Efficient Bilevel Source Mask Optimization[*]

Guojin Chen
Chinese University of Hong Kong

Hongquan He
ShanghaiTech University

Peng Xu
Chinese University of Hong Kong

Hao Geng
ShanghaiTech University

Bei Yu
Chinese University of Hong Kong

Abstract

Resolution Enhancement Techniques (RETs) are critical to meet the demands of advanced technology nodes. Among RETs, Source Mask Optimization (SMO) is pivotal, concurrently optimizing both the source and the mask to expand the process window. Traditional SMO methods, however, are limited by sequential and alternating optimizations, leading to extended runtimes without performance guarantees. This paper introduces a unified SMO framework utilizing the accelerated Abbe forward imaging to enhance precision and efficiency. Further, we propose the innovative `BiSMO` framework, which reformulates SMO through a bilevel optimization approach, and present three gradient-based methods to tackle the challenges of bilevel SMO. Our experimental results demonstrate that `BiSMO` achieves a remarkable 40% reduction in error metrics and 8× increase in runtime efficiency, signifying a major leap forward in SMO.

1 Introduction

Lithography, vital for semiconductor manufacturing, advances integrated circuit (IC) development. The semiconductor industry's drive for miniaturization and efficiency challenges traditional lithography in creating finer patterns, emphasizing the need for resolution enhancement techniques (RETs) to meet advanced semiconductor requirements. Among various RETs such as sub-resolution assist features (SRAFs) [1], optical proximity correction (OPC) [2], and source mask optimization (SMO), SMO stands out due to its broader solution space. SMO uniquely optimizes both illumination source and mask pattern, ensuring pivotal lithographic fidelity vital for advancing next-generation IC manufacturing.

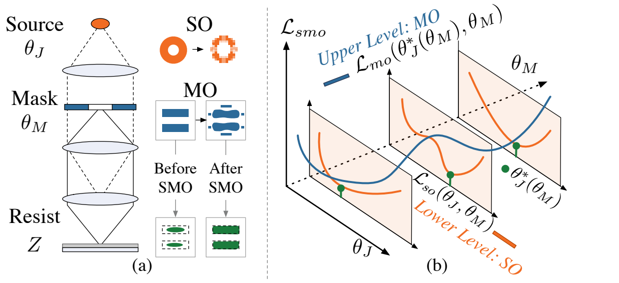


Figure 1: (a) The forward lithography and SMO process. (b) Bilevel SMO with upper-level MO and lower-level SO.

As shown in Figure 1(a), SMO integrates source optimization (SO), mask optimization (MO), and their iterative optimization refinement. The efficiency and performance of SMO are primarily influenced by two factors: the imaging model and the optimization strategy. Central to both SO and MO are the forward imaging models: Abbe's [3] and Hopkins' [4], each with unique computational characteristics. Abbe's model, celebrated for its precision, demands extensive computation through the summation of intensities from discrete source points. Conversely, Hopkins' model, employing truncated singular value decomposition (SVD), reduces computational load but is unsuitable for SO due to its loss of source information. Beyond isolated SO and MO, the strategy employed in their combined optimization significantly influences the optimization trajectory and solution space, thereby affecting SMO outcomes.

Within MO, Hopkins' model is foundational to various methods. This includes MOSAIC [2] which blends design target and process window considerations, and techniques such as GAN-OPC [5], DAMO [6], Neural-ILT [7], and DevelSet [8] that employ deep neural networks for enhanced performance. Hardware acceleration strategies like those in GPU-LS [9] and Multi-ILT [10] leverage GPU and parallel acceleration to expedite Hopkins-based MO tasks. To the best of our knowledge, the potential of Abbe-based MO optimization and acceleration remains unexplored.

For SO, the impracticality of Hopkins' model necessitates exclusive reliance on Abbe's. Previous SO strategies have employed compressive sensing [11] and sampling-based methods [12] for complexity reduction. However, the benefit is limited. By contrast, MO using the Hopkins model with GPU or DNN acceleration achieve average optimization times of five seconds, while SO dependent on Abbe's model typically exceed 30 minutes. Due to its computational intensity, accelerating Abbe's model is essential.

In SMO, alongside the imaging model, the co-optimization strategy is a critical determinant of performance. Since the inception of SMO in early 2010s, the alternating minimization (AM) strategy has been a dominant approach in SMO [11–13]. As shown in Figure 2(a), AM method involves isolated iterative minimization of source parameters across multiple SO epochs while maintaining constant mask variables, followed by mask optimization in MO epochs with the source fixed, repeating until convergence. However, AM strategy's simplicity does not guarantee effectiveness. Due to the concurrent impact of source and mask parameters on the aerial image, AM's localized focus can result in SMO being confined to local minima, and the absence of global, gradient-based guidance can prolong convergence, necessitating more iterations. These limitations highlight the need for more advanced co-optimization strategies in SMO.

To address SMO challenges, we utilize the Abbe model for forward imaging, leveraging its capability for concurrent SO and MO

[*] This work is partially supported by The Research Grants Council of Hong Kong SAR (No. CUHK14208021).

