol{M} - M_s)]}, \tag{3}$$

where β_1 is called the mask steepness and M_s is a shift parameter and is usually set to 0.5 that yields better SRAF generation [8]. Similarly, the resist image is also converted to the continuous domain,

$$\boldsymbol{Z} = \frac{1}{1 + \exp[-\beta_2(\boldsymbol{I} - D_{\text{th}})]}. \tag{4}$$

Accordingly, ILT can be mathematically formulated as follows:

$$\min_{\boldsymbol{M}} f(\boldsymbol{M}, \boldsymbol{Z}^*), \tag{5}$$

$$\text{s.t. Equations (1), (3) and (4),} \tag{6}$$

where f is some objective function to satisfy ILT QoR requirements.

2.3 Morphological Operator

Morphological operator [10] started to get noticed in the early 1980s as a theory and technique for processing geometrical structures. The basic operators include erosion, dilation, opening and closing, which poses different processing effects on binary geometrical images [1]. Following tradition, we use the following notations to

[1] Morphological operators are also applicable on other formats like grayscale images and graphs, which are beyond the scope of this paper.

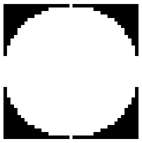
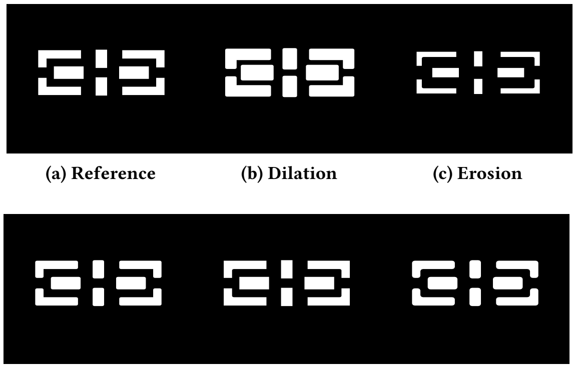


Figure 1: Example disc-shaped structuring element with size 39×39.



(a) Reference (b) Dilation (c) Erosion

(d) Opening (e) Closing (f) CDR

Figure 2: Visualization of morphological operators with eclipse structural element applied on binary images. (a) The original design image with Manhattan shapes; (b) Dilation enlarges the shapes in the original image; (c) Erosion etches the original image that yields smaller shapes; (d) Opening rounds the convex corners of each shape; (e) Closing rounds the concave corners of each shape; (f) CDR rounds both the convex and concave corners of each shape.

represent basic morphological operators:

$$f_e(\boldsymbol{A},\boldsymbol{B}) = \boldsymbol{A} \ominus \boldsymbol{B}, \ \text{(Erosion)} \tag{7}$$

$$f_d(\boldsymbol{A},\boldsymbol{B}) = \boldsymbol{A} \oplus \boldsymbol{B}, \ \text{(Dilation)} \tag{8}$$

$$f_o(\boldsymbol{A},\boldsymbol{B}) = f_d(\boldsymbol{A},\boldsymbol{B}) \ominus \boldsymbol{B}, \ \text{(Opening)} \tag{9}$$

$$f_c(\boldsymbol{A},\boldsymbol{B}) = f_e(\boldsymbol{A},\boldsymbol{B}) \oplus \boldsymbol{B}, \ \text{(Closing)} \tag{10}$$

where $\boldsymbol{A}$ is the input and B is the predefined structuring element. In this paper we use a disc shape as the structuring element, as exemplified in Figure 1. The effects of four basic morphological operators using eclipse structuring element are depicted in Figure 2, where *Dilation* of $\boldsymbol{A}$ by $\boldsymbol{B}$ results in the locus of $\boldsymbol{B}$ when the center of $\boldsymbol{B}$ traverses inside $\boldsymbol{A}$, and *Erosion* of $\boldsymbol{A}$ by $\boldsymbol{B}$ gives the locus of the center of $\boldsymbol{B}$ when $\boldsymbol{B}$ traverses inside $\boldsymbol{A}$. *Opening* and *Closing* can be derived from combinations of dilation and erosion. By definition, we can observe several interesting properties of morphological operators:

(1) Erosion removes shapes smaller than the structuring element.
(2) Dilation merges shapes if their closest distance is smaller than the half diameter of the structuring element.
(3) Opening and Closing do not modify the critical dimension of Manhattan shapes with properly set structuring elements.

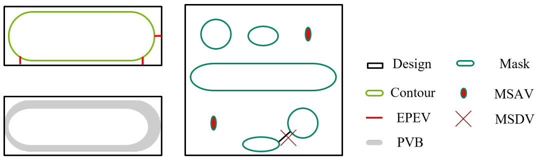


Figure 3: Mask optimization evaluation metrics. EPE violation (EPEV) and PVB are two major measurement in terms of mask lithography quality. We also employ mask shape area violation (MSAV) and mask shape distance violation (MSDV) to represent the curvilinear mask rules.

2.4 Evaluation

We follow the standard mask evaluation metrics to measure the performance of our ILT algorithm, which include edge placement error violation (EPEV) count and the process variation band area (PVB) (Figure 3).

Definition 1 (EPEV)**.** Edge placement error (EPE) quantifies the distance between the edge of the target design and the edge of the actual printed feature on the wafer. When this distance exceeds a certain threshold, typically a few nanometers, the design is at risk of failing. Each instance where this threshold is surpassed is referred to as an EPE violation. A well-optimized mask should minimize the occurrence of these EPE violations as much as possible.

Definition 2 (PVB)**.** The process variation band (PVB) illustrates how the printed wafer image fluctuates due to variations in the manufacturing process. A common approach to quantitatively assess this variation involves perturbing simulation parameters related to system settings, such as the lens focus plane and UV dose strength. The area between the innermost and outermost contours of the printed image represents the PVB. A smaller PVB indicates greater robustness of the mask against process variations.

