Enabling Robust Inverse Lithography with Rigorous Multi-Objective Optimization

Yang Luo
HKUST(GZ)
yluo208@connect.hkust-gz.edu.cn

Xiaoxiao Liang
HKUST(GZ)
xliang603@connect.hkust-gz.edu.cn

Yuzhe Ma*
HKUST(GZ)
yuzhema@hkust-gz.edu.cn

Abstract

Inverse lithography technology (ILT) shows great power in optical proximity correction, which enlarges the solution space of mask optimization and generates high-quality masks in terms of various criteria, including process window. Optimizing the process window involves improving the fidelity of printed wafer patterns on various process conditions. It is non-trivial to explicitly optimize the process window during ILT optimization, which is essentially a multi-objective optimization problem. In this paper, we propose a robust inverse lithography method, RMO-ILT, to optimize the process window effectively. Instead of aggregating all the objectives into a single one, we target the multi-objective optimization directly and explicitly. Specifically, we design a rigorous multi-objective optimization algorithm that computes uniform gradients during the mask optimization process. Furthermore, we improve the algorithm efficiency from both the algorithm level and implementation level to address the intrinsic increase in the computation overhead, significantly reducing the time consumption and enhancing the scalability. Experimental results show that the proposed algorithm achieves superior performance on the process window.

ACM Reference Format:
Yang Luo, Xiaoxiao Liang, and Yuzhe Ma*. 2024. Enabling Robust Inverse Lithography with Rigorous Multi-Objective Optimization . In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD '24), October 27–31, 2024, New York, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3676536.3676763

1 Introduction

Lithography has been developed for several decades in the semiconductor industry. This progress is attributed to continuous technological advancements aimed at shorter exposure wavelengths and larger numerical apertures (NA) to achieve smaller minimum printed feature sizes. However, as advanced semiconductor critical dimensions (CD) continue to decrease, the proximity effect and optical diffraction become unavoidable, resulting in a degradation of manufacturing yield [1]. As one of the widely used resolution enhancement techniques (RETs), nowadays, optical proximity correction (OPC) has developed and become essential for maintaining good printed image quality by adjusting the mask's topology to align the pattern on the wafer as close to the desired layout as possible.

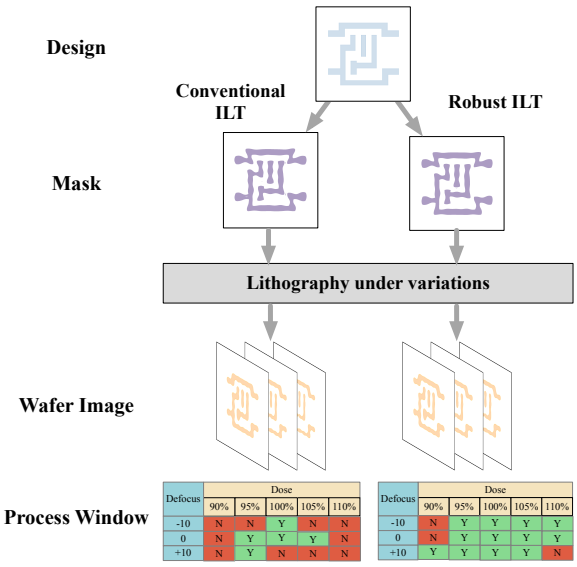

Figure 1: The mask obtained by the robust ILT algorithm has a larger process window under rigorous evaluation.

In general, OPC optimization methodologies are categorized into three classes: the rule-based approach [2], the model-based one [3,4], and inverse lithography technique (ILT) [5–7]. The rule-based approach is simple to implement but is limited to compensating for deformation effects in local features. The model-based approaches gradually adjust the segmented edges to find optimal locations. When it comes to the advanced technology node, ILT has been paid more and more attention [8]. ILT method pixelates the mask through pixel-wise function [5,9] or level-set function [10,11], treating the mask optimization as an inverse problem to find an appropriate solution. With the expansion of search space, the optimization problem becomes more challenging. To obtain an optimal solution, ILT algorithms usually employ adopt iterative methods [5,7,9] and deep learning-based techniques [6,12]. The former utilizes gradient descent to optimize the objective function, while the latter regards ILT tasks as image generation by leveraging deep neural networks [13]. For example, GANOPC [6] generates ideal masks using an adversarial training method.

Process window (PW), as one of the most important metrics in the lithography process, characterizes the robustness against the process variation (PV) by measuring the performance across a defined depth of focus (DOF) and exposure latitude (EL). Since the printed feature dimensions are highly sensitive to process variations in the low-$K1$ regime [14], it is important to design PV-aware ILT algorithms nowadays. As shown in Figure 1, even though the printed mask

*Corresponding author

ICCAD '24, October 27–31, 2024, New York, NY, USA

ACM ISBN 979-8-4007-1077-3/24/10…$15.00
https://doi.org/10.1145/3676536.3676763

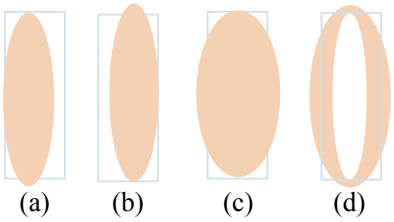

Figure 2: (a)-(c) Printed images under different process conditions. (d) Resulted PV Band.

shows good performance under the nominal condition, the mask quality deteriorates as the defocus or dose changes, with the result worsening when both parameters are altered simultaneously.

There has been considerable progress in previous ILT research. Several studies [5, 15–19] have attempted to incorporate process variation into the optimization procedure. PV-OPC [15] first proposes the process variation-aware OPC guided by variational edge placement error metrics. MOSAIC [5] tries to minimize the differences among printed masks under two extreme process conditions to optimize the process window. However, since they does not consider a wide range of process conditions, the resulting printed mask may not achieve a satisfactory process window. On the other hand, other methods [18, 19] adopt a statistical approach to obtain the optimal mask by minimizing the expected difference between printed masks under different conditions and the target pattern. However, these methods typically assume the probability density function of other corners follows a normal distribution, which is neither general nor practical. Additionally, they only consider the incorporation of defocus variation, neglecting the importance of exposure latitude variation.

These studies mostly adopt edge placement error (EPE) for mask quality evaluation and process variation band (PV band) for mask robustness evaluation. However, since the standard PV band measurement involves the XOR operation among the printed contours from all process conditions, directly optimizing the PV band is non-applicable in ILT methods which require a continuous objective function. Existing ILT evaluation mostly adopts the workaround mentioned in [20], where the PV band is assumed to be formed by an "outermost contour" and "innermost contour", and thus the PV band is just the difference between these two contours, which can be formulated as matrix subtraction. However, we argue that this assumption is not true in reality. There may not always exist two process corners that correspond to the "outermost contour" and "innermost contour" of a given mask, respectively, as shown in Figure 2. In these scenarios (also the typical scenarios), the PV band is hard to directly optimize to obtain a robust mask against PV.

To achieve a more robust inverse lithography technique in terms of the process window, we advocate for the comprehensive consideration of all process corners during a rigorous optimization procedure. Therefore, we formulate the mask optimization problem as a multi-objective optimization (MOO) problem. However, solving such a problem is non-trivial [21] due to a complicated priority balance. When searching for such a solution, it typically follows a strategy that considers the essential trade-offs between the objectives to avoid *Pareto dominated point*. Unfortunately, it is difficult to find a common descent direction to ensure full improvement for each target, especially when there are several conflicting objectives. In this work, we propose to optimize all process corners simultaneously via a **uniform gradient computation** method to resolve the gradient conflict issues. There are two key considerations of our algorithm: gradient conflict alleviation by projecting gradient onto the orthogonal direction of other conflict gradients, and amplitude balancing through dynamically selecting the magnitude of gradients to adjust the step size adaptively, which accelerates convergence speed and more importantly, ensures performance. Moreover, we also present theoretical proof to the convergence of the algorithm.