DAC '24, June 23–27, 2024, San Francisco, CA, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0601-1/24/06.
https://doi.org/10.1145/3649329.3656252

gradient computation, and its superior precision due to avoidance of approximate decomposition. We counterbalance the Abbe model's intensive computational demand with GPU acceleration. This enables us to develop an Abbe-based unified SMO framework incorporating process window considerations, consequently enhancing SMO outcomes, improving efficiency, and reducing process variability. Furthermore, we innovatively reconceptualize the SMO problem as a bilevel optimization (BLO) challenge to gain a better co-optimization strategy, as depicted in Figure 1(b). BLO, a hierarchical mathematical program, is defined as an optimization problem where the feasible region is constrained by another nested optimization problem. It is widely applied in areas such as hyperparameter optimization [14], neural architecture search [15], and multitask or meta-learning [16]. We then propose three novel gradient-based bilevel SMO solutions, featuring global perspectives achieved through MO and SO gradient fusion. These solutions demonstrate substantial improvements in efficiency and accuracy over the traditional SMO approaches. Our primary contributions are as follows:

- We establish the first unified Abbe-based SMO framework incorporating process window considerations, significantly accelerating Abbe imaging through parallel computation, achieving speeds comparable to Hopkins' method.
- We pioneer the modeling of SMO as a unified bilevel framework, developing three efficient gradient-based methods with global perspectives and improved exploration of solution space, superseding the conventional SMO.
- Our experimental results indicate that, compared to state-of-the-art (SOTA) SMO methods [12], our approach reduces error metrics by approximately 40% and achieves an eightfold increase in throughput. In comparison to SOTA MO methods [10], our error metrics are half as large.

2 Preliminaries

2.1 Lithography Simulation

In optical lithography systems, the intensity of aerial image $I(x,y)$ on the wafer plane can be formulated via lithography theory [4] as:

$$I(x,y) = \iiiint\!\!\!\iint_{-\infty}^{\infty} J(f,g)O(f',g')O^*(f'',g'')H(f+f',g+g')H^*(f+f'',g+g'') \exp(-i2\pi((f'-f'')x+(g'-g'')y))\mathrm{d}f\,\mathrm{d}g\,\mathrm{d}f'\,\mathrm{d}g'\,\mathrm{d}f''\,\mathrm{d}g'', \tag{1}$$

where $\mathbf{J}$ is the illumination source. $\mathbf{H}$ is projection system's optical transfer function. $O(f',g')$ captures the binary mask pattern $M(x,y)$'s frequency spectrum, derived via a $2D$ fast Fourier transform (FFT), $\mathcal{F}(\cdot)$. * signifies the Hermitian transpose. (x,y) denotes spatial coordinates, while (f,g), (f',g'), and (f'',g'') refer to the frequency coordinates of the source, mask spectrum, and its conjugate, respectively. The formation of aerial image in Equation (1) is computed using two distinct methods: Abbe's and Hopkins' method.

Abbe's Approach:. Abbe's approach, also known as the source points integration approach, discretizes the source space and independently computes the contribution of each source point, subsequently summing these contributions to form the aerial image. Regardless of the discretization technique, the source can hence be represented as a set of source points $\{(f_\sigma, g_\sigma; j_\sigma)\}$, where each source point defines a pair of spatial frequencies and its discrete magnitude $j_\sigma \in [0,1]$. By setting $\mathbf{A}_{(f,g)}(f',g') = H(f+f',g+g')O(f',g')$, and applying the Inverse Fast Fourier Transform (IFFT), the total intensity in Equation (1) can be formulated in Abbe's approach:

$$I(x,y) = \sum_{\sigma} j_\sigma |\mathbf{A}_{(f_\sigma,g_\sigma)}(x,y)|^2. \tag{2}$$

Hopkins' Approach:. Hopkins' approach separates the calculation of source and projection system from the processing of the mask for Equation (1). It formulates the source and projector into the *transmission cross-coefficients* (*TCC*), as defined by:

$$TCC = \iint_{-\infty}^{\infty} J(f,g)H(f+f',g+g')H^*(f+f'',g+g'')\mathrm{d}f\mathrm{d}g. \tag{3}$$

The *Sum of Coherent Systems* (SOCS) [4] provides an approximation to the Hopkins imaging equations, simplifying the *TCC* spectrum using SVD. Due to the rapid decay of eigenvalue κ_q with q, only the top truncated Q eigenvalues are retained. By applying the IFFT, SOCS can be expressed in spatial domain as:

$$I(x,y) = \sum_{q=1}^{Q} \kappa_q |\phi_q(x,y) \otimes M(x,y)|^2, \tag{4}$$

where $\phi_q(x,y)$ is the spatial distribution of eigenvector $\mathbf{\Phi_q}$. Here, $\otimes$ denotes convolution and $|\cdot|$ is the absolute operator.

Hopkins' Approach vs. Abbe's Approach:. In Equation (4), Hopkins' method reduces computational demands from Abbe's $\mathcal{O}(N_j^2 \cdot N_m^4)$ to $\mathcal{O}(Q \cdot N_m^4)$, with $Q < N_j^2$ for source $\mathbf{J} \in \mathbb{R}^{N_j \times N_j}$ and mask $\mathbf{M} \in \mathbb{R}^{N_m \times N_m}$. Under identical optical conditions, Hopkins' method outperforms Abbe's in speed, leading to its preference in various MO algorithms [2, 5–10]. However, Hopkins' reliance on truncated SVD, as shown in Equation (4), prevents SO due to the inability to calculate source gradients. In contrast, Abbe's method (Equation (2)) inherently suits SO by summing the impacts of all source points to form the aerial image. Moreover, Abbe's richer source information enhances lithography precision, thus improving MO outcomes. Consequently, Abbe's method is essential for SO, higher MO precision, and indispensable for gradient-based bilevel SMO.

2.2 Evaluation Metrics

Definition 1 (Squared L_2 Error (L2))**.** Given target pattern $\mathbf{Z}_t$ and resist image under nominal process condition $\mathbf{Z}$, the squared L_2 error is calculated as $\|\mathbf{Z} - \mathbf{Z}_t\|_2^2$.

Definition 2 (Process Variation Band (PVB))**.** PVB [17] is used in manufacturing to represent the expected range of variation in a production process. PVB denotes the XOR area between the resist images $\mathbf{Z}_{\text{min}}$ and $\mathbf{Z}_{\text{max}}$ under the *min* and *max* process conditions.

Definition 3 (Edge Placement Error (EPE))**.** EPE [17] refers to the deviation between the intended position of a feature on a wafer and its actual position after lithography.