We aim to incorporate additional metrics related to the mask shape itself. Although the definitive guidelines for curvilinear masks are still being developed [11–13], we introduce the isolated minimum shape area and minimum shape distance as key considerations for mask manufacturability. These metrics are commonly anticipated in the context of curvilinear mask design.

Definition 3 (MSA)**.** The minimum shape area is the area in terms of nm^2 of the isolated shapes in the mask tile. Smaller isolated islands will easily cause process variations and mask manufacturing challenges. MSA is mathematically given by:

$$\text{MSA} = \min \sum_{(i,j)\in\mathcal{S}_k} \boldsymbol{M}(i,j), \ \forall k = 0,1,...,N-1, \tag{11}$$

where $\mathcal{S}_k$ denotes the k^{th} isolated shape in the mask $\boldsymbol{M}$ and N is the total number of the isolated shapes.

Definition 4 (MSD)**.** Similar to the minimum shape area (MSA), we want to avoid any two shapes being too close to each other. Therefore, we use the minimum shape distance to quantify this spacing.

$$\text{MSD} = \min \text{dist}(\mathcal{S}_i, \mathcal{S}_j), \ \forall i,j = 0,1,...,N-1, \tag{12}$$

where

$$\mathrm{dist}(\mathcal{S}_i, \mathcal{S}_j) = \min \sqrt{(m-p)^2 + (n-q)^2}, \qquad (13)$$
$$\forall (m,n) \in \mathcal{S}_i, (p,q) \in \mathcal{S}_j.$$

3 Algorithm

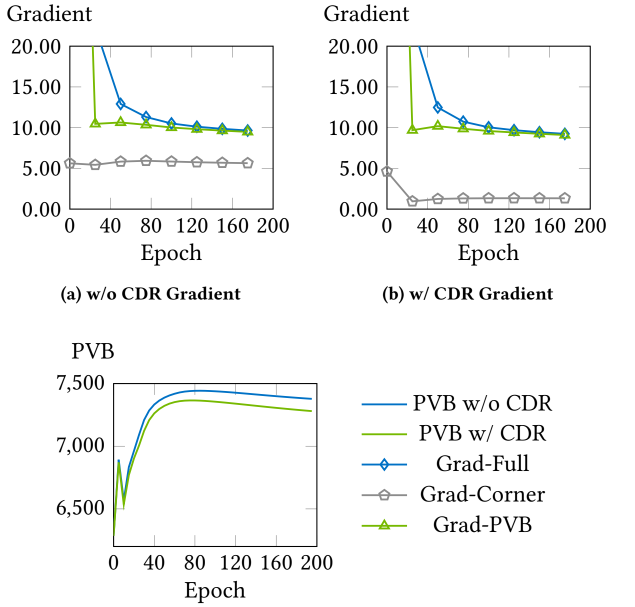
This section outlines our GPU-accelerated curvilinear ILT algorithm. The key innovations include design retargeting and a differentiable morphological operator. These advancements are followed by the final ILT algorithm, which incorporates an enhanced optimization strategy compared to previous approaches.

3.1 Design Retargeting

3.1.1 Lithography Is Low Pass Filter It is a well-established fact that the lithography process functions as a low-pass filter. This is mathematically represented by the complex lithography kernels $\boldsymbol{H}_i$ in Equation (1). Each entry in $\boldsymbol{H}_i$ corresponds to coefficients that affect the entire spectrum of the mask. However, the lithography kernel primarily captures information at locations associated with lower-frequency components of the mask. Consequently, high-frequency details, such as corners, are not effectively transferred to the silicon. As a result, optimizing a mask solely for Manhattan-shaped corners is not feasible and can lead to suboptimal performance at other critical locations, ultimately compromising the process windows.

3.1.2 Curvilinear Design Retargeting In edge-based Optical Proximity Correction (OPC), a common industrial practice involves adjusting the placement of critical dimension error (EPE) measurement points to better guide the OPC process. Rather than positioning these measurement points directly on the polygon edge near corners—where they may lead to over-optimization—we instead relocate them either inside the shape at convex corners or outside the shape at concave corners. This adjustment helps the contour align more effectively with the target during OPC, thereby reducing the risk of over-optimization and enlarging process windows [14].

While the primary goal of inverse lithography technology (ILT) is to optimize the entire design image rather than just sampled control points, discrepancies between the resist and design can still result in significant gradients during optimization. Figure 4(a) illustrates the gradients produced by the objective functions change along the optimization process. At a later training stage, we can observe that the gradients produced by the mismatch between polygon corner regions (≈0.08% of the total design area) still contribute 50% of the overall gradient value. Consequently, the optimizer may focus excessively on objectives that are unattainable, leading to a negative impact on overall process variation. Additionally, the conventional approach of corner retargeting by moving EPE measurement points proves inadequate for pixel-based optimization objectives. To address this issue, we introduce a method called curvilinear design retargeting (CDR). This technique smooths the original Manhattan design corners into curvilinear shapes, using the retargeted design as the objective for ILT optimization. The detailed algorithm is presented in Algorithm 1, where we apply opening and closing operation on the original Manhattan design respectively (lines 3–4), followed by merging the morphological effects together (line 5). An example can be found in Figure 2(f).



