BEAM: Bidirectional MEEF-Driven Mask Optimization for Curvilinear Photonic Design

Xiaoxiao Liang[1,2], Yang Luo[1], Bei Yu[3], Yuzhe Ma[1†]
[1]Microelectronics Thrust, HKUST(GZ)
[2]Guangdong-Macao Joint Laboratory for Modular Chip Design and Testing, HKUST(GZ)
[3]The Chinese University of Hong Kong
Email: yuzhema@hkust-gz.edu.cn

2026 31st Asia and South Pacific Design Automation Conference (ASP-DAC) | 979-8-3315-9123-6/26/$31.00 ©2026 IEEE | DOI: 10.1109/ASP-DAC66049.2026.11420526

***Abstract*—The photonic integrated circuit (PIC) is a promising direction for future computing and interconnect, which involves many curvilinear geometries to modulate and transmit signals. To ensure the functionality, the PIC manufacturing requires very meticulous optimization to refrain from geometry distortion resulting from the lithography process. While conventional optical proximity correction (OPC) methods can handle curvilinear features, they face challenges in mask manufacturability, computational cost, and the ability to correct any-angle edge placement error (EPE). This paper proposes BEAM, a native framework designed for photonic designs with curvilinear patterns, including lossless curvilinear pattern representation and a powerful OPC solver. BEAM uses control points to represent curvilinear mask shapes directly, avoiding Manhattanization and approximation errors. Instead of manually specifying movement directions, control points are bidirectionally updated along two orthogonal basis directions, ensuring versatile corrections. To further enhance efficiency, we propose a fast batch-based sensitivity measurement strategy that effectively guides the movement of control points while substantially reducing the computational overhead. The effectiveness of BEAM is demonstrated on multiple fundamental layout components of photonic designs, achieving state-of-the-art correction performance in terms of mask quality and computational efficiency.**

I. INTRODUCTION

Photonic integrated circuits (PICs) have emerged as a compelling technology for future computing and interconnect. To ensure functionality, PIC manufacturing requires meticulous optimization to prevent geometry distortion resulting from the lithography process, which necessitates advanced resolution enhancement techniques (RETs) to achieve the desired pattern after lithography [1], [2]. Unlike electronic circuits, where layouts are usually Manhattan-style, PIC structures often rely on curvilinear geometries [3]. For instance, PIC layouts use waveguides with smooth bends, such as Euler or circular curves, to minimize scattering and reflection loss. Design rules in PIC layouts often specify minimum bend radii and avoid sharp corners to reduce radiation loss. Additionally, smooth transitions between segments are essential for maintaining optical mode integrity. These characteristics of PIC designs pose unique challenges to photomask optimization for manufacturing as well as optical proximity correction (OPC).

Despite the existence of traditional OPC techniques for electronic ICs, several challenges emerge when applying them to PIC scenarios, which motivates the development of specialized OPC solutions tailored for photonic designs. One of the key challenges in this domain is the absence of a unified standard for the primary evaluation metric in the OPC task.Currently, there is no standardized definition or methodology for measuring edge placement error (EPE) on curvy patterns in photonic designs, as traditional EPE measurement methods are tailored to straight edges. Other challenges lie in the OPC methodologies. Existing OPC methods mainly involve the inverse lithography technique (ILT) and model-based OPC (MBOPC). ILT pixelizes the layout and takes the OPC problem as an image-transferring task [4]–[8], and performs pixel-level optimization using numerical methods. MBOPC involves fracturing the edges in rectilinear designs and moving the segments inwards or outwards, utilizing the simulation results of a lithography model [9]–[13]. Although both have demonstrated their potential to handle PIC layouts to some extent, their effectiveness is often constrained. For instance, ILT performs free-form optimization and can naturally fit the design. However, this freedom often makes mask manufacturing difficult due to the complex shapes [14]. Meanwhile, the repeated calculation of the gradient back-propagation creates bottlenecks in both computational cost and runtime.