3 Algorithm

3.1 Abbe-based Unified SMO Framework

The Hopkins model, hindered by the need for frequent, inefficient, and non-differentiable SVD truncation of the TCC, limits effective gradient-based co-optimization of source and mask. Addressing this, we introduce a unified SMO framework employing the Abbe model, which facilitates efficient, joint SMO without the TCC's burdensome processing. The framework is further enhanced by incorporating parallel computing techniques for acceleration.

Abbe-based unified SMO:. Utilizing freeform illumination, the pixelated source point is denoted as $J(f,g) \in [0,1]$, while binary mask values $M(x,y)$ are either 0 or 1. To render the SMO framework differentiable, we introduce optimization parameters θ_J for Source **J** and θ_M for Mask **M**, where $\theta_J \in \mathbb{R}^{N_j \times N_j}$, $\theta_M \in \mathbb{R}^{N_m \times N_m}$, and both parameters can assume any real value. Here, N_j and N_m represent the dimensions of the source and mask. Appropriate activation and initialization enable deriving source and mask from these parameters. The `Sigmoid` function $\sigma(x) = 1/(1+\exp(-x))$ is employed for both grayscale source and binary mask, as listed in Table 1.

Table 1: The activation and initialization for Abbe-imaging.

	Activation	Initialization
Mask **M**	$\mathbf{M} = \sigma(\alpha_m \cdot \theta_M)$	$\theta_M(x,y) = m_0,\ \text{if } M_0(x,y) = 1; \text{else } -m_0.$
Source **J**	$\mathbf{J} = \sigma(\alpha_j \cdot \theta_J)$	$\theta_J(f,g) = j_0,\ \text{if } J_0(f,g) = 1;\ \text{else } -j_0.$

In Table 1, α_m and α_j are the sigmoid steepness. Initial values for θ_M are assigned as either m_0 or $-m_0$ based on initial mask pattern $\mathbf{M}_0$. Typically, the initial mask pattern is the same as the binary target pattern $\mathbf{Z}_t$. This mask initialization also facilitates SRAF generation during MO. Grayscale source **J** requires careful selection of steepness α_j and initialization j_0 to maintain its grayscale property. Hyperparameters are detailed in Section 4. The shape of initial source pattern $\mathbf{J}_0$ is derived from parametric templates like annular, quasar, or dipole, characterized by outer and inner radii σ_o and σ_i. Although the `Cosine` function is an alternative source activation, its use may lead to training instability due to gradient issues, leading us to prefer the `Sigmoid` function. The transfer function $H(f,g)$ can be accurately characterized by a low-pass filter, expressed as:

$$H(f,g) = 1, \text{if } \sqrt{f^2+g^2} \le \frac{NA}{\lambda}; H(f,g) = 0, \text{otherwise}, \tag{5}$$

where the cut-off frequency is determined by the projection system's numerical aperture NA and the illumination wavelength λ. By integrating Equation (2), Equation (5) and Table 1, we formulate the Abbe forward imaging f_{abbe}, determining the aerial image **I** as a function of the optimization parameters for source θ_J and mask θ_M: $\mathbf{I} = f_{abbe}(\theta_J, \theta_M)$. Then we can utilize a straightforward threshold model for resist modeling. The `Sigmoid` activation is also adopted to ensure a smooth transition and maintain differentiability: $\mathbf{Z} = \sigma(\beta \cdot (\mathbf{I} - I_{tr}))$, where **Z** represents the resist pattern, I_{tr} denotes the intensity threshold, and β is the steepness.

We have now established a complete Abbe forward imaging model that maps the parameters θ_J and θ_M to the aerial image **I** and the resist image **Z**. To realize SMO, it is essential to define the corresponding objective function and optimization method. We employ the mean squared loss to quantify the discrepancy between the resist pattern **Z** and the target pattern $\mathbf{Z}_t$: $\mathcal{L}_2 = \|\mathbf{Z} - \mathbf{Z}_t\|^2$. In alignment with [2] and considering a ±2% dose range, we pioneer the integration of process window considerations into Abbe-based SMO to mitigate process variation via PVB loss. By substituting $\mathbf{M}_{\min} = d_{\min} \cdot \sigma(\alpha_m \cdot \theta_M)$ and $\mathbf{M}_{\max} = d_{\max} \cdot \sigma(\alpha_m \cdot \theta_M)$ into f_{abbe}, we obtain the resist patterns $\mathbf{Z}_{\min}$, $\mathbf{Z}_{\max}$ under minimum $d_{\min}$ and maximum $d_{\max}$ process conditions. The PVB loss is formulated as:

$$\mathcal{L}_{pvb} = \|\mathbf{Z}_{\max} - \mathbf{Z}_t\|^2 + \|\mathbf{Z}_{\min} - \mathbf{Z}_t\|^2. \tag{6}$$

Consequently, the comprehensive SMO loss $\mathcal{L}_{smo}$ is formulated as:

$$\mathcal{L}_{smo} := \mathcal{L}_{so} := \mathcal{L}_{mo} = \gamma\mathcal{L}_2 + \eta\mathcal{L}_{pvb}, \tag{7}$$

where γ and η are weighting factors for the respective loss components. SO loss $\mathcal{L}_{so}$ and MO loss $\mathcal{L}_{mo}$ can utilize same objective functions. The SMO problem is thus defined as:

$$(\hat{\theta_J}, \hat{\theta_M}) = \underset{(\theta_J, \theta_M)}{\operatorname{argmin}}\ \mathcal{L}_{smo}(\theta_J, \theta_M), \tag{8}$$

where $\hat{\theta_J}, \hat{\theta_M}$ represent the optimal parameter values for the source and mask, respectively.