(a) w/o CDR Gradient

(b) w/ CDR Gradient

(c) PVB Loss

Figure 4: Corner pixel mismatch generates over 20% of the gradients during optimization. Design retargeting avoids over optimization on objectives that are unattainable.

```
Algorithm 1 CDR
Input: Manhattan design $Z^*$, convex corner smoothness coefficient $k_{\mathrm{cvx}}$, concave corner smoothness coefficient $k_{\mathrm{ccv}}$.
Output: Corner retargeted design.
1: $\boldsymbol{B}_{\mathrm{cvx}} \leftarrow$ Disc-shaped structuring element with size $k_{\mathrm{cvx}}$;
2: $\boldsymbol{B}_{\mathrm{ccv}} \leftarrow$ Disc-shaped structuring element with size $k_{\mathrm{ccv}}$;
3: $Z^*_{\mathrm{cvx}} \leftarrow Z^* \oplus \boldsymbol{B}_{\mathrm{cvx}} \ominus \boldsymbol{B}_{\mathrm{cvx}}$;
4: $Z^*_{\mathrm{ccv}} \leftarrow Z^* \ominus \boldsymbol{B}_{\mathrm{ccv}} \oplus \boldsymbol{B}_{\mathrm{ccv}}$;
5: $Z^* \leftarrow Z^*_{\mathrm{cvx}} + Z^*_{\mathrm{ccv}} - Z^*$.
```

Note that CDR only modifies the vertex region of each polygon without touching the critical dimensions.

3.2 Differentiable Morphological Operator

Mask rules in ILT are typically addressed through post-processing [8], where small artifacts generated by ILT are manually removed. However, this approach inevitably leads to a loss of optimality, as will be demonstrated in the results section. To mitigate this issue, it is necessary to develop specific algorithms that can handle mask rule cleaning during the optimization process. Fortunately, we have observed that the properties of morphological operators align, to some extent, with the requirements of curvilinear mask rules. To leverage the benefits of morphological operators in ILT, these operators must be implemented in a differentiable manner. We present the forward computing of dilation and erosion in Algorithm 2, which resembles regular convolution operation except that the summation over the sliding window is replaced with min (erosion) or max (dilation). We follow the tradition in Pytorch to

Algorithm 2 Differentiable Morphological Operator

```
function Dilation_Forward($\boldsymbol{A} \in \mathbb{R}^{N\times N}, \boldsymbol{B} \in \mathbb{R}^{k\times k}$)
    $\boldsymbol{A}^+ \leftarrow$ ZeroPadding($\boldsymbol{A}, \lfloor \frac{k}{2} \rfloor$);
    for each thread $i, j \in [0, N-1]$ do
        $\boldsymbol{A}(i,j) = \max(\boldsymbol{A}^+(i:i+k, j:j+k) \odot \boldsymbol{B})$;
    return $\boldsymbol{A}$.
function Erosion_Forward($\boldsymbol{A} \in \mathbb{R}^{N\times N}, \boldsymbol{B} \in \mathbb{R}^{k\times k}$)
    $\boldsymbol{A}^+ \leftarrow$ ZeroPadding($\boldsymbol{A}, \lfloor \frac{k}{2} \rfloor$);
    for each thread $i, j \in [0, N-1]$ do
        $\boldsymbol{A}(i,j) = \min(\boldsymbol{A}^+(i:i+k, j:j+k) \odot \boldsymbol{B})$;
    return $\boldsymbol{A}$.
function Opening_Forward($\boldsymbol{A} \in \mathbb{R}^{N\times N}, \boldsymbol{B} \in \mathbb{R}^{k\times k}$)
    $\boldsymbol{A} \leftarrow$ Dilation_Forward(Erosion_Forward($\boldsymbol{A}, \boldsymbol{B}$), $\boldsymbol{B}$);
    return $\boldsymbol{A}$.
function Closing_Forward($\boldsymbol{A} \in \mathbb{R}^{N\times N}, \boldsymbol{B} \in \mathbb{R}^{k\times k}$)
    $\boldsymbol{A} \leftarrow$ Erosion_Forward(Dilation_Forward($\boldsymbol{A}, \boldsymbol{B}$), $\boldsymbol{B}$);
    return $\boldsymbol{A}$.
```

handle the gradients of min and max. The opening and closing can then be achieved following Equation (9) and Equation (10).

3.3 CurvyILT

In this section, we will delve into our CurvyILT algorithm, detailing how it addresses the challenges of ILT and improves upon previous solutions. We will also discuss its expansion into a comprehensive full-chip solver.

3.3.1 The Objectives The forward lithography process is well-established with precise mathematical formulations. The remaining components of the ILT solver involve the optimization objectives outlined in Equation (5). In line with conventional approaches, we specifically optimize the resist image mismatch under nominal conditions, along with the PVB area, with the former being closely linked to the EPE measurement. We also introduce a term that implicitly improves the mask smoothness, noticing that rule-violating artifacts and notches generated by ILT algorithms are usually related to the perturbation in the high-frequency components of the mask. Therefore, we define,

$$f(\boldsymbol{M}, \boldsymbol{Z}^*) = ||\boldsymbol{Z}_{\text{nom}} - \boldsymbol{Z}^*||_2^2 + ||\boldsymbol{Z}_{\text{max}} - \boldsymbol{Z}_{\text{min}}||_2^2 + \beta_3 ||\mathcal{F}(\boldsymbol{M})(k:; k:)||_2^2, \tag{14}$$

where $\boldsymbol{Z}_{\text{nom}}, \boldsymbol{Z}_{\text{min}}, \boldsymbol{Z}_{\text{max}}$ are resist image computed through Equations (1) and (4) under different process conditions (controlled by $\mathcal{H}_i$'s), $\mathcal{F}(\boldsymbol{M}_c)(k:; k:)$ is the Fourier transform of the continuous mask image dropping k smallest frequency modes, and β_3 is a manually determined parameter (at the scale of $1e-3$) that controls the mask's smoothness. With the continuous and differentiable $f(\boldsymbol{M}, \boldsymbol{Z}^*)$, we are able to solve ILT through a gradient-based approach. We use Adam optimizer for better convergence.