Since more process corners are explicitly considered in the rigorous optimization, more computation overhead on lithography simulation and backward gradient calculation is introduced as well compared with conventional ILT. To further address the potential scalability issues, in this work, we improve the algorithm efficiency from both the algorithm level and implementation level. At the algorithm level, we propose an objective sampling technique to reduce the computation complexity. At the implementation level, despite that the ILT algorithm is GPU-friendly, we demonstrate that the efficiency of the proposed algorithm can be further improved with a straightforward batch execution on multiple GPUs.

Our main contributions are summarized as follows:

- We proposed RMO-ILT, a rigorous multi-objective optimization for robust inverse lithography, which explicitly optimizes all process corners during mask optimization. To the best of our knowledge, it is the first time that rigorous multi-objective optimization is investigated for process window optimization.
- We designed a uniform gradient computation approach for the multi-objective optimization problem to address poor performance resulting from conflicts among multiple objectives. The convergence of the algorithm is mathematically proved.
- We improve the algorithm efficiency from both the algorithm level and implementation level to address the intrinsic increase in the computation overhead, significantly reducing the time consumption and enhancing the scalability of our algorithm.
- Experimental results show that the proposed algorithm achieves a substantial improvement in the process window compared with other ILT methods.

The rest of the paper is organized as follows. Section 2 shows some preliminaries. Section 3 gives the details of Rigorous-ILT and discusses the theoretical convergence of our algorithm. Section 4 introduces our acceleration implementation strategies and Section 5 presents experimental results, followed by a conclusion in Section 6.

2 Preliminary

2.1 Lithography Process and Model

A typical optical lithography comprises four basic elements: a light source (an illuminator), a condenser lens, a mask on the reticle plane, an objective lens (exposure system) and a silicon wafer [22]. The mathematical formulation of the forward lithography process generally adopts the Hopkins model [23], which can be approximated by the singular value decomposition (SVD) model [24]. In computing the intensity map $\mathbf{I}(x, y)$, the SVD model decomposes the optical system into a set of coherent kernels, as revealed by Equation (1):

$$\mathbf{I}(x,y) = \sum_{k=1}^{K} w_k \mid \boldsymbol{h}_k(x,y) \otimes \mathbf{M}(x,y) \mid^2, \tag{1}$$

where $\mathbf{M}$ indicates the mask, $\boldsymbol{h}_k$ is the k^{th} kernel, and w_k is its weight in the coherent system. The photoresist is then exposed to the light, and an image is developed where the light intensity exceeds a threshold I_{th} as follows:

$$\mathbf{Z}(x,y) = f_{resist}(\mathbf{I}) = \begin{cases} 0, & \mathbf{I}(x,y) \le I_{th}, \\ 1, & \text{otherwise}, \end{cases} \tag{2}$$

where $\mathbf{Z}(x,y)$ denotes the wafer pattern.

We can infer from Equations (1) and (2) that the wafer pattern mainly depends on the kernel functions $\boldsymbol{h}_k$ with a given input mask. In the forward lithography process, the kernel values are affected by process variations which mainly involve *dose variation* and *defocus*. It is widely observed that the wafer image is perturbed with dose and focus variations, resulting in yield fluctuations [25].

2.2 Inverse Lithography Technique

Given the lithography model, the objective of standard ILT is to find the optimal input mask $\mathbf{M}^*$ by solving an inverse imaging problem to ensure the resulting printed wafer image $\mathbf{Z}$ closely matches the desired target pattern $\mathbf{Z}_t$. A general approach is to minimize the mean square error (MSE) of printed wafer images under nominal conditions and target layout:

$$\min_{\mathbf{M}} \quad \|\mathbf{Z} - \mathbf{Z}_t\|_2^2 \tag{3}$$

$$\text{s.t.} \quad \mathbf{Z} = f_{resist}(\mathbf{I}) \tag{4}$$

$$\mathbf{I} = \sum_{k=1}^{K} w_k \mid \boldsymbol{h}_k \otimes \mathbf{M} \mid^2 \tag{5}$$

$$\mathbf{M}(x,y) \in \{0,1\}. \tag{6}$$

However, as we mentioned above, in practice, the mask performance would be influenced by exposure dose variation and defocus. The real image plane may deviate from the nominal focal plane and the exposure dose varies, which would drastically change the dimension of features. Therefore, it is essential to design the ILT algorithm to accommodate process variations.

2.3 Problem Formulation

Given a target layout $\mathbf{Z}_t$, the objective of robust inverse lithography technique is to obtain a mask $\mathbf{M}^*$ by solving the following N-dimensional multi-objective optimization problem:

$$\begin{aligned} \min_{\mathbf{M}} \quad & \mathcal{L}(\mathbf{M}) = [\mathcal{L}_1(\mathbf{M}), \mathcal{L}_2(\mathbf{M}), ..., \mathcal{L}_N(\mathbf{M})]^T \\ \text{s.t.} \quad & \mathcal{L}_n(\mathbf{M}) = \|\mathbf{Z}_n - \mathbf{Z}_t\|_2^2, \quad \forall n \in \{1,2,...,N\}, \\ & \mathbf{Z}_n = f_{resist}(\mathbf{I}_n), \quad \forall n \in \{1,2,...,N\}, \\ & \mathbf{I_n} = \sum_{k=1}^{K} w_k \mid \boldsymbol{h}_{n_k} \otimes \mathbf{M} \mid^2, \quad \forall n \in \{1,2,...,N\}, \\ & \mathbf{M}(x,y) \in \{0,1\}, \end{aligned} \tag{7}$$

where $\boldsymbol{h}_{n_k}$ denotes the k-th kernel in the n-th process condition. Each objective function $\mathcal{L}_n(\mathbf{M})$ denotes the difference between the target pattern $\mathbf{Z}_t$ and the printed wafer $\mathbf{Z}_n$ under the n-th process condition.

3 Proposed Method

3.1 Issues of Conventional Gradient Calculation

Solving such numerical optimization problems generally adopts a gradient-based method. However, constraints like Equations (4) and (6) in the conventional ILT formulation are non-differentiable, which requires relaxation before the gradient calculation.

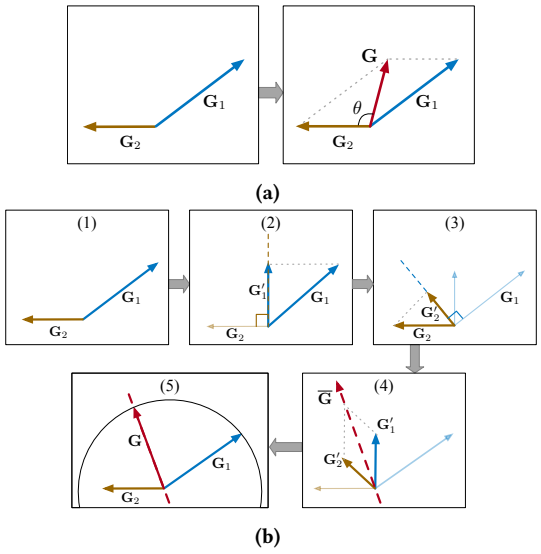

Figure 3: The final update vector G (in red) of a two-objective optimization with (a) standard gradient accumulation (GA) and (b) uniform gradient computation. For GA, the gradient angle $\theta > 90^\circ$ leads to conflict and the gradient with larger amplitude dominates the direction, while the uniform gradient G finds a good balance between two directions.