Abbe acceleration.. The primary computational bottleneck in SMO is the forward imaging model and its gradient calculations. As noted in Equations (2) and (4), contributions from source points can be independently calculated, making the complexity ratio between Abbe's and Hopkins's models σ/Q, where $\sigma \in [0, N_j^2]$ represents the number of effective source points (*i.e.*, where $j_\sigma > 0$). Both models can be accelerated with parallel computing, using multicore CPUs or GPUs. Theoretically, the parallel computation time ratio is $\lceil \frac{\sigma}{P} \rceil / \lceil \frac{Q}{P} \rceil$, where P denotes the maximum number of parallel threads and $\lceil \cdot \rceil$ is the ceiling operator. This suggests that Abbe's runtime can match Hopkins' if $P \ge \sigma$. In our implementation, GPUs are utilized for parallel computation of each effective source point's contribution to the aerial image, owing to their greater thread parallelism, larger memory bandwidth, and faster `FFT` or `IFFT` operations compared to multicore CPUs. Experimental results in Section 4.1 demonstrate that our Abbe-imaging model achieves runtime performance comparable to Hopkins' model.

3.2 Efficient Bilevel SMO

Previous alternating minimization-based SMO:. Since the introduction of SMO technology, the significant computational demands have compelled previous methods to compromise on a simple alternating minimization (AM) strategy. As illustrated in Algorithm 1, AM-based SMO (`AM-SMO`) alternates between two minimization cycles, updating parameters θ_J and θ_M sequentially. This process iterates until reaching the specified convergence criteria for SO and MO, as shown in Figure 2(a). However, `AM-SMO` has several notable drawbacks: 1) `AM-SMO` tends to converge to local minima due to its narrow focus on localized aspects of SO or MO, ignoring the global structure of the problem. 2) The convergence is often slow because the source and mask are highly interdependent. Adjusting $(\theta_M)_k$ as per line 5 makes $(\theta_J)_k$ suboptimal, requiring numerous iterations for stabilization. 3) The absence of global gradient guidance complicates establishing effective early stopping criteria, often resulting in either prolonged optimization or suboptimal convergence.

Algorithm 1 Alternating Minimization-based SMO (`AM-SMO`) [11, 12]

```
1: for $k = 1, 2, 3, \ldots$; do                                   ▷ Alternating SO & MO.
2:     while not converged do                                      ▷ SO iterations.
3:         $(\theta_J)_k \leftarrow \operatorname{argmin}_{\theta_J} \mathcal{L}_{so}\left(\theta_J, (\theta_M)_{k-1}\right)$;   ▷ $\theta_M$ is fixed.
4:     while not converged do                                      ▷ MO iterations.
5:         $(\theta_M)_k \leftarrow \operatorname{argmin}_{\theta_M} \mathcal{L}_{mo}\left((\theta_J)_k, \theta_M\right)$;   ▷ $\theta_J$ is fixed.
   return $(\theta_J)_k, (\theta_M)_k$.
```

Proposed bilevel SMO:. Equation (8) frames SMO as a typical multivariate optimization problem. Yet, the `AM-SMO`, as detailed in Algorithm 1, often leads to suboptimal results due to its localized focus. A gradient-based approach with a global perspective is essential to overcome these limitations. In SMO, SO and MO have a nested

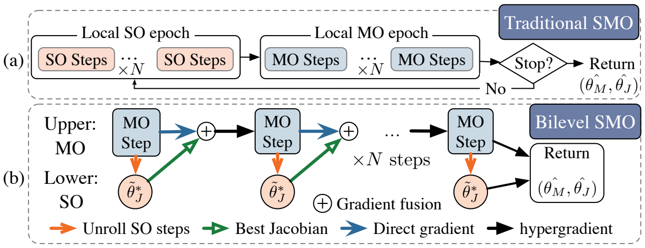


Figure 2: (a)Previous AM-SMO flow. (b)Our BiSMO flow.

relationship. The goal is to rapidly and effectively determine the optimal source response for each altered mask, offering MO global gradient direction to boost overall SMO efficiency. This scenario is particularly well-suited for a bilevel optimization approach. From a bilevel viewpoint, the upper-level MO, constrained by the optimal solutions from the lower-level SO, forms a dependent hierarchy, as illustrated in Figure 1(b) and Figure 2(b). This structure allows the MO to offer a global perspective by solving SO, guiding the optimization beyond local minima. Consequently, Equation (8) can be reformulated in a bilevel context:

$$\begin{aligned} &\min_{\theta_M} \mathcal{L}_{mo}(\theta_J^*(\theta_M), \theta_M), && \triangleright \textit{ Upper-Level: MO} \\ &\text{s.t. } \theta_J^*(\theta_M) = \operatorname{argmin}_{\theta_J} \mathcal{L}_{so}(\theta_J, \theta_M). && \triangleright \textit{ Lower-Level: SO} \end{aligned} \tag{9}$$

In BLO, the inner and outer loops are termed lower-level and upper-level subproblems. Here the inner loop seeks optimal source parameters θ_J^* for the current θ_M while the outer loop endeavors to optimize the mask parameters θ_M with the best-response source $\theta_J^*(\theta_M)$. The gradient of the outer loop in bilevel SMO, also referred to as the hypergradient, which is derived from the fusion of gradients from the upper and lower levels, is then calculated as:

$$\nabla_{\theta_M}\mathcal{L}_{mo} = \frac{\partial \mathcal{L}_{mo}}{\partial \theta_M} + \frac{\partial \mathcal{L}_{mo}}{\partial \theta_J}\frac{\partial \theta_J^*(\theta_M)}{\partial \theta_M}. \tag{10}$$

Within the Abbe-based SMO framework introduced in Section 3.1, the direct gradient $\frac{\partial \mathcal{L}_{mo}}{\partial \theta_M}$ and $\frac{\partial \mathcal{L}_{mo}}{\partial \theta_J}$ can be efficiently computed. However, two principal challenges arise: (1) the precise approximation of the SO optimal solution $\theta_J^*(\theta_M)$, and (2) differentiating the best-response Jacobian: $\frac{\partial \theta_J^*(\theta_M)}{\partial \theta_M}$. To address the former, we approximate θ_J^* by unrolling a few SO gradient steps T, significantly reducing the computational cost. Fortunately, extensive research [14–16] indicates that BLO, with weight sharing, can effectively adapt θ_J to θ_J^* with a small unrolling step T. For the latter issue, we propose three methods to compute the best-response Jacobian: bilevel SMO using finite difference (`BiSMO-FD`), `BiSMO-NMN` utilizing Neumann series, and `BiSMO-CG` using conjugate gradients.