3.3.2 CurvyILT Solver The detailed CurvyILT algorithm is elaborated in Algorithm 3, which can be viewed in three phases. The first phase is design preprocessing including curvilinear design retargeting (line 1) and mask initialization (lines 2–3). The second phase covers the major optimization steps and we update the mask iteratively till reach the maximum optimization steps (lines 5–16).

Algorithm 3 CurvyILT

```
Input: Rasterized design target $\boldsymbol{Z}^* \in \{0,1\}$, maximum optimization steps $T$, mask steepness $\beta_1$, resist steepness $\beta_2$, mask smoothness $\beta_3$, learning rate $\lambda$, $k_{\text{cvx}}$, $k_{\text{ccv}}$, mask resolution scaling factor $s$, morphological operator kernel $k_{\text{morph}}$, mask binarization threshold $M_s$, morphological gradient step $t_{\text{morph\_step}}$, iteration number morphological gradient taking effect $t_{\text{morph}}$;
Output: Optimized mask $\boldsymbol{M}$;
1:  $\boldsymbol{Z}_r^* \leftarrow$ CDR($\boldsymbol{Z}^*$,$k_{\text{cvx}}$,$k_{\text{ccv}}$);    ▷ Algorithm 1
2:  $\boldsymbol{Z}_s^* \leftarrow$ Design target $\boldsymbol{Z}_r^*$ downsample by $s$;    ▷ Adjust rasterized design resolution to balance speed and performance
3:  $\boldsymbol{M} \leftarrow \boldsymbol{Z}_s^*$;    ▷ Mask initialized as design target
4:  $\boldsymbol{B}_{\text{morph}} \leftarrow$ Disc-shaped structuring elements with size $k_{\text{morph}}$;
5:  $\boldsymbol{C}_{\text{morph}} \leftarrow$ Disc-shaped structuring elements with size $s \times k_{\text{morph}}$;
6:  for t=1,2,...,T do
7:      $\boldsymbol{M}_c \leftarrow \frac{1}{1+\exp[-\beta_1(\boldsymbol{M} - M_s)]}$;
8:      if $t > t_{\text{morph}}$ and $t \% t_{\text{morph\_step}} = 0$ then    ▷ Cleaning mask artifacts through differentiable morphological operator.
9:          $\boldsymbol{M}_{\text{open}} \leftarrow$ Opening_Forward($\boldsymbol{M}_c, \boldsymbol{B}_{\text{morph}}$);
10:         $\boldsymbol{M}_{\text{close}} \leftarrow$ Closing_Forward($\boldsymbol{M}_c, \boldsymbol{B}_{\text{morph}}$);
11:         $\boldsymbol{M}_c \leftarrow \boldsymbol{M}_{\text{open}} + \boldsymbol{M}_{\text{close}} - \boldsymbol{M}_c$;
12:     $\boldsymbol{Z}_{\text{nom}}, \boldsymbol{Z}_{\text{max}}, \boldsymbol{Z}_{\text{min}} \leftarrow$ LithoSim($\boldsymbol{M}_c$);    ▷ eqs. (1) and (4).
13:     $f(\boldsymbol{M}_c, \boldsymbol{Z}_s^*) \leftarrow$ Compute the current loss w.r.t. the current mask;
14:     $\boldsymbol{G} \leftarrow$ AdamOpt($\nabla_f \boldsymbol{M}, \lambda$);    ▷ Compute the actual gradient step to update the mask, we discovered Adam optimizer brings best optimization quality.
15:     $\boldsymbol{M} \leftarrow \boldsymbol{M} - \boldsymbol{G}$;
16: $\boldsymbol{M} \leftarrow$ Interpolation($\boldsymbol{M}$, scale_factor = $s$, anti_alias = True, mode = "bicubic");    ▷ Scale the mask back to desired resolution.
17: $\boldsymbol{M}(i,j) \leftarrow 1, \forall \boldsymbol{M}(i,j) > M_s$;    ▷ Mask processing.
18: $\boldsymbol{M}(i,j) \leftarrow 0, \forall \boldsymbol{M}(i,j) \leq M_s$;
19: $\boldsymbol{M}_{\text{open}} \leftarrow$ Opening_Forward($\boldsymbol{M}, \boldsymbol{C}_{\text{morph}}$);
20: $\boldsymbol{M}_{\text{close}} \leftarrow$ Closing_Forward($\boldsymbol{M}, \boldsymbol{C}_{\text{morph}}$);
21: $\boldsymbol{M} \leftarrow \boldsymbol{M}_{\text{open}} + \boldsymbol{M}_{\text{close}} - \boldsymbol{M}_c$;
22: $\boldsymbol{M} \leftarrow$ Opening_Forward($\boldsymbol{M}, \boldsymbol{C}_{\text{morph}}$);
23: $\boldsymbol{M} \leftarrow$ Closing_Forward($\boldsymbol{M}, \boldsymbol{C}_{\text{morph}}$).
```

During each optimization step, we heuristically apply morphological operators on the mask image to perform corner smoothing (lines 8–10) and remove artifacts (lines 11–12). We employ the Adam optimizer to compute the true mask gradients and mask update steps for best practices (lines 13–16). The last phase can be deemed as post-processing where the mask is scaled back to the desired resolution through interpolation and binarized with a preset threshold (lines 17–19).

3.4 Discussion

Compared to previous methods, the proposed algorithm offers **faster convergence** by utilizing curvilinear design retargeting and delivers **improved mask quality** through the application of differentiable morphological operators in accordance with mask

Table 3: Benchmark Statistics.