First, we consider the relaxation for Equation (6). The binary constraint is generally relaxed by forming the 0-1 variables into the interval of $[0,1]$. It could be achieved by applying the *sigmoid* function. We first introduce a new variable $\mathbf{P}(x,y)$, which is with the same dimension as $\mathbf{M}(x,y)$. Then $\mathbf{M}(x,y)$ can be relaxed by applying the *sigmoid* function:

$$\mathbf{M}(x,y) = \frac{1}{1 + \exp[-\beta_M(\mathbf{P}(x,y))]}, \tag{8}$$

where β_M is the steepness factor.

Similarly, we consider using the *sigmoid* function to smooth the step function in Equation (4):

$$\mathbf{Z}(x,y) = \frac{1}{1 + \exp[-\beta_Z(\mathbf{I}(x,y)) - I_{th}]}, \tag{9}$$

where β_Z is the steepness factor. After relaxation, the problem formulation in Equations (3) to (6) finally becomes:

$$\begin{aligned} \min_{\mathbf{P}} \quad & \|\mathbf{Z} - \mathbf{Z}_t\|_2^2 \\ \text{s.t.} \quad & \text{Equations (5), (8)and (9),} \end{aligned} \tag{10}$$

where all constraints are differentiable and can be utilized for gradient propagation.

Concurrently optimizing the discrepancies between the target layout and printed wafer masks under all process conditions is a significant challenge during the optimization process. Assuming we need to simultaneously consider N objectives, after gradient backpropagation, we obtain N gradients:

$$\mathbf{G}_n = \nabla_{\mathbf{M}} \mathcal{L}_N, \quad 1 \le n \le N, \tag{11}$$

which need to be aggregated to a single overall gradient $\mathbf{G}$ for updating the mask. Typically, such aggregation can be done with a

weighted accumulation of all gradients, i.e.,

$$\mathbf{G} = \sum_{n}^{N} \lambda_n \mathbf{G}_n, \tag{12}$$

where λ_n is the weight factor, which can be customized based on heuristics, manually fixed, or dynamically adjusted during the updating process. However, this approach may produce sub-optimal results due to *non-uniform gradient*. Through an analysis of the phenomena depicted in Figure 3a, we claim that optimization challenges come from non-uniform gradients:

(1) Gradients from different objectives may diverge in terms of the direction, i.e., the angle between many gradient pairs is very large, resulting in conflicting gradients that can negatively impact the performance of specific tasks when directly optimizing the average loss [26].
(2) Gradients from different objectives may differ in scale, where the largest gradient may dominate the update process [27], potentially resulting in incomplete optimization of specific goals.

To address these issues, we will introduce a uniform gradient computation approach.

3.2 Uniform Gradient Computation in RMO-ILT

To obtain a more uniform gradient for mask updates in ILT, we consider the gradient calculation in two aspects, namely gradient direction and gradient magnitude, which also correspond to the aforementioned two challenges. Figure 3 simply illustrates the visual difference between our method and standard gradient accumulation.

3.2.1 Uniform Gradient Direction. To address the first challenge, we need to explicitly handle the diverged direction among different gradients. First, we define a pair of gradient vectors $\mathbf{G}_i$ and $\mathbf{G}_j$ as a *conflicting gradient pair*, where $\mathbf{G}_i$ and $\mathbf{G}_j$ represent gradients under two different process parameters, using Equation (11). This definition holds if the angle θ_{ij} between them satisfies $\cos\theta_{ij} < 0$, i.e., $\theta_{ij} > 90°$. We resolve this conflict by projecting one gradient $\mathbf{G}_i$ onto the orthogonal direction of the other gradient $\mathbf{G}_j$, aiming to eliminate the conflict while preserving the updated step size and direction along the original gradient direction as much as possible.

For simplification, we will focus on $\mathbf{G}_i$ as the reference gradient. Specifically, we start by obtaining the projection lengths $\mathbf{G}_{norm}$ of $\mathbf{G}_i$ and $\mathbf{G}_j$ onto each other's gradient direction:

$$\mathbf{G}_{ij}^{norm} = \frac{\mathbf{G}_i \cdot \mathbf{G}_j}{\|\mathbf{G}_j\|}, \tag{13}$$

where $\|\cdot\|$ denotes the l_2 norm, and $\mathbf{G}_i \cdot \mathbf{G}_j$ is the inner product. Then, we compute the difference between each gradient and its corresponding projected gradient $\mathbf{G}_{proj} = \mathbf{G}_{ij}^{norm} \frac{\mathbf{G}_j}{\|\mathbf{G}_j\|}$ to obtain the projection onto the normal vector of the other gradient, respectively. So, we can get final projected gradients $\mathbf{G}_i'$:

$$\mathbf{G}_i' = \mathbf{G}_i - \frac{\mathbf{G}_i \cdot \mathbf{G}_j}{\|\mathbf{G}_j\|^2} \mathbf{G}_j. \tag{14}$$

By iteratively traversing all gradient pairs and gradually resolving conflicts, we ensure that the angles between updated gradients are within the range of $[0, 90]$ degrees, which promotes a set of more uniform gradient components.

It should be noticed that since we mitigate the direction conflicts between pairwise gradients by traversing and then gradually updating the gradient directions, the traversing order may introduce extra bias. Therefore, we use random perturbations each time before traversal to reduce bias. After mitigating the conflicts between gradient pairs, we sum up the obtained N projected gradients:

$$\mathbf{G}'' = \sum_{n=1}^{N} \mathbf{G}_n'. \tag{15}$$

Finally, we normalize it to obtain the final gradient direction by

$$\overline{\mathbf{G}} = \frac{\mathbf{G}''}{\|\mathbf{G}''\|}, \tag{16}$$

which is essentially a unit vector in the direction of $\mathbf{G}''$.

3.2.2 Gradients Magnitude-balancing. After obtaining conflict-free gradient directions, the approach described above neglects the gradient magnitudes issues. To address the gradient scale issue, we first consider the general formulation for gradient descent:

$$\mathbf{M}_t = \mathbf{M}_{t-1} - \eta_t \mathbf{G}_{\mathbf{M}_t}^{norm} \nabla_{\mathbf{M}_t} \tag{17}$$

where $\nabla_{\mathbf{M}_t}$ is a unit vector, representing the gradient direction. $\mathbf{G}_{\mathbf{M}_t}^{norm}$ and η_t stand for the magnitude and step size during the current optimization iteration t, respectively. Determining the appropriate gradient magnitude, or step size, affects the algorithm's convergence speed and final performance. Larger gradient magnitudes can accelerate convergence, while conversely, reducing magnitudes in later iterations can fine-tune the results to find a satisfactory point [28, 29]. We take the maximum magnitude of different gradients as the current step size. Therefore, combined with Equation (16), the balanced gradient for the t-th iteration can be written as

$$\mathbf{G}_{\mathbf{M}_t} = \mathbf{G}_{\mathbf{M}_t}^{norm} \nabla_{\mathbf{M}_t} = \max_{n=1,2,\ldots,N} ||\mathbf{G}_{n,t}|| \overline{\mathbf{G}}_t. \tag{18}$$

The basic idea stems from the observation that in the early stages of optimization, as shown in Figure 3, there are significant discrepancies between the printed images and the target layout, and different gradients $\mathbf{G}_n$ have large magnitudes. To accelerate convergence, a sufficiently large step size is needed. In the later stages, large magnitudes still exist, which indicates the need for continued optimization of these objectives to meet different process condition robustness requirements. When all gradient magnitudes diminish, it suggests that the current mask M is nearing a stationary point, reducing the step size is advisable to maintain performance.