3.2.1 BiSMO-FD:

The finite difference (`FD`) strategy uses a single inner SO step, $\theta_J^*(\theta_M) = \theta_J - \xi\nabla_{\theta_J}\mathcal{L}_{so}$, and obtaining $\frac{\partial \theta_J^*(\theta_M)}{\partial \theta_M} = -\xi\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_M \partial \theta_J}$, the `BiSMO-FD` calculates the hypergradient as:

$$\nabla_{\theta_M}\mathcal{L}_{mo}^{\text{FD}} = \frac{\partial \mathcal{L}_{mo}}{\partial \theta_M} - \xi\frac{\partial \mathcal{L}_{mo}}{\partial \theta_J}\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_M \partial \theta_J}, \tag{11}$$

where ξ is the inner-loop learning rate for θ_J. While increasing the number of inner SO steps can lead to a more precise approximation for θ_J^*, akin to `AM-SMO`, it results in a linear increase in memory and computational load, becoming impractical due to the need to store optimization paths and all intermediate gradients for differentiation following the chain rule.

In contrast, using the implicit function theorem (IFT), the hypergradient can be computed without retaining intermediate gradients, thus independent of the optimization path and significantly reducing memory usage. The IFT-based hypergradient for SMO can be formulated as the following lemma.

Lemma 1. *Implicit Function Theorem: Consider $\theta_J^*(\theta_M)$ defined in Equation (9), with first-order optimality condition $\frac{\partial \mathcal{L}_{so}(\theta_J^*,\theta_M)}{\partial \theta_J} = 0$,*

$$\frac{\partial}{\partial \theta_M}\left[\frac{\partial \mathcal{L}_{so}(\theta_J^*(\theta_M), \theta_M)}{\partial \theta_J}\right] = 0, \Rightarrow \frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_M \partial \theta_J} + \frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\frac{\partial(\theta_J^*(\theta_M))}{\partial \theta_M} = 0,$$

$$\Rightarrow \textit{best-response Jacobian: } \frac{\partial(\theta_J^*(\theta_M))}{\partial \theta_M} = -\left[\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\right]^{-1}\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_M \partial \theta_J}.$$

With Equation (10), we have hypergradient formulated by:

$$\nabla_{\theta_M}\mathcal{L}_{mo} = \frac{\partial \mathcal{L}_{mo}}{\partial \theta_M} - \frac{\partial \mathcal{L}_{mo}}{\partial \theta_J}\left[\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\right]^{-1}\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_M \partial \theta_J}. \tag{12}$$

However, the inverse Hessian $\left[\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\right]^{-1}$ in Equation (12) is hard to calculate. In Equation (11), `BiSMO-FD` employs finite difference to naively approximate the inverse $\left[\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\right]^{-1} = \xi\mathfrak{I}$, where $\mathfrak{I}$ denotes the identity matrix. For a more precise approximation of the inverse, we introduce two IFT-based methods: Neumann series (`BiSMO-NMN`) and conjugate gradient (`BiSMO-CG`), to reformulate the hypergradient.

3.2.2 BiSMO-NMN

Lemma 2. *Neumann series [14]: With a matrix $\mathbf{A}$ that $\|\mathfrak{I} - \mathbf{A}\| < 1$, we have $\mathbf{A}^{-1} = \sum_{k=0}^{\infty}(\mathfrak{I} - \mathbf{A})^k$.*

Based on Lemma 2, with small enough learning rate, the hypergradient in Equation (12) for `BiSMO-NMN` is formulated by:

$$\nabla_{\theta_M}\mathcal{L}_{mo} = \frac{\partial \mathcal{L}_{mo}}{\partial \theta_M} - \frac{\partial \mathcal{L}_{mo}}{\partial \theta_J}\sum_{k=0}^{\infty}\left[\mathfrak{I} - \frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\right]^k\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_M \partial \theta_J}. \tag{13}$$

The approximation of the hypergradient $\nabla_{\theta_M}\tilde{\mathcal{L}}_{mo}$ can be derived by considering only the first K terms of the Neumann series, thereby avoiding the need to calculate the inverse of the Hessian as:

$$\nabla_{\theta_M}\tilde{\mathcal{L}}_{mo}^{\text{NMN}} = \frac{\partial \mathcal{L}_{mo}}{\partial \theta_M} - \frac{\partial \mathcal{L}_{mo}}{\partial \theta_J}\sum_{k=0}^{K}\left[\mathfrak{I} - \frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\right]^k\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_M \partial \theta_J}. \tag{14}$$

3.2.3 BiSMO-CG:

Instead of calculating the Neumann series, another efficient way to approximate the inverse Hessian is to solve the linear systems. Specifically, $\frac{\partial \mathcal{L}_{mo}}{\partial \theta_J}\left[\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\right]^{-1}$ can be computed as the solution to the linear system $\left[\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\right]\mathbf{w} = \frac{\partial \mathcal{L}_{mo}}{\partial \theta_J}$. The vector $\mathbf{w}$ can be obtained by solving the optimization problem:

$$\min_{\mathbf{w}} \mathbf{w}^{\top}\left[\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\right]\mathbf{w} - \mathbf{w}^{\top}\frac{\partial \mathcal{L}_{mo}}{\partial \theta_J}. \tag{15}$$