Benchmark	Layer	Statistic
ICCAD13	Metal	10
LithoBench	Metal	271
	Via	165

design rules. Additionally, by addressing ILT-generated artifacts during the optimization process, our approach eliminates the need for unnecessary post-processing, preserving the optimized resist image quality. While earlier efforts have employed strided average pooling to smooth the mask shape, this occurs after each iteration's mask binarization step, leading to inaccuracies in the forward lithography calculations.

4 Experiments

4.1 Dataset and Configurations

To assess the algorithm's performance, we utilize the widely recognized benchmark suites from the ICCAD13 CAD Contest (ICCAD13) [15] and the more recent LithoBench dataset [16]. The details of these benchmarks are summarized in Table 3. The ICCAD13 benchmark consists of ten $2\mu m \times 2\mu m$ clips featuring M1 layer polygons. Meanwhile, LithoBench is a larger dataset originally developed for AI applications, comprising over 10,000 metal and via layer clips. For demonstration purposes, we focused on the standard cell collection within LithoBench, which includes 271 metal layer designs and 165 via layer designs. We implement the algorithm using Pytorch with CUDA support and adopt the EPE checker from NeuralILT [17]. To compare our approach with prior arts, we also implemented the multi-level ILT [8] that reproduces the original results as closely as possible. All experiments are conducted on a single NVIDIA RTX A6000 platform with 48GB memory.

4.2 Comparison with State-of-the-art

In this first experiment, we compare our method with state-of-the-art ILT solvers on ten ICCAD13 clips as shown in Table 4, where columns "MSE", "PV", "EPE", "MSA", and "MSD" denote the mean square error between the resist image and the design, PVBand area, EPE violation count, minimum shape area (nm^2) and minimum shape distance (nm), respectively; column "A2-ILT [6]" lists the results of ILT solve with the gradient regularization of mask image attention map; column "MultiILT [8] Ref" corresponds to the results of the SOTA academic GPU-accelerated ILT solver that takes advantage of lithography simulation at different rasterization resolutions; column "MultiILT [8] Reimp" corresponds to the author re-implemented MultiILT (exact) as we need to reproduce the mask results to measure certain mask rules and perform additional studies; column "Ours" lists our full algorithm implementation results with all techniques enabled.

We performed a comprehensive parameter search of our implementation on MultiILT to align the reported results, with only a minor discrepancy observed in case 3, which does not impact our overall conclusions. Our findings demonstrate that our method significantly outperforms existing approaches in terms of MSE, PVB, and EPE. Notably, in case 3—a particularly complex high-density clip—our solution achieved a reduction in EPE violations to below 20 for the first time. On average, our method lowers the EPE violation count by 30% and the resist MSE by 8.6%, while maintaining a favorable process window. Additionally, our approach exhibits notable improvements in curvilinear mask complexity, offering larger minimum shape areas and distances, thus enhancing its potential for curvilinear ILT solver development. It is important to mention that the MultiILT results are subject to post-processing, where minor artifacts are manually removed, which may affect optimality. Table 5 also lists the optimization result on the standard cell collection in LithoBench. Similarly, on more practical design layers, our solver outperforms prior work by an even larger amount with 2× smaller EPE violation count and 11% PV area reduction.

We have adjusted our post-processing configurations to ensure that the final quality of results for MultiILT [8] aligns closely with the values reported in the original manuscript. Furthermore, we compare the efficiency of different solvers in Table 6, with the first row presenting throughput (seconds per clip) for three recent ILT solvers. Our algorithm demonstrates superior efficiency in terms of both runtime and peak memory usage during the optimization runtime. Notably, MultiILT requires ten times more GPU memory than our approach due to its memory-intensive high-resolution phase, which our algorithm effectively avoids.

4.3 Post Processing Sabotages Optimality

In the second experiment, we investigate the effects of post processing in MultiILT [8] and demonstrate the necessity of mask cleansing during optimization runtime. As shown in Table 7, column "MultiILT w/o PP" lists the optimization results without post-processing, and column "MultiILT w/ PP" corresponds to the optimization results with post-processing. We observe a substantial degradation in results when ILT-generated artifacts are removed, highlighting a critical trade-off between mask complexity and ILT quality of results (QoR). In contrast, our approach manages mask complexity throughout the optimization process and eliminates the need for post-processing, thereby preserving optimality.

4.4 Ablation Study

In this experiment, we will demonstrate the effectiveness of each techniques proposed in this paper through ablation study. Again, we use the ten ICCAD13 clips as example for simplicity. We show the ILT results change with different techniques being included gradually in Table 8. Compared to the baseline implementation, CDR avoids excessive optimization on polygon corners, encourages sub-resolution assist feature (SRAF) growth and brings better process window (1.5% PV reduction) while inducing small artifacts that challenges mask manufacturing (51% MSA degradation).

Morphological operators enhance the ILT process in several key ways: 1) They eliminate small artifacts generated during ILT, resulting in a final mask with a larger minimum shape area (58% improvement in MSA). 2) The frequent removal of small shapes during optimization provides regularization and introduces stochastic effects, leading to a slight improvement in EPE violations in case 3, although this benefit is marginal.

Table 4: Result comparison on `ICCAD13` with state-of-the-art ILT solvers.