3.3 Theoretical Analysis

In this section, we theoretically analyze the convergence of our update rule. Suppose there are two objectives for parameter $\mathbf{M}$, and their corresponding losses are $\mathcal{L}_1$ and $\mathcal{L}_2$, and we denote their gradient cosine similarity as $\cos\theta_{12}$. When $\theta_{12} \geq 0$, our method is equivalent to standard gradient descent [30]. So we only consider the other case when $\cos\theta_{12} < 0$.

Theorem 1. *Suppose $\mathcal{L}_1$ and $\mathcal{L}_2$ are both convex and differentiable, and the gradient of $\mathcal{L}$ is Lipschitz continuous with a constant $L > 0$. Then, by employing the uniform update rule with a step size $\eta_t < \frac{1}{L}$, convergence to the optimal value $\mathcal{L}(\theta^*)$ is expected, or the process may settle near a stationary point when $\cos\theta_{12} = -1$.*

Proof. Before our proof, we introduce some notations for ease of understanding subsequent discussions. For the brevity of our

Algorithm 1 RMO-ILT

Input: numbers of iterations T, learning rate η, initial mask $\mathbf{M}_0$, target layout $\mathbf{Z}_t$, N process parameters.
Output: Optimized mask $\mathbf{M}^*$.

$\mathbf{M}^* \leftarrow \mathbf{M}_0$
for $t \leftarrow 1$ to T **do**
  $\mathbf{G}_{n,t} \leftarrow \nabla_{M_{t-1}} \mathcal{L}_n(M_{t-1}), \forall n \in \{1, 2, ..., N\}$
  $\mathbf{G}'_{n,t} \leftarrow \mathbf{G}_{n,t}, \forall n \in \{1, 2, ..., N\}$
  for $n \leftarrow 0$ to N **do**
    $index\_set \leftarrow \{1, 2, ..., N\} \setminus n$
    randomly shuffle $index\_set$
    for $m \in index\_set$ **do**
      if $\cos\theta_{nm} < 0$ **then** ▷ Alleviate gradient conflict
        $\mathbf{G}'_{n,t} \leftarrow \mathbf{G}'_{n,t} - \frac{\mathbf{G}'_{n,t} \cdot \mathbf{G}_{j,t}}{\|\mathbf{G}_{j,t}\|^2} \mathbf{G}_{j,t}$
      end if
    end for
  end for
  $\mathbf{G}''_t = \sum_{n=1}^{N} \mathbf{G}'_{n,t}$; $\overline{\mathbf{G}}_t \leftarrow \frac{\mathbf{G}''_t}{\|\mathbf{G}''_t\|}$
  $\mathbf{G}_{\mathbf{M}_t}^{norm} \leftarrow \max_{n=1,2,...,N} \|\mathbf{G}_{n,t}\|$ ▷ Amplitude balancing
  $\mathbf{M}_t \leftarrow \mathbf{M}_{t-1} - \eta \mathbf{G}_{\mathbf{M}_t}^{norm} \overline{\mathbf{G}}_t$
  if $\mathbf{M}_t$ is better than $\mathbf{M}^*$ **then**
    $\mathbf{M}^* \leftarrow \mathbf{M}_t$
  end if
end for
return $\mathbf{M}^*$

proof process, we omit the iteration index unless specified otherwise. We let $\|\cdot\|$ denote the L_2-norm and $\Delta\mathcal{L}$ represent the gradient of parameter $\mathbf{M}$. Therefore, the gradients for objectives $\mathcal{L}_1$ and $\mathcal{L}_2$ are $\mathbf{G}_1 = \Delta\mathcal{L}_1$, $\mathbf{G}_2 = \Delta\mathcal{L}_2$, and $\mathbf{G} = \mathbf{G}_1 + \mathbf{G}_2 = \Delta\mathcal{L}$ is the general gradient accumulation. Thus, according to our update rule, with Equation (17) and Equation (18), we know:

$$\begin{aligned}\mathbf{M}_t &= \mathbf{M}_{t-1} - \eta_t \mathbf{G}_{\mathbf{M}_t}^{norm} \nabla_{\mathbf{M}_t} = \mathbf{M}_{t-1} - \eta_t (\max_{n=1,2} \|\mathbf{G}_n\|) \overline{\mathbf{G}} \\ &= \mathbf{M}_{t-1} - \alpha\eta_t \sum_{n}^{2} \mathbf{G}'_n \end{aligned} \tag{19}$$

where we let $\alpha = \left(\max_{n=1,2} \|\mathbf{G}_n\|\right) / \left\|\sum_n^2 \mathbf{G}'_n\right\|$. Then, we can perform a quadratic Taylor expansion around $\mathcal{L}(\mathbf{M}_t)$:

$$\begin{aligned}\mathcal{L}_(\mathbf{M}_t) \leq& \mathcal{L}(\mathbf{M}_{t-1}) + \nabla\mathcal{L}(\mathbf{M}_{t-1})^T (\mathbf{M}_t - \mathbf{M}_{t-1}) \\ &+ \frac{1}{2}\nabla^2\mathcal{L}(\mathbf{M}_{t-1}) \|\mathbf{M}_t - \mathbf{M}_{t-1}\|^2 \\ \leq& \mathcal{L}(\mathbf{M}_{t-1}) + \nabla\mathcal{L}(\mathbf{M}_{t-1})^T (\mathbf{M}_t - \mathbf{M}_{t-1}) + \frac{1}{2} L \|\mathbf{M}_t - \mathbf{M}_{t-1}\|^2 \end{aligned} \tag{20}$$

Thus, we plug in our uniform update rule, so we have:

$$\begin{aligned}\mathcal{L}(\mathbf{M}_t) \leq& \mathcal{L}(\mathbf{M}_{t-1}) - \alpha\eta_t g^T \left(-\mathbf{G} + \frac{\mathbf{G}_1 \cdot \mathbf{G}_2}{\|\mathbf{G}_1\|^2}\mathbf{G}_1 + \frac{\mathbf{G}_1 \cdot \mathbf{G}_2}{\|\mathbf{G}_2\|^2}\mathbf{G}_2\right) \\ &+ \frac{1}{2}\alpha^2 L\eta_t^2 \left\|-\mathbf{G} + \frac{\mathbf{G}_1 \cdot \mathbf{G}_2}{\|\mathbf{G}_1\|^2}\mathbf{G}_1 + \frac{\mathbf{G}_1 \cdot \mathbf{G}_2}{\|\mathbf{G}_2\|^2}\mathbf{G}_2\right\|^2 \\ =& \mathcal{L}(\mathbf{M}_{t-1}) - \left(\alpha\eta_t - \frac{1}{2}\alpha^2 L\eta_t^2\right)\left(1 - cos^2\theta_{12}\right)\left(\|\mathbf{G}_1\|^2 + \|\mathbf{G}_2\|^2\right) \\ &- \alpha^2 L\eta_t^2 \left(1 - cos^2\theta_{12}\right) \|\mathbf{G}_1\| \|\mathbf{G}_2\| cos\theta_{12} \end{aligned} \tag{21}$$