The conjugate gradient (`CG`) algorithm is well-suited for this task, given its efficient iteration complexity and use of Hessian-vector

Algorithm 2 Bilevel SMO via BiSMO-NMN, BiSMO-CG

```
Input: Unroll step $T$, stepsizes $\xi_J, \xi_M$, initializations $\theta_J^0, \theta_M^0, \mathbf{w}_0$, term $K$
Output: $\theta_J, \theta_M$
1: while not converged do
2:     for Inner step $t = 1, \ldots, T$ do                ▷ Unroll $T$ steps of inner-SO.
3:         Update $\theta_J^t \leftarrow \theta_J^{t-1} - \xi_J \nabla_{\theta_J} \mathcal{L}_{so}(\theta_J^{t-1}, \theta_M)$; // Or Adam.
4:     Approximate $\theta_J^* \leftarrow \theta_J^t$; Re-initialize $\theta_J^0 \leftarrow \theta_J^t$;
5:     if BiSMO-NMN then                                   ▷ Hypergradient via BiSMO-NMN.
6:         1) Get $K$ Neumann series via HVP: $\frac{\partial \mathcal{L}_{mo}}{\partial \theta_J} \sum_{k=0}^{K} \left[\mathbb{I} - \frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\right]^k$;
7:         2) Get Jacobian-vector product JVP in Equation (14);
8:         3) $\nabla_{\theta_M} \tilde{\mathcal{L}}_{mo}^{\mathrm{NMN}} \leftarrow$ Equation (14);
9:     if BiSMO-CG then                                    ▷ Hypergradient via BiSMO-CG.
10:        1) Solve $\mathbf{w}_K$ from $\left[\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\right] \mathbf{w} = \frac{\partial \mathcal{L}_{mo}}{\partial \theta_J}$, via $K$ steps of CG starting from $\mathbf{w}_0$; then re-initialize $\mathbf{w}_0 \leftarrow \mathbf{w}_K$;
11:        2) Get Jacobian-vector product JVP:$\left[\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_M \partial \theta_J}\right] \mathbf{w}$;
12:        3) $\nabla_{\theta_M} \tilde{\mathcal{L}}_{mo}^{\mathrm{CG}} \leftarrow$ Equation (16);
13:    Update $\theta_M \leftarrow \theta_M - \xi_M \nabla_{\theta_M} \tilde{\mathcal{L}}_{mo}$; // Or Adam.
```

Table 2: Details of the Dataset.

Dataset	From	Area†	Test num.	Layer	CD‡	tile
ICCAD13	[2, 5]	202655	10	Metal	$32nm$	$4\mu m^2$
ICCAD-L	[7, 10]	475571	10	Metal	$32nm$	$4\mu m^2$
ISPD19	[18]	698743	100	Metal+Via	$28nm$	$4\mu m^2$

Area†: average area: unit nm^2; CD‡: critical dimension.

products (HVP) for $\left[\frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J}\right] \mathbf{w}$. Such HVP can be obtained cheaply without explicitly forming or storing the Hessian. The hypergradient in Equation (12) for BiSMO-CG is then computed as:

$$\nabla_{\theta_M} \tilde{\mathcal{L}}_{mo}^{\mathrm{CG}} = \frac{\partial \mathcal{L}_{mo}}{\partial \theta_M} - \left[\operatorname*{argmin}_{\mathbf{w}} \left(\mathbf{w}^\top \frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_J \partial \theta_J} \mathbf{w} - \mathbf{w}^\top \frac{\partial \mathcal{L}_{mo}}{\partial \theta_J}\right)\right] \frac{\partial^2 \mathcal{L}_{so}}{\partial \theta_M \partial \theta_J}. \quad (16)$$

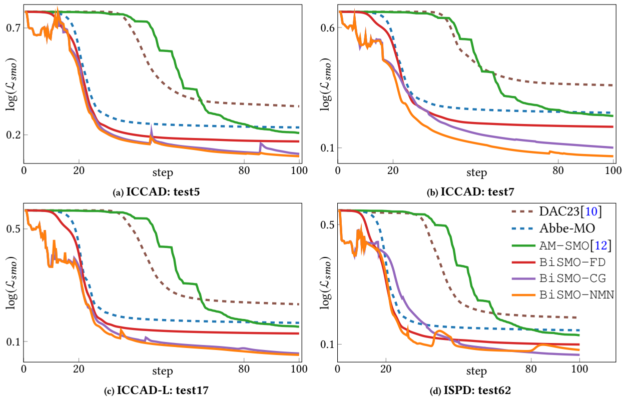
3.2.4 BiSMO-FD vs. BiSMO-NMN vs. BiSMO-CG vs. AM-SMO: The optimization flow of BiSMO is demonstrated in Figure 2(b) and Algorithm 2. When $K = 0$, the $\nabla_{\theta_M} \tilde{\mathcal{L}}_{mo}^{\mathrm{NMN}}$ in Equation (14) reduces to match $\nabla_{\theta_M} \mathcal{L}_{mo}^{\mathrm{FD}}$ in Equation (11). In Algorithm 2, set $K = 0; T = 1$; the FD can be executed through the same process as NMN. Both NMN and CG use T unroll SO steps to approximate θ_J^* (line 2), and employ HVP or Jacobian-vector product (JVP) for computational acceleration. The key difference between them lies in approximating the inverse Hessian: NMN uses the first K terms of the Neumann series (line 6), while CG applies K CG steps (line 10). Compared to AM-SMO, the unroll strategy in line 2, due to the small T, substantially reduces runtime by avoiding full SO cycle convergence. Furthermore, in line 8 and line 12, IFT-based gradient fusion provides BiSMO with a global perspective and a more thorough exploration of the solution space, thereby facilitating enhanced and accelerated convergence.