Case	A2-ILT [6]			MultiILT [8] Ref			MultiILT [8] Reimp					Ours				
	MSE	PV	EPE	MSE	PV	EPE	MSE	PV	EPE	MSA	MSD	MSE	PV	EPE	MSA	MSD
1	45824	59136	7	38495	47015	3	39533	44887	3	832	1	38066	44447	3	1062	16
2	33976	52054	3	28173	37555	0	32516	37374	0	640	12	28623	36914	0	2029	28
3	94634	82661	62	67949	69361	22	65315	75011	23	768	9	61650	70580	15	981	10
4	20405	29435	2	10307	21514	0	9099	21484	0	704	1	9211	21584	0	1342	10
5	37038	62068	1	28482	49683	0	30015	48696	0	896	9	27859	47870	0	865	11
6	40701	54842	2	30334	44127	0	33400	42788	0	896	9	30391	42288	0	1088	17
7	21840	48474	0	14635	36961	0	17419	36241	0	768	9	12791	34389	0	2061	12
8	14912	24598	0	11194	20985	0	11552	18987	0	640	9	11468	18649	0	621	9
9	47489	68056	2	34900	54948	0	37219	54792	0	640	1	32720	54387	0	3940	13
10	9399	20243	0	7266	16581	0	7180	14979	0	2304	9	7130	15014	0	2964	68
Avg	36621.8	50156.7	7.9	27173.5	39873.0	2.5	28324.8	39523.9	2.6	908.8	6.9	**25990.9**	**38612.2**	**1.8**	**1695.3**	**19.4**

Table 5: Result comparison on `LithoBench` with state-of-the-art ILT solvers.

Case	MultiILT [8] Ref					Ours				
	MSE	PV	EPE	MSA	MSD	MSE	PV	EPE	MSA	MSD
Metal	13814.2	24928.2	0.03	720.4	16.1	14643.3	21631.3	0.0	1709.0	24.4
Via	34813.4	39997.4	8.6	464.2	19.2	29183.8	36172.4	3.8	1072.9	11.3
Avg	24313.8	32462.8	4.31	592.3	17.6	21913.5	28901.9	1.9	1390.9	17.8

Table 6: Efficiency of state-of-the-art ILT solvers.

Solver	A2-ILT [6]	MultiILT [8]	Ours
Throughput	4.51	3.45	2.11
OptPeakMemory	-	7.2GB	0.6GB

Table 7: Post processing removes rule-violating artifacts at the cost of optimality.

Case	MultiILT w/o PP					MultiILT w/ PP				
	MSE	PV	EPE	MSA	MSD	MSE	PV	EPE	MSA	MSD
1	37976	45423	3	64	1	39533	44887	3	832	1
2	31070	37754	0	576	12	32516	37374	0	640	12
3	63036	73396	19	64	1	65315	75011	23	768	9
4	8498	21561	0	64	1	9099	21484	0	704	1
5	28478	49400	0	64	9	30015	48696	0	896	9
6	29666	43162	0	64	1	33400	42788	0	896	9
7	17333	36319	0	256	9	17419	36241	0	768	9
8	11486	19048	0	64	1	11552	18987	0	640	9
9	34459	55688	0	64	1	37219	54792	0	640	1
10	7180	14979	0	2304	9	7180	14979	0	2304	9
Avg	26918.2	39673	2.2	358.4	4.5	28324.8	39523.9	2.6	908.8	6.9

While morphological operators can impose penalties on minimum shape distance—due to the sequential closing after opening in our algorithm, which prevents nearby shapes from merging—this penalty can be effectively mitigated through design retargeting, thanks to the SRAF growing effect. With all techniques applied, our approach achieves significantly improved MSA and MSD without compromising performance.

4.5 Result Visualization

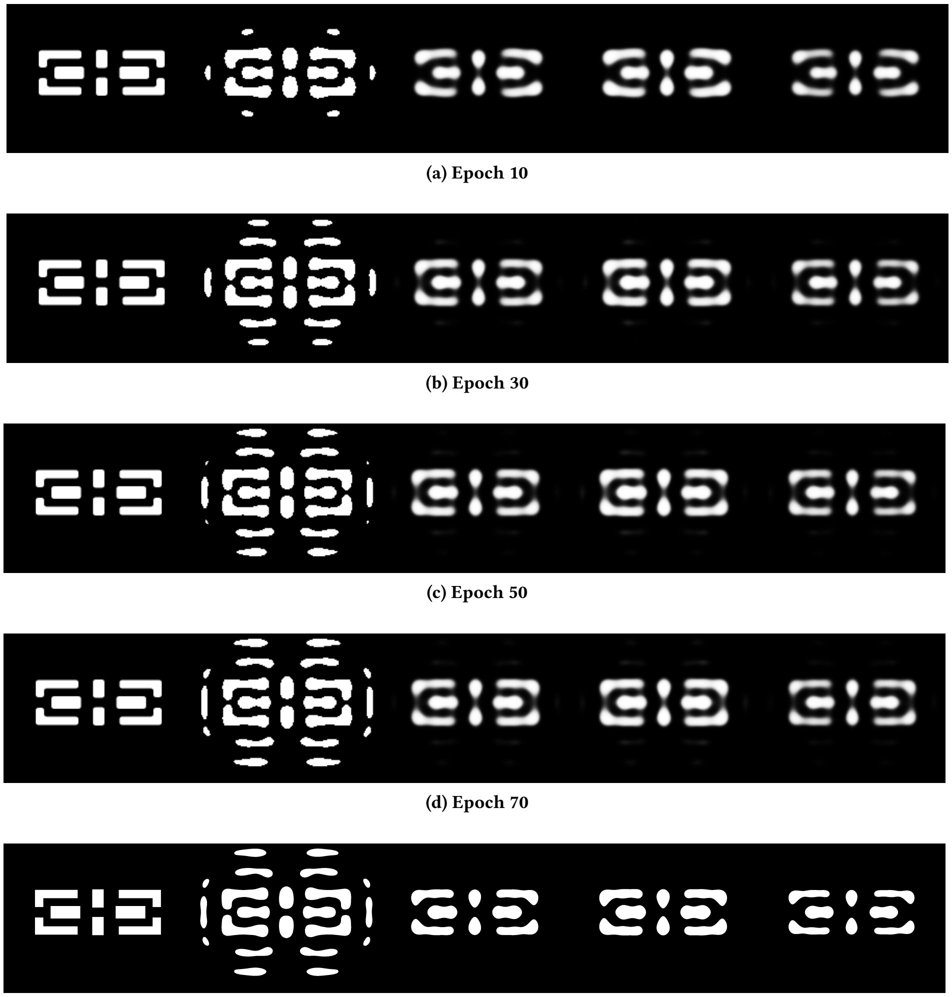
Finally, we will visualize some ILT results to illustrate the advantages of our algorithm more clearly. Here, we use a simple clip from `ICCAD13` as an example and depict the optimization trajectory. We can observe at a later optimization stage (Figure 5(c)), the morphological operators start to take effect and try to remove notches generated on the mask edge and enlarge thin mask critical dimensions. This also ensures a minimum change of the optimization status when producing the final smoothed high-resolution masks (Figure 5(e)).