Since we know that $\alpha\eta_t \left(1 - cos^2\theta_{12}\right)\left(\|\mathbf{G}_1\|^2 + \|\mathbf{G}_2\|^2\right) \geq 0$, so we have:

$$\begin{aligned}\mathcal{L}(\mathbf{M}_t) \leq& \mathcal{L}(\mathbf{M}_{t-1}) - \frac{\alpha\eta_t}{2}\left(1 - cos^2\theta_{12}\right)\left(\|\mathbf{G}_1\|^2 + \|\mathbf{G}_2\|^2\right) \\ &+ \frac{\alpha^2}{2} L\eta_t^2 \left(1 - cos^2\theta_{12}\right)\left(\|\mathbf{G}_1\|^2 + \|\mathbf{G}_2\|^2\right) \\ &- \alpha^2 L\eta_t^2 \left(1 - cos^2\theta_{12}\right) \|\mathbf{G}_1\| \|\mathbf{G}_2\| cos\theta_{12} \end{aligned} \tag{22}$$

Using $\eta_t \leq \frac{1}{L}$, we can have $L\eta_t^2 \leq \eta_t$, and then plug this inequality into the above expression, we have:

$$\begin{aligned}\mathcal{L}(\mathbf{M}_t) \leq& \mathcal{L}(\mathbf{M}_{t-1}) - \frac{\eta_t}{2}(1 - cos^2\theta_{12})\Big[(\alpha + \alpha^2)(\|\mathbf{G}_1\|^2 + \|\mathbf{G}_2\|^2) \\ &+ 2\alpha^2 \|\mathbf{G}_1\| \|\mathbf{G}_2\| cos\theta_{12}\Big] \\ =& \mathcal{L}(\mathbf{M}_{t-1}) - \frac{\eta_t}{2}\left(1 - cos^2\theta_{12}\right)\Big[\alpha\left(\|\mathbf{G}_1\|^2 + \|\mathbf{G}_2\|^2\right) \\ &+ \alpha^2\left(\|\mathbf{G}_1\|^2 + \|\mathbf{G}_2\|^2 + 2\|\mathbf{G}_1\| \|\mathbf{G}_2\| cos\theta_{12}\right)\Big] \\ =& \mathcal{L}(\mathbf{M}_{t-1}) - \frac{\eta_t}{2}(1 - cos^2\theta_{12})\Big[\alpha\left(\|\mathbf{G}_1\|^2 + \|\mathbf{G}_2\|^2\right) + \alpha^2 \|\mathbf{G}\|^2\Big] \\ \leq& \mathcal{L}(\mathbf{M}_{t-1}) - \frac{\eta_t}{2}(1 - cos^2\theta_{12})(\alpha + \alpha^2) \|\mathbf{G}\|^2 \end{aligned} \tag{23}$$

In discussing α, it is evident that $\alpha > 0$ always holds. It should be noticed that $\left\|\sum_n^2 \mathbf{G}'_n\right\|$ may be equal to 0 (the cosine similarity $\cos\theta_{12} = -1$ is uncommon in practice). For numerical stability, in practical applications, we set $\alpha = \left(\max_{n=1,2} \|\mathbf{G}_n\|\right) / \left(\left\|\sum_n^2 \mathbf{G}'_n\right\| + \epsilon\right)$, where ϵ is a very small constant. Next, we give more detailed exploration when $\alpha > 0$. We let $\beta = \left\|\sum_n^2 \mathbf{G}'_n\right\|$. So $\alpha = \left(\max_{n=1,2} \|\mathbf{G}_n\|\right) / \beta$. Then we can expand it and obtain:

$$\begin{aligned}\beta =& \left\|\mathbf{G}_1 + \mathbf{G}_2 - \frac{\mathbf{G}_1 \cdot \mathbf{G}_2}{\|\mathbf{G}_1\|^2}\mathbf{G}_1 - \frac{\mathbf{G}_1 \cdot \mathbf{G}_2}{\|\mathbf{G}_2\|^2}\mathbf{G}_2\right\| \\ \leq& \|\mathbf{G}_1 + \mathbf{G}_2\| + \left\|\frac{\mathbf{G}_1 \cdot \mathbf{G}_2}{\|\mathbf{G}_1\|^2}\mathbf{G}_1 + \frac{\mathbf{G}_1 \cdot \mathbf{G}_2}{\|\mathbf{G}_2\|^2}\mathbf{G}_2\right\| \\ \leq& \|\mathbf{G}_1\| + \|\mathbf{G}_2\| + \frac{|\|\mathbf{G}_1\| \|\mathbf{G}_2\| cos\theta_{12}|}{\|\mathbf{G}_1\|^2} \|\mathbf{G}_1\| + \frac{|\|\mathbf{G}_1\| \|\mathbf{G}_2\| cos\theta_{12}|}{\|\mathbf{G}_2\|^2} \|\mathbf{G}_2\| \\ \leq& 4 \max_{n=1,2} \|\mathbf{G}_n\| \end{aligned} \tag{24}$$

so, we have $\alpha \geq \frac{1}{4}$ when $\cos\theta_{12} \neq -1$, and we can infer that $(\alpha + \alpha^2) \geq \frac{5}{16}$. Finally, we plug this conclusion into Equation (23):

$$\mathcal{L}(\mathbf{M}_t) \leq \mathcal{L}(\mathbf{M}_{t-1}) - \frac{5\eta_t}{32}\left(1 - cos^2\theta_{12}\right) \|\mathbf{G}\|^2 \tag{25}$$

It suggests that if $cos\theta_{12} > -1$, then $\mathcal{L}(\mathbf{M}_t) < \mathcal{L}(\mathbf{M}_{t-1})$ unless $\|\mathbf{G}\| = 0$. This indicates that utilizing our uniform update rule can consistently reduce the loss value. Through multiple iterations, it can converge to the optimal value $\mathcal{L}(\mathbf{M}) = \mathcal{L}(\mathbf{M}^*)$ or when the step size η_t is sufficiently small, it will converge to the point where $\theta_{12} = 180^\circ$. □

Corollary 1. *Suppose N objectives $\mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_N$ are both convex and differentiable, and the gradient of $\mathcal{L}$ is Lipschitz continuous with a constant $L > 0$. Assume that angle between the gradient $\sum_n^N \mathbf{G}$ and $\sum_n^N \mathbf{G}'$ is smaller than 60°. Then, by employing the balanced update rule with a step size $\eta_t \leq \frac{2N\left\|\sum_n^N \mathbf{G}_n\right\|}{L\|\mathbf{G}''\|}$, convergence to the optimal value*

$\mathcal{L}(\mathbf{M}^*)$ *is expected, or the process may settle near a stationary point when* $\cos\theta_{ij} = -1$ *for all gradient* $\mathbf{G}_i$ *and* $\mathbf{G}_j$ *pairs.*

Proof. We first let $\gamma = \left(\max_{n=1,2,\ldots,N} \|\mathbf{G}_n\|\right) / \|\mathbf{G}''\|$. Then according to Equation (19), we have: $\mathbf{M}_t = \mathbf{M}_{t-1} - \gamma\eta_t\mathbf{G}''$. Let plug it in Equation (21), we have:

$$\begin{aligned}\mathcal{L}_(\mathbf{M}_t) &\leq \mathcal{L}(\mathbf{M}_{t-1}) - \frac{1}{2}\eta_t\gamma\left\|\sum_n^N \mathbf{G}_n\right\|\left\|\mathbf{G}''\right\| + \frac{1}{2}L\eta_t^2\gamma^2\left\|\mathbf{G}''\right\|^2 \quad (26)\\ &\leq \mathcal{L}(\mathbf{M}_{t-1})\end{aligned}$$

Therefore, by iteratively applying the update rule, it can reach the optimal value $\mathcal{L}(\mathbf{M}^*)$ since the objective function strictly decreases. More detailed proof is omitted due to the page limit. □

4 Efficient implementation

4.1 Acceleration using Multiple GPUs

While ILT algorithms enable finer mask optimization, the expanded solution space, compared to rule-based and model-based methods [2–4], brings both precision benefits and increased time consumption, limiting the widespread application of ILT. To address this efficiency challenge, recent studies have adopted GPU acceleration for ILT [12, 31].