4 Experiments

The BiSMO is implemented in PyTorch framework and tested on an Nvidia RTX4090 GPU card across three datasets as listed in Table 2. The hyperparameters settings are as follows $\gamma = 1000; \eta = 3000; \lambda = 193nm; NA = 1.35; \sigma_o = 0.95; \sigma_i = 0.63. Q = 24; N_j = 35; N_m = 2048, \alpha_m = 9; m_0 = 1; \alpha_j = 2; j_0 = 5; \beta = 30; P = 256; \xi = \xi_M = \xi_J = 0.1; K = 5; T = 3$. All the tiles are converted to 2048×2048-pixel images.

4.1 Results Comparison with SOTA

We have conducted a comparative analysis of the performance between our Abbe-based MO and the previous SOTA MO methods DAC23-MILT [10] and NILT [7] in Table 3 and Table 4. Furthermore, we compare the performance of BiSMO with the previous SOTA AM-SMO [12, 13]. AM-SMO is implemented in two ways: one involves hybrid Abbe-SO and Hopkins-MO [13], while the other employs the Abbe model for both SO and MO [12]. To highlight advantages of BiSMO, Figure 3 shows log-scaled loss $\log(\mathcal{L}_{smo})$ convergence compared to SOTA MO [10] and AM-SMO [12] using random cases from test datasets in Table 2, with a 0.01 learning rate. Result samples are depicted in Figure 4, appropriately scaled and cropped to enhance visualization.



(a) ICCAD: test5

(b) ICCAD: test7

(c) ICCAD-L: test17

(d) ISPD: test62

Figure 3: Loss comparison between different MO methods (dashed lines) and SMO methods (solid lines).

Effectiveness of Abbe-MO. Table 3 and Table 4 demonstrate that our Abbe-MO achieved a 25% reduction in L2, 19% in PVB, and 24% in EPE when compared to the SOTA MO DAC23-MILT [10]. Furthermore, within the AM-SMO, the Abbe-based SMO[12] surpassed the Abbe-Hopkins hybrid SMO[13] by decreasing L2 by 28%, PVB by 21%, and EPE by 29%. In Figure 3, all cases indicate that Abbe-MO converges more rapidly and effectively than SOTA MO [10]. This can be attributed to the fact that truncated decomposition in Hopkins' approach leads to a loss of accuracy in lithography, thereby allowing our lossless Abbe method to achieve superior MO and SMO results.

BiSMO vs. AM-SMO vs. MO. Table 3, Table 4 and Figure 3 reveal that BiSMO variants significantly outperform AM-SMO[12, 13] in error reduction, with BiSMO-NMN achieving decreases of 41% in L2, 46% in PVB, and 37% in EPE, and even the basic BiSMO-FD showing reductions of 36% in L2, 34% in PVB, and 27% in EPE. Figure 3 illustrates AM-SMO's [12] 'zigzag' loss curve, a result of its alternating optimization, which ultimately settles below Abbe-MO yet above BiSMO variants. This suggests that while AM-SMO's broader solution space improves outcomes compared to pure MO. However, its alternating approach risks entrapment in local minima, hindering it from reaching the optimal results achievable by BiSMO. This fact underscores the superior performance of the BiSMO method. Additionally, BiSMO demonstrates a significant improvement over the SOTA MO [10], achieving a ~50% reduction in all error metrics.

Runtime comparison. In our implementation, Abbe-MO and Hopkins-MO have been accelerated to average 0.16s and 0.12s per

Table 3: Result comparison with SOTA.

Bench	MO						AM−SMO				BiSMO (Ours)					
	NILT [7]		DAC23-MILT [10]		Abbe-MO(Ours)		Abbe-Hopkins★ [13]		Abbe-Abbe [12]		BiSMO−FD		BiSMO−CG		BiSMO−NMN	
	L2	PVB	L2	PVB	L2	PVB	L2	PVB	L2	PVB	L2	PVB	L2	PVB	L2	PVB
ICCAD13	37515	50964	28362	40044	20419	29697	27299	37278	17539	23944	13828	17872	13603	16274	**13059**	**15839**
ICCAD-L	71570	108162	53143	87010	44478	66092	48879	77062	40455	58560	29779	42643	29762	40543	**28946**	**38706**
ISPD19	97891	119732	85234	105592	61374	93132	79634	97073	55588	84402	39959	64211	39488	61190	**38737**	**59832**
Average	68992	92953	55580	77549	42090	62974	51937	70471	37861	55635	27855	41576	27618	39336	**26914**	**38126**
Ratio	2.56	2.44	2.07	2.03	1.56	1.65	1.93	1.85	1.41	1.46	1.03	1.09	1.03	1.03	**1.00**	**1.00**

Abbe-Hopkins★ [13]: AM−SMO employs Abbe model for SO and Hopkins model for MO. L2 and PVB unit: nm^2.

Table 4: EPE and runtime comparison.

	MO			AM−SMO		BiSMO		
	NILT [7]	DAC23 [10]	Abbe -MO	A∼H★ [13]	A∼A† [12]	FD	CG	NMN
EPE avg.	10.1	3.6	2.8	3.3	2.4	1.8	1.6	**1.6**
Ratio	6.2	2.2	1.7	2.0	1.5	1.1	1.0	**1.0**
TAT‡ avg.	12.4	3.8	11.7	287	122.5	**12.6**	15.3	14.7
Ratio	0.84	0.26	0.80	19.52	8.33	**0.86**	1.04	1.00

A∼H★: Abbe-Hopkins; A∼A†: Abbe-Abbe; TAT‡: Turn around time (s).

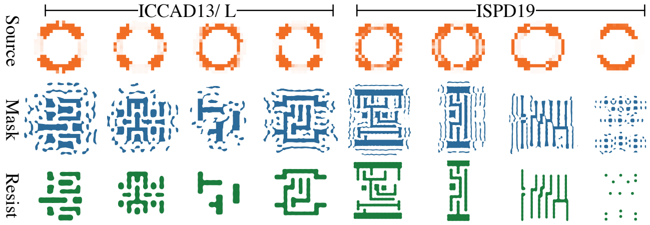


Figure 4: Result samples from ICCAD13 and ISPD19 datasets.