5 Conclusion

In this paper, we explored GPU-accelerated ILT algorithms for curvilinear mask generation, assessing current pixel-based mask optimization methods and identifying key challenges in enhancing mask optimization for future multi-beam mask makers. We introduced innovative techniques to reduce mask complexity and enhance printability, such as curvilinear design retargeting, the incorporation of differentiable morphological operators in optimization, and the inclusion of mask smoothing objectives. These elements converge to form the curvyILT algorithm, which demonstrates superior performance against leading academic solvers across various pattern complexities. For practical application and deployment, future research will focus on covering more curvilinear mask rules, scaling to full-chip designs, and integrating AI technologies. To encourage further innovation in computational lithography and semiconductor manufacturing, we will make our source code publicly available.

Table 8: Ablation study.

Case	Ours (Baseline)					Ours (CDR)					Ours (Morph)					Ours (CDR+Morph)				
	MSE	PV	EPE	MSA	MSD	MSE	PV	EPE	MSA	MSD	MSE	PV	EPE	MSA	MSD	MSE	PV	EPE	MSA	MSD
1	35938	44969	3	1049	10	37077	44355	3	101	11	36583	44975	3	587	24	37783	44446	3	1062	16
2	27796	37311	0	504	14	28996	36959	0	28	25	27661	37314	0	2101	19	28644	36940	0	2030	28
3	62410	72333	15	320	6	60599	70559	15	82	12	60124	72046	14	999	3	62212	70545	15	979	12
4	8334	21801	0	8	2	8958	21446	0	358	11	8496	21957	0	1912	20	9193	21577	0	1346	10
5	27121	48497	0	508	8	27831	47911	0	85	13	27116	48453	0	547	8	27824	47861	0	841	11
6	29858	42777	0	1746	12	30243	42277	0	50	26	29970	42826	0	1484	19	30391	42287	0	1088	17
7	12388	34568	0	628	22	12603	34253	0	465	23	12638	34809	0	1384	1	12801	34409	0	2060	12
8	10963	18838	0	1034	26	11343	18572	0	484	30	11087	18921	0	789	23	11473	18644	0	656	9
9	31695	55250	0	98	16	32723	54399	0	36	12	31830	55296	0	843	12	32723	54393	0	3939	13
10	6743	15097	0	3217	66	7054	14951	0	2797	68	6864	15288	0	3750	66	7127	15013	0	2981	68
Avg	25324.6	39144.1	1.8	911.2	18.2	25742.7	38568.2	1.8	448.6	23.1	25236.9	39188.5	1.7	1439.6	19.5	26017.1	38611.5	1.8	1698.2	19.6



(a) Epoch 10

(b) Epoch 30

(c) Epoch 50

(d) Epoch 70

(e) Epoch Final

Figure 5: Visualization of the optimization trajectory of our algorithm. From left to right: CDR design, mask, nominal condition image, outermost image, and innermost image.

References

[1] J. Kuang, W.-K. Chow, and E. F. Y. Young, "A robust approach for process variation aware mask optimization," in *IEEE/ACM Proceedings Design, Automation and Test in Eurpoe (DATE)*, 2015, pp. 1591–1594.

[2] T. Matsunawa, B. Yu, and D. Z. Pan, "Optical proximity correction with hierarchical bayes model," *Journal of Micro/Nanolithography, MEMS, and MOEMS (JM3)*, vol. 15, no. 2, p. 021009, 2016.

[3] A. K. Wong, R. A. Ferguson, and S. M. Mansfield, "The mask error factor in optical lithography," *IEEE Transactions on Semiconductor Manufacturing (TSM)*, vol. 13, no. 2, pp. 235–242, 2000.

[4] H. Matsumoto, J. Yasuda, T. Motosugi, H. Kimura, M. Kawaguchi, Y. Kojima, H. Yamashita, M. Saito, T. Tamura, and N. Nakayamada, "Multi-beam mask writer mbm-3000 for next generation euv mask production," in *Photomask Technology 2023*, vol. 12751. SPIE, 2023.

[5] J. Sturtevant, "Curves ahead! ic manufacturing prepares for curvilinear masks," Siemens, Tech. Rep., 2023.

[6] Q. Wang, B. Jiang, M. D. Wong, and E. F. Young, "A2-ILT: GPU accelerated ILT with spatial attention mechanism," in *ACM/IEEE Design Automation Conference (DAC)*, 2022.

[7] Z. Yu, G. Chen, Y. Ma, and B. Yu, "A GPU-enabled level set method for mask optimization," in *IEEE/ACM Proceedings Design, Automation and Test in Eurpoe (DATE)*, 2021.

[8] S. Sun, F. Yang, B. Yu, L. Shang, and X. Zeng, "Efficient ilt via multi-level lithography simulation," in *ACM/IEEE Design Automation Conference (DAC)*, 2023, pp. 1–6.