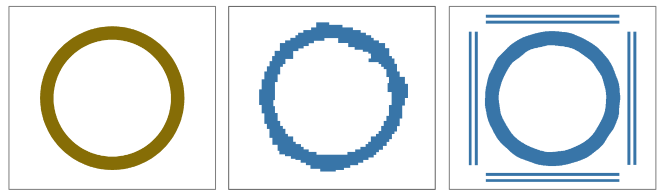

Fig. 1 The visualization of PIC design component, the optimized mask from the Manhattanized MBOPC solver, and PIC design-native OPC.

Although efforts have been made to extend MBOPC solutions to support PIC designs, their capability is still limited. Fundamentally, there are two issues to be tackled, including the representation of the mask and computational efficiency. First, an appropriate representation of the mask is crucial. In electronic designs, conventional MBOPC solvers represent the mask with a series of successive linear segments and move the segments unidirectionally. However, the features in photonic designs can be oriented at arbitrary angles. Previous work proposed an adaptation strategy to PIC designs, which approximates the curves by Manhattanization [15] and optimizes the movement of line segments. In [16], a piece-wise linear approximation of the target shape is performed, and heuristics are designed to move the vertices or segments for correction. While the distortion is less severe than rectilinear masks, the moving directions of vertices or segments in existing literature are often manually specified and fixed [16], which still restricts the optimization to a unidirectional movement, constraining the potential solution space exploration. Therefore, a proper representation of the mask and its modifications for curvilinear designs is needed. It is expected to capture not only the curvy nature of the target boundary but also support alternative adjustments at arbitrary angles. The second issue is computational efficiency, which arises from the first problem—the support of any-angle mask modifications. If the optimization is free to adjust the mask patterns in any direction, the solution space expands significantly. Therefore, a careful balance of computational efficiency and optimization flexibility is in demand.

In this paper, we propose a high-performance and efficient OPC framework tailored for photonic designs. First, we design an EPE measuring strategy for curvy patterns in photonic designs, which can well reflect the mismatch between the curvy target and printed

[†]Corresponding author

contour. Secondly, the mask geometries are directly represented using curve control points, avoiding error-prone Manhattanization or approximation. The control points can be moved in any direction in the plane, and hence guarantee a sufficiently large design space to obtain a desired mask. Fig. 1 demonstrates a visualization of our mask in comparison with the Manhattanized OPC result. To enable a stable and efficient optimization, we develop a bidirectional mask error enhancement factor (MEEF) analysis approach to guide the movement of control points, which supports any-directional control points moving by analyzing individual movement along two orthogonal basis directions. Considering that the bidirectional MEEF analysis may introduce additional overhead, we develop a fast yet effective batch-based strategy to accelerate the process, which provides very accurate moving guidance for control points. To summarize, this paper presents the following contributions:

- We propose a unified framework for OPC on curvilinear PIC designs, including lossless curvilinear pattern representation and an effective OPC solver. To the best of our knowledge, this is the first work to demonstrate a PIC-native OPC approach.
- We design an EPE measurement strategy that is compatible with curvilinear shapes, enabling the extension of other MBOPC flows to photonic designs.
- With mask geometries represented by control points, we develop a bidirectional MEEF analysis approach to guide control point movement along two orthogonal directions to realize versatile corrections.
- We introduce a fast batch-based strategy to reduce the computational overhead due to correction flexibility while ensuring accurate optimization.
- We verify the effectiveness of BEAM on fundamental layout components of PIC designs, achieving state-of-the-art correction performance.

II. Preliminary

A. MEEF-based OPC

Mask error enhancement factor (MEEF) describes how changes in the mask patterns translate to the wafer plane. The MEEF-OPC theory models the EPE change upon mask update using a linear approximation [17]–[19]. Given a set of EPE measuring sites, we use $f(M_t)$ to represent the observed EPEs on each site after lithography simulation with mask M_t in iteration t. Taylor expansion of $f(M_{t+1})$ at M_{t+1} gives:

$$f(M_{t+1}) = f(M_t) + f'(M_t) \times \Delta M_t + \epsilon, \tag{1}$$

where ΔM_t represents the mask modification at iteration t, and ϵ indicates the high order terms. Ignoring ϵ makes Equation (1) a linear approximation of EPE in the next iteration after applying ΔM_t to the mask. Since $f(M_t)$ is already known, reducing the EPE can be recognized by an estimation of $f'(M_t)$, which can be seen as the EPE response imposed by small perturbations on M_t. Therefore, accurately and efficiently modeling $f'(M_t)$ becomes critical in predicting and minimizing the EPE.