MO iteration, aligning with theoretical derivations in Section 3.1. As shown in Table 4, for parity, we have applied GPU acceleration to AM−SMO [12, 13] with settings identical to BiSMO. In their original implementations, the runtime for [12] was 910s, and [13] was 69 minutes. Utilizing our accelerated Abbe imaging, this has been accelerated to 122.5s and 287s, respectively. Despite these improvements, BiSMO, leveraging hypergradient, still achieves faster convergence than [12, 13], boosting throughput by 8.3 times compared to the Abbe-based AM−SMO [12]. Furthermore, it's about 19.5 times quicker than the Abbe-Hopkins hybrid AM−SMO [13], which is slowed down by its complex iterative TCC generation and decomposition.

4.2 Ablation Study

BiSMO−FD vs. BiSMO−NMN vs. BiSMO−CG: Figure 3, Table 3, and Table 4 clearly show that NMN typically outperforms other methods, followed by CG, with FD being relatively weaker among all BiSMO variants. The relative instability of CG is indicated by its largest standard deviation (STD) in Figure 5. Meanwhile, FD boasts the shortest runtime (Table 4), and CG's advantage lies in outperforming NMN in some cases, as shown in Figure 3(d).

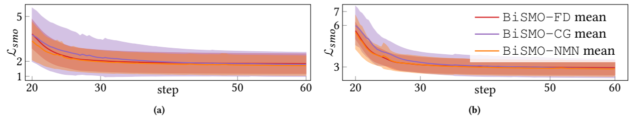


Figure 5: Mean and STD of (a) ICCAD (b) ICCAD-L datasets.

5 Conclusion

In this paper, we first establish a unified SMO framework utilizing Abbe-imaging, enabling simultaneous SO and MO gradient computation with enhanced lithographic precision and rapid calculation. Building on this foundation, BiSMO is introduced, conceptualizing SMO as a bilevel problem and proposing three innovative methods for calculating source-mask best-response gradients, effectively addressing bilevel SMO challenges. BiSMO's gradient-based approach, with its global perspective and improved exploration of solution space facilitates navigation out of local minima, ensuring better and faster converging SMO outcomes. This method surpasses traditional AM−SMO limitations, positioning bilevel SMO as a future promising mainstream approach in the field.

References

[1] C. H. Wallace, P. A. Nyhus, and S. S. Sivakumar, "Sub-resolution assist features," Dec. 15 2009, US Patent.

[2] J.-R. Gao, X. Xu, B. Yu, and D. Z. Pan, "MOSAIC: Mask optimizing solution with process window aware inverse correction," in *Proc. DAC*, 2014.

[3] P. Evanschitzky, A. Erdmann, and T. Fuehner, "Extended abbe approach for fast and accurate lithography imaging simulations," in *25th European Mask and Lithography Conference*, 2009.

[4] N. Cobb, "Sum of coherent system decomposition by SVD," *Berkeley CA*, 1995.

[5] H. Yang, S. Li, Y. Ma, B. Yu, and E. F. Young, "GAN-OPC: Mask optimization with lithography-guided generative adversarial nets," in *Proc. DAC*, 2018.

[6] G. Chen, W. Chen, Y. Ma, H. Yang, and B. Yu, "DAMO: Deep agile mask optimization for full chip scale," in *Proc. ICCAD*, 2020.

[7] 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 *Proc. ICCAD*, 2020.

[8] G. Chen, Z. Yu, H. Liu, Y. Ma, and B. Yu, "DevelSet: Deep neural level set for instant mask optimization," in *Proc. ICCAD*, 2021.

[9] Z. Yu, G. Chen, Y. Ma, and B. Yu, "A gpu-enabled level set method for mask optimization," in *Proc. DATE*, 2021.

[10] S. Sun, F. Yang, B. Yu, L. Shang, and X. Zeng, "Efficient ILT via multi-level lithography simulation," in *Proc. DAC*, 2023.

[11] Z. Wang, X. Ma, R. Chen, S. Zhang, and G. R. Arce, "Fast pixelated lithographic source and mask joint optimization based on compressive sensing," *IEEE TAI*, vol. 6, pp. 981–992, 2020.

[12] Y. Sun, Y. Li, G. Liao, M. Yuan, P. Wei, Y. Li, L. Zou, and L. Liu, "Sampling-based imaging model for fast source and mask optimization in immersion lithography," *Appl. Opt.*, 2022.

[13] M. Ding, Z. Niu, F. Zhang, L. Zhu, W. Shi, A. Zeng, and H. Huang, "Gradient-based source mask and polarization optimization with the hybrid hopkins–abbe model," *JM3*, 2020.

[14] J. Lorraine, P. Vicol, and D. Duvenaud, "Optimizing millions of hyperparameters by implicit differentiation," in *Proc. AISTATS*, 2020.

[15] M. Zhang, S. W. Su, S. Pan, X. Chang, E. M. Abbasnejad, and R. Haffari, "iDARTS: Differentiable architecture search with stochastic implicit gradients," in *Proc. ICML*, 2021.

[16] S. K. Choe, S. V. Mehta, H. Ahn, W. Neiswanger, P. Xie, E. Strubell, and E. Xing, "Making scalable meta learning practical," in *Proc. NeurIPS*, 2023.

[17] S. Banerjee, Z. Li, and S. R. Nassif, "ICCAD-2013 CAD contest in mask optimization and benchmark suite," in *Proc. ICCAD*, 2013.

[18] W.-H. Liu, S. Mantik, W.-K. Chow, Y. Ding, A. Farshidi, and G. Posser, "ISPD 2019 initial detailed routing contest and benchmark with advanced routing rules," in *Proc. ISPD*, 2019.