Designing ILT algorithms for process variation robustness introduces additional computational costs during optimization. Motivated by the parallel training approach on deep neural networks, we concatenate optical kernels with different simulation parameters and deploy them across multiple GPUs to achieve parallel lithography forward simulation and backward propagation under various process conditions. Through this acceleration approach, our framework demonstrates good scalability, allowing mask optimization across different numbers of lithography models.

4.2 Acceleration through Objective Sampling

With the utilization of multiple GPUs, the overall runtime of our algorithm is significantly reduced. However, due to our uniform update rule adhering to a temporal design approach (see Line 5 to Line 13 in Algorithm 1), the time complexity is $O(nm)$, where m denotes the number of conflicts generated. As the number of corners considered during the optimization process increases, this becomes the primary bottleneck. Therefore, to further accelerate the optimization process, we introduce the objective sampling strategy: achieving acceleration by filtering out some objectives, which may greatly enhance runtime efficiency. In practice, we randomly sample a subset of process corners during each optimization iteration. We will discuss the runtime and associated performance under this strategy in Section 5.4.

5 Experimental Results

5.1 Implementation Details

We implement our RMO-ILT framework on the Pytorch platform. Both mask optimization and evaluation are conducted under a Linux system equipped with a 2.8GHz AMD EPYC 7543 32-core processor and 8 Nvidia 3090 GPUs. We validate the performance on the ICCAD2013 CAD Contest [20], provided with 10 M1 design cases on 32nm technology node, along with a lithography model. All design masks are resized to a high-resolution scale, i.e., 2048 × 2048, to rigorously validate the effectiveness of our approach. We adhere to the parameter configurations in [31] for lithography simulation. The exposure dose range is restricted to within ±2%, while specific values are chosen from the set $\{0.98, 0.99, 1.00, 1.01, 1.02\}$, where the nominal dose value is 1.00. Defocus error settings are based on [5]. The optimization process consists of $T = 20$ iterations, adopted with SGDM optimizer [32], where we set the fixed stepsize η and momentum parameters to 3.0 and 0.0001, respectively.

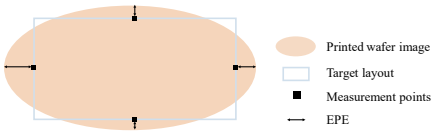

Figure 4: EPE measurement.

To assess the process window, we need to evaluate the wafer pattern on each process corner. In this work, edge placement error (EPE) is utilized, which is a primary metric in mask optimization. It is obtained by examining the deviation of sampling points along the contour of the target layout in both horizontal and vertical directions, as illustrated in Figure 4.

5.2 Comparison with Existing ILT Methods

To validate the effectiveness of RMO-ILT, we conduct detailed comparisons with state-of-the-art (SOTA) methods. These methods include conventional process variation-aware ILT [5] and deep learning-based OPC approaches, namely GANOPC [6], CFNOILT [33], and NeuraILT [12]. The baseline methods are provided by open-source frameworks [31, 34]. For a fair comparison, the proposed approach is also implemented on [31]. All parameters are maintained consistent with their original settings. Particularly for deep learning methods, we directly utilize pre-trained models for inference.

The comparison results are summarized in Table 1. We rigorously evaluate each design and obtain four relevant metrics based on the EPE distribution across six distinct process conditions. These process conditions are formed by dose variations of $\{-2\%, 0, +2\%\}$, and focus/defocus settings. The “Nominal" column represents the optimized EPE results under the nominal process condition, serving as the standard evaluation metric. The “Worst" column depicts the maximum EPE distribution under all process corners, focusing on the maximum disparity between the wafer image and target layout in worst-case scenarios. The “Mean" and “Std" columns statistically analyze the mean and standard deviation of the EPE distribution, verifying their robustness under various process conditions.

The results from Table 1 demonstrate that our method consistently maintains the lowest mean EPE across nearly all conditions, with an average of 6.89 across the 10 design cases, significantly outperforming other methods. This indicates that after mask optimization, the wafer images obtained through simulation under six different process conditions closely align with the target layout. Moreover, RMO-ILT achieves the lowest standard deviation for the majority of test cases,which suggests that our masks exhibit higher robustness against various process variations through multi-objective optimization. Furthermore, considering all worst-case EPE distributions, RMO-ILT effectively mitigates the impact of extreme process errors on the quality of the final wafer image, with values approximately 2X and 3.6X lower than those of other methods such as GANOPC and CFNOILT, respectively. Lastly, as shown in Table 1, even without any specific additional optimization for the nominal

Table 1: Performance comparisons with SOTA methods.

Design	MOSAIC [5]				NeuralILT [12]				GANOPC [6]				CFNOILT [33]				Ours			
	Nominal	Worst	Mean	Std	Nominal	Worst	Mean	Std	Nominal	Worst	Mean	Std	Nominal	Worst	Mean	Std	Nominal	Worst	Mean	Std
Case 1	8	23	15	5.02	8	19	13.83	3.92	17	41	26.33	9.71	17	39	24.83	8.26	**6**	**13**	**8.67**	**3.33**
Case 2	**0**	11	5.5	3.62	8	25	15.67	6.89	4	24	14.33	8.64	17	49	27.83	13.12	**0**	**6**	**2.5**	**2.26**
Case 3	49	**60**	49.33	**6.25**	**42**	62	50.5	8.04	47	74	56.17	10.65	67	98	76.67	12.27	47	61	**49**	7.46
Case 4	2	5	3	1.26	**1**	5	2.5	1.64	2	10	4.67	2.94	9	27	13.67	7.94	**1**	**2**	**1.83**	**0.41**
Case 5	**0**	7	1.5	2.74	2	5	3.67	1.21	2	7	3.17	2.14	4	47	15.67	17	**0**	**3**	**0.67**	**1.21**
Case 6	3	4	**1.67**	1.86	4	5	3.5	1.22	3	17	8.67	5.24	3	41	17.5	14.39	**2**	**3**	**1.67**	**0.82**
Case 7	2	19	5.83	7.86	**0**	**6**	**1.67**	**2.34**	**0**	18	4.17	6.88	**0**	21	5	8	**0**	16	3.17	6.4
Case 8	**0**	**0**	**0**	**0**	**0**	4	1.17	1.47	2	11	4.33	4.23	1	13	5.83	4.58	**0**	1	0.17	0.41
Case 9	1	6	2.5	1.87	4	18	9.17	5.81	6	25	15	7.56	7	24	13.17	5.78	**0**	**3**	**1.17**	**1.17**
Case 10	**0**	**0**	**0**	**0**	4	5	2.83	2.04	5	14	7	4.86	**0**	3	0.83	1.33	**0**	**0**	**0**	**0**
Average	6.5	13.5	8.43	-	7.3	15.4	10.45	-	8.8	24.1	14.38	-	12.5	36.2	20.1	-	**5.6**	**10.8**	**6.89**	-

Table 2: Comparison of overall runtime.