[9] S. Zheng, B. Yu, and M. Wong, "Openilt: An open source inverse lithography technique framework," in *IEEE International Conference on ASIC (ASICON)*, 2023, pp. 1–4.

[10] M. L. Comer and E. J. Delp, *Morphological operations*. Boston, MA: Springer US, 1998, pp. 210–227.

[11] Y. Choi, A. Fujimura, and A. Shendre, "Curvilinear masks: an overview," *Proceedings of SPIE*, vol. 11855, pp. 157–172, 2021.

[12] R. Pearman, D. O'Riordan, J. Ungar, M. Niewczas, L. Pang, and A. Fujimura, "How utilizing curvilinear design enables better manufacturing process window," in *Proceedings of SPIE*, vol. 11328. SPIE, 2020, pp. 183–191.

[13] A.-M. Armeanu, E. Malankin, N. Lafferty, C.-I. Wei, M. K. Sears, G. Fenger, X. Zhang, W. Gillijns, D. Trivkovic, R.-h. Kim *et al.*, "Application of resolution enhancement techniques at high na euv for next generation dram patterning," in *Proceedings of SPIE*, vol. 12495. SPIE, 2023, pp. 52–63.

[14] "Calibre," https://eda.sw.siemens.com/en-US/ic/calibre-design/.

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

[16] S. Zheng, H. Yang, B. Zhu, B. Yu, and M. Wong, "Lithobench: Benchmarking ai computational lithography for semiconductor manufacturing," *Conference on Neural Information Processing Systems (NIPS)*, vol. 36, 2024.

[17] B. Jiang, L. Liu, Y. Ma, H. Zhang, E. F. Y. Young, and B. Yu, "Neural-ILT: Migrating ILT to nerual networks for mask printability and complexity co-optimizaton"," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2020.

ISPD2025-Review Comments

Reviewer 1:

2: (accept) The work is related to computational lithography. By stating that multi-beam mask writing technique enables creation of litho mask with curvy shapes, a complicated computation algorithm is proposed for curvy inverse lithography technology. In view of significant computational difficulty of optimization problem for curvy shapes, GPU acceleration is employed. The algorithm is presented; authors promise also to present source code. The work definitely can be useful for innovations in lithography.

Autors Response: We appreciate the positive feedback, source code will be available after the internal review process.

Reviewer 2:

2: (accept) The paper comes up with a GPU accelerated inverse lithography algorithm for mask optimization. IT is well written with clear definitions and illustrations. The results are impressive and the algorithm has smaller memory footprint and a faster runtime than the prior-art. A better explanation of the Fig. 5 solution evolution would help. Enclose caps in in bib to render them properly in the PDF.

Autors Response: We appreciate the positive feedback and crafting suggestions.

Reviewer 3:

0: (borderline paper) The work proposed an inverse lithography method to optimize mask image for better target image. The major new ideas are using retargeting for faster convergence; applying morphological operation repeatedly to intermediate and final mask solution for cleaning artifacts. From the paper's results and the algorithm, the morphological operation seems very effective on removing mask artifacts without affecting the quality, which is the most important contribution. However I still have several questions on the work:

1. Although the title mentions the work is a "GPU-Accelerated" method, neither how the proposed algorithm is related to GPU nor how much acceleration is achieved is mentioned. There is no runtime reported in the results.

Autors Response: Everything in the algorithm is GPU-accelerated including shape rasterization, morphological layer, gradient optimization, etc. Runtime is explicitly reported in Table 6.

2. In section 3.1.2, the authors proposed design retargeting to adjust EPE measurement points. Does this EPE adjustment also applied to the result reported? Also, citation [14] does not point to any document or article that supports the idea of retargeting.

Autors Response: If read the paper carefully, the reviewer would found that adjusting EPE measurement point is the common approach used in OPC (edge-based optimization), which however not directly applicable in ILT (freeform pixel-based). Our approach has nothing to do with the adjustment of EPE measurement point. To incorporating the benefits of OPC retargeting, we proposed the CDR algorithm.

3. The authors proposed two new metrics MSA and MSD. For MSA, the authors states that small shapes cause process variation and manufacturing difficulty. However if a certain area constraint is satisfied, relatively small area should not be a problem. Same for MSD, as long as the minimal spacing satisfies the manufacture constraint, a small spacing between shapes should not be a problem. I think these two metrics should be reported as number of mask shape area violation (MSAV) and number of mask shape distance violation (MSDV) as they are shown in Figure 3.

Autors Response: We did not setup a threshold to count the violations, as the number will be on the mask-shop's criteria, instead we show the exact number to quantitatively demonstrate the effectiveness of our proposed algorithm.

Reviewer 4:

2: (accept) - This paper presents an ILT algorithm designed for curvilinear mask optimization. The proposed approach includes design retargeting based on the observation that polygon corners tend to be rounded and optimizer may over emphasize the corner mismatch and degrade the overall optimization result. The paper also introduce morphological operator which can eliminate small features that are bad for manufacturing during optimization instead of post-processing. - The experimental result is well presented to highlight the effectiveness of the proposed design retargeting and morphological operators. Compared to existing works, the proposed approach outperform in all metrics and show better manufacturability with less small area SRAFs. - typo: Eq 13, "n-p"->"n-q" - Alg 3, what does line 19 23 try to do? There seems to be some error as M is re-assigned. - Fig 4(a), it looks the gradient produced by corner is around 5 while the full gradient is around 10; but it is claimed to contribute 20% in the paragraph. The figure and context doesn't seem to match.

Autors Response: We appreciate the positive feedback, we have revised the manuscript accordingly for better readability.