Design	MOSAIC [5]	NeuralILT [12]	GANOPC [6]	CFNOILT [33]	Ours	
					Ori.	Accel.
Case1	7.50s	4.92s	6.19s	5.46s	41.40s	17.72s
Case2	7.55s	5.25s	5.75s	5.11s	41.70s	15.16s
Case3	7.45s	4.95s	6.05s	5.13s	41.03s	15.13s
Case4	7.48s	5.12s	5.82s	5.26s	41.43s	16.28s
Case5	7.66s	4.94s	5.88s	5.32s	41.42s	16.10s
Case6	7.60s	4.90s	5.89s	5.51s	41.47s	14.96s
Case7	7.52s	4.92s	5.72s	5.05s	42.04s	15.87s
Case8	7.56s	5.08s	5.77s	5.12s	41.19s	16.05s
Case9	7.78s	5.17s	5.84s	5.00s	41.84s	14.97s
Case10	8.03s	4.87s	5.82s	4.95s	42.44s	16.42s
Average	7.61s	5.01s	5.87s	5.19s	41.60s	15.86s

process parameter, our method still achieves the best nominal EPE results across nearly all test cases, with an average of 5.6, far exceeding other state-of-the-art methods.

In Table 2, we list the overall runtime for the entire end-to-end mask optimization process on a given input mask. Compared to other methods, RMO-ILT requires more time to obtain the optimized mask. **It is mainly because RMO-ILT needs to rigorously compute the wafer pattern Z and gradient G on all the process conditions, which the baseline methods failed to consider.** In the subsequent Section 5.4, we will discuss how to leverage multiple GPUs for parallel acceleration and further accelerate the process using sampling strategies as outlined in Section 4.2.

Figure 5 shows our optimized masks for 6 test cases, along with their corresponding printed nominal wafer images and PV bands. All printed images exhibit good performance, demonstrating the adaptability of our algorithm across diverse test cases.

5.3 Comparison with Gradient Accumulation

In this section, we demonstrate the superiority of the proposed algorithm to the conventional gradient accumulation approach for the multi-objective optimization problem. We introduce a larger number of process conditions to reflect the capability of our algorithm. More specifically, we enlarge the dose variation to cover more configurations, i.e., $\{-2\%, -1\%, 0, +1\%, +2\%\}$. Considering more process variations during the optimization process implies a higher likelihood of encountering conflicts between different process corners. To validate the effectiveness of our conflict resolution approach, we compare it with the standard gradient accumulation method, similar to MOSAIC [5], where the gradient for mask update is computed by accumulating the gradient of each objective component. As mentioned earlier, this approach is prone to gradient direction conflicts and domination of gradient magnitudes over others, leading to biased updates. As depicted in Table 3, we present the EPE

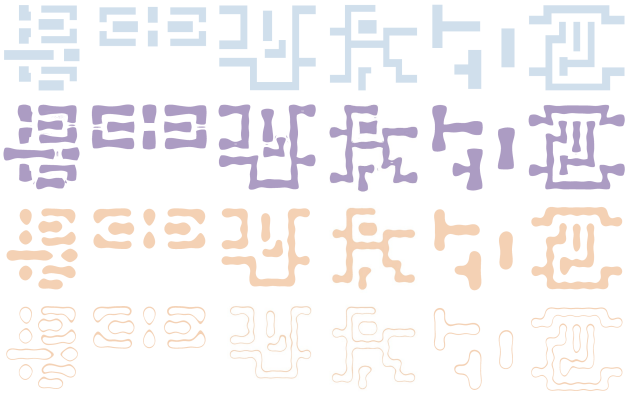

Figure 5: Visualizations of results. Rows from top to bottom are: target layouts, optimized masks, printed wafer images, and PV bands.

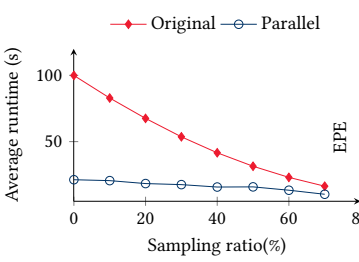

Figure 6: Runtime with objective sampling.

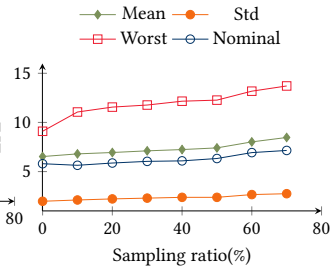

Figure 7: Mask quality with objective sampling.

distribution of some test cases to show the real process window. For each case, the top row represents the results obtained using the gradient accumulation method, while the bottom row corresponds to our results, with each column comparing the performance of the same test case. It is evident that the process window achieved with our approach is much better across all these test cases compared to the gradient accumulation-based approach. Furthermore, in nearly all cases, our method yields significantly better mean EPE and EPE deviation compared to the gradient accumulation method, indicating the effectiveness and significance of our uniform update rule.

Table 3: Process window comparison between gradient accumulation on ten process corners. In each group, the top and bottom tables correspond to gradient accumulation and RMO-ILT, respectively.

(a) Case 2 (#Measurement points: 116)

Focus \ Dose	0.98	0.99	1.0	1.01	1.02
Nominal	4	3	**0**	**0**	**6**
Defocus	10	8	3	8	13
Mean	5.50				
Std.	4.28				

Focus \ Dose	0.98	0.99	1.0	1.01	1.02
Nominal	**1**	**0**	**0**	2	8
Defocus	**5**	**2**	**0**	**2**	**5**
Mean	**2.50**				
Std.	**2.68**				

(b) Case 4 (#Measurement points: 64)

Focus \ Dose	0.98	0.99	1.0	1.01	1.02
Nominal	6	6	6	3	**2**
Defocus	6	6	6	6	4
Mean	5.10				
Std.	1.52				

Focus \ Dose	0.98	0.99	1.0	1.01	1.02
Nominal	**2**	**1**	**1**	**1**	4
Defocus	**2**	**2**	**1**	**1**	**2**
Mean	**1.70**				
Std.	**0.95**				

(c) Case 5 (#Measurement points: 169)

Focus \ Dose	0.98	0.99	1.0	1.01	1.02
Nominal	3	2	2	3	5
Defocus	4	3	3	3	3
Mean	3.10				
Std.	**0.88**				

Focus \ Dose	0.98	0.99	1.0	1.01	1.02
Nominal	**0**	**0**	**1**	**1**	**3**
Defocus	**1**	**0**	**0**	**0**	**1**
Mean	**0.70**				
Std.	0.95				

(d) Case 6 (#Measurement points: 161)

Focus \ Dose	0.98	0.99	1.0	1.01	1.02
Nominal	**1**	4	4	4	5
Defocus	**1**	**1**	**1**	**1**	**2**
Mean	2.40				
Std.	1.65				

Focus \ Dose	0.98	0.99	1.0	1.01	1.02
Nominal	2	**1**	**1**	**2**	**2**
Defocus	2	**1**	**1**	**1**	3
Mean	**1.60**				
Std.	**0.70**				

(e) Case 7 (#Measurement points: 134)

Focus \ Dose	0.98	0.99	1.0	1.01	1.02
Nominal	3	2	2	**0**	2
Defocus	16	12	5	2	**0**
Mean	4.40				
Std.	5.34				

Focus \ Dose	0.98	0.99	1.0	1.01	1.02
Nominal	**0**	**0**	**0**	**0**	**0**
Defocus	**14**	**8**	**2**	**0**	**0**
Mean	**2.40**				
Std.	**4.79**				

(f) Case 9 (#Measurement points: 189)

Focus \ Dose	0.98	0.99	1.0	1.01	1.02
Nominal	**0**	**0**	1	3	6
Defocus	**1**	**0**	**0**	**0**	4
Mean	1.50				
Std.	2.12				

Focus \ Dose	0.98	0.99	1.0	1.01	1.02
Nominal	**0**	**0**	**0**	**1**	**3**
Defocus	2	**0**	**0**	1	**1**
Mean	**0.80**				
Std.	**1.03**				

5.4 Scalability Analysis

As previously mentioned, the rigorous approach introduces extra objectives and inevitably leads to an increase in runtime. In this section, we demonstrate the scalability of our algorithm and the efficiency of our accelerated implementation in Section 4 using multiple GPUs and the objective sampling strategy.

Firstly, as illustrated in Table 2, acceleration with multiple GPUs for batch execution results in approximately 2.6 times faster overall runtime under six process conditions compared to serial processing on a single GPU. However, despite this significant acceleration, without factoring in the runtime of other modules, our algorithm still presents a bottleneck in terms of time cost, with time complexity of $O(nm)$, where the computational time cost approximately quadratically increases when considering more process conditions.

Therefore, we further design an objective sampling strategy to speed up the optimization runtime across many process corners. As the sampling ratio increases, fewer process parameters need to be considered. As shown in Figure 6, we incorporate this strategy before and after parallel processing to explore its impact on the overall runtime, with employing different ratios. The maximum ratio is set at 70% to ensure that the total number of our algorithm's objectives still meets the requirements of the multi-objective optimization problem and preserves the presence of process variation. Comparing before and after implementing the sampling strategy across ten process corners, as depicted in Figure 6, the results demonstrate a clear linear decrease in runtime, especially pronounced in the absence of parallel acceleration. Despite the substantial reduction in time consumption achieved by such parallel acceleration, the incorporation of the strategy still yields significant acceleration benefits. For instance, when the masked sampling ratio is 70%, approximately a 2× acceleration effect is achieved.

Furthermore, to validate the trade-off between performance and runtime, we examined the impact of the sampling strategy on mask quality. Shown in Figure 7, the EPE deviation remains relatively constant as the sampling ratio increases, with little change even when 70% of process conditions are excluded. This highlights our algorithm's robustness. While worst-case EPE increases slightly with higher sampling ratios, the average and nominal EPEs exhibit only slight fluctuations, indicating that RMO-ILT maintains strong performance across iterations despite some compromise in conflict alleviation.

6 Conclusion

In this paper, we propose RMO-ILT, an inverse lithography algorithm employing rigorous multi-objective optimization to enhance robustness against process variations. By alleviating conflicts and balancing amplitudes among multiple objectives during optimization, our method effectively expands the process window. Moreover, with two acceleration implementations, experimental results showcase the superior efficiency and effectiveness of our approach.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 62204066) and Guangzhou Municipal Science and Technology Project (Municipal Key Laboratory Construction Project, Grant No.2023A03J0013).

References

[1] D. Z. Pan, B. Yu, and J.-R. Gao, "Design for manufacturing with emerging nanolithography," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2013.

[2] O. W. Otto, J. G. Garofalo, K. Low, C.-M. Yuan, R. C. Henderson, C. Pierrat, R. L. Kostelak, S. Vaidya, and P. Vasudev, "Automated optical proximity correction: a rules-based approach," in *Optical/Laser Microlithography VII*, vol. 2197, 1994.

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

[4] T. Matsunawa, B. Yu, and D. Z. Pan, "Optical proximity correction with hierarchical bayes model," in *Optical Microlithography XXVIII*, vol. 9426, 2015.

[5] J.-R. Gao, X. Xu, B. Yu, and D. Z. Pan, "Mosaic: Mask optimizing solution with process window aware inverse correction," in *Proceedings of the 51st Annual Design Automation Conference*, 2014.

[6] H. Yang, S. Li, Y. Ma, B. Yu, and E. F. Young, "Gan-opc: Mask optimization with lithography-guided generative adversarial nets," in *Proceedings of the 55th Annual Design Automation Conference*, 2018.

[7] Y. Ma, W. Zhong, S. Hu, J.-R. Gao, J. Kuang, J. Miao, and B. Yu, "A unified framework for simultaneous layout decomposition and mask optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020.

[8] L. Pang, Y. Liu, and D. Abrams, "Inverse lithography technology (ilt): a natural solution for model-based sraf at 45-nm and 32-nm," in *Photomask and Next-Generation Lithography Mask Technology XIV*, vol. 6607, 2007.

[9] A. Poonawala and P. Milanfar, "Mask design for optical microlithography–an inverse imaging problem," 2007.

[10] Y. Shen, N. Wong, and E. Y. Lam, "Level-set-based inverse lithography for photomask synthesis," *Optics Express*, 2009.

[11] W. Lv, S. Liu, Q. Xia, X. Wu, Y. Shen, and E. Y. Lam, "Level-set-based inverse lithography for mask synthesis using the conjugate gradient and an optimal time step," *Journal of Vacuum Science & Technology B*, 2013.

[12] B. Jiang, L. Liu, Y. Ma, H. Zhang, B. Yu, and E. F. 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*, 2020.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 2015.

[14] M. Neisser and S. Wurm, "Itrs lithography roadmap: 2015 challenges," *Advanced Optical Technologies*, 2015.

[15] P. Yu, S. X. Shi, and D. Z. Pan, "Process variation aware opc with variational lithography modeling," in *Proceedings of the 43rd annual Design Automation Conference*, 2006.

[16] B. Zhu, S. Zheng, Z. Yu, G. Chen, Y. Ma, F. Yang, B. Yu, and M. D. Wong, "L2o-ilt: Learning to optimize inverse lithography techniques," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.

[17] S. Sun, F. Yang, B. Yu, L. Shang, and X. Zeng, "Efficient ilt via multi-level lithography simulation," in *2023 60th ACM/IEEE Design Automation Conference (DAC)*, 2023.

[18] S. K. Choy, N. Jia, C. S. Tong, M. L. Tang, and E. Y. Lam, "A robust computational algorithm for inverse photomask synthesis in optical projection lithography," *SIAM Journal on Imaging Sciences*, 2012.

[19] N. Jia, A. K. Wong, and E. Y. Lam, "Robust mask design with defocus variation using inverse synthesis," in *Lithography Asia 2008*, vol. 7140, 2008.

[20] 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.

[21] M. T. Emmerich and A. H. Deutz, "A tutorial on multiobjective optimization: fundamentals and evolutionary methods," *Natural computing*, 2018.

[22] A. K.-K. Wong, *Resolution enhancement techniques in optical lithography*. SPIE press, 2001, vol. 47.

[23] H. H. Hopkins, "The concept of partial coherence in optics," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 1951.

[24] N. B. Cobb, *Fast optical and process proximity correction algorithms for integrated circuit manufacturing*. University of California, Berkeley, 1998.

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

[26] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," *Advances in Neural Information Processing Systems*, 2020.

[27] L. Liu, Y. Li, Z. Kuang, J. Xue, Y. Chen, W. Yang, Q. Liao, and W. Zhang, "Towards impartial multi-task learning," 2021.

[28] B. Lin, W. Jiang, Y. Feiyang, Y. Zhang, P. Chen, Y.-C. Chen, S. Liu, and J. Kwok, "Dual-balancing for multi-task learning," 2023.

[29] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research*, 2011.

[30] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[31] S. Zheng, B. Yu, and M. Wong, "Openilt: An open source inverse lithography technique framework," in *2023 IEEE 15th International Conference on ASIC (ASICON)*, 2023.

[32] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, 1999.

[33] H. Yang and H. Ren, "Enabling scalable ai computational lithography with physics-inspired models," in *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, 2023.

[34] S. Zheng, H. Yang, B. Zhu, B. Yu, and M. Wong, "Lithobench: Benchmarking ai computational lithography for semiconductor manufacturing," *Advances in Neural Information Processing Systems*, 2024.