B. Problem Formulation

In this paper, we focus on the mask optimization problem for curvilinear photonic designs. Given a curvilinear layout, the objective is to generate an optimized photomask. With an appropriate EPE measuring strategy, the distortion of the wafer image after lithography simulation is minimized.

III. Methods

The notations mentioned in this section are summarized in TABLE I. Fig. 2 shows the workflow of BEAM. Given a target layout, rule-based sub-resolution assist features (SRAFs) are first added to the mask. Afterward, the control points are evenly placed at the target boundaries and are connected using B-spline interpolation for the mask patterns [20]. In each iteration, the EPE sensitivity is modeled first. Then the movement of control points is determined and committed to the mask. The mask is updated iteratively until the quality is satisfactory.

TABLE I Notations

Symbol	Description	Symbol	Description
M	Photomask	N	# control points
$\boldsymbol{\Delta P}$	Optimized mask update	K	# EPE measure points
e	EPE measuring site	$\boldsymbol{\Delta e}$	EPE change
p	Control point	$\boldsymbol{\Delta p}$	Control point perturbation
n	Sampling batch size	k	# batches sampled per iteration
$f(\cdot)$	Simulated EPE	$\boldsymbol{\mathcal{J}}$	Tensor for EPE sensitivity

To demonstrate the BEAM framework properly, the following sections start by introducing the fundamental concept of the EPE measuring strategy for curvilinear design. Next, the mask initialization and formulation are illustrated, followed by the optimization framework.

A. EPE Measurement on Curvilinear Design

As a fundamental component of the OPC problem, we first propose an EPE measuring strategy dedicated to curvy edges in photonic layouts. As Fig. 3 shows, two groups of measuring sites are placed evenly along the target boundary in two opposite directions: the clockwise direction and the inverse direction, starting from the same location. Merging the sampled sites in the two directions gives the EPE measuring point set $\{e_i\}_{i=1}^{K}$. Sites on straight edges follow the standard EPE measuring method which counts the wafer image displacement orthogonal to the target boundary. As depicted in Fig. 3, for each site e_i falling on curvy edges, the strategy begins by identifying its predecessor e_{i-1} and the successor e_{i+1} sampled from the same direction. The EPE at e_i is then measured in the direction orthogonal to the line connecting e_{i-1} and e_{i+1}. The EPE on any measuring site is a signed integer, where a positive number refers to an outer EPE, and a negative number is an inner one. This measuring paradigm is easy to implement since the coordinates of the two neighboring points needed are already known, and it only requires a simple calculation of the normal direction.

B. Curvilinear Mask Initialization

The initialization setup of our proposed framework includes rule-based SRAF insertion, control points placement, and B-spline-based points connection.

For simplicity, we initialize the mask by rule-based SRAF insertion first. Bar-shaped SRAFs are placed surrounding the target pattern area, with their width and spacing to the main patterns fixed, as revealed in Fig. 2. The target pattern edges are then split into intervals, as demonstrated in Fig. 4(a). For corners, short segments are created with length l_c. The remaining parts of the target boundary are evenly split by length l_u, which is longer than l_c. Control points are placed at the midpoint of each interval, represented by $P = \{p_i\}_{i=1}^{N}$. They are then connected with B-spline-based interpolation, as depicted in Fig. 4(b). In BEAM, we utilize the cubic B-spline interpolation, where the curve function $p(\cdot)$ between any neighboring pair of control points (p_i, p_{i+1}) is determined by their predecessor p_{i-1} and successor p_{i+1}. The curve function can be mathematically formulated as:

$$p(t) = \frac{1}{6}\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \\ p_{i+2} \end{bmatrix},$$

$$t \in [0, 1],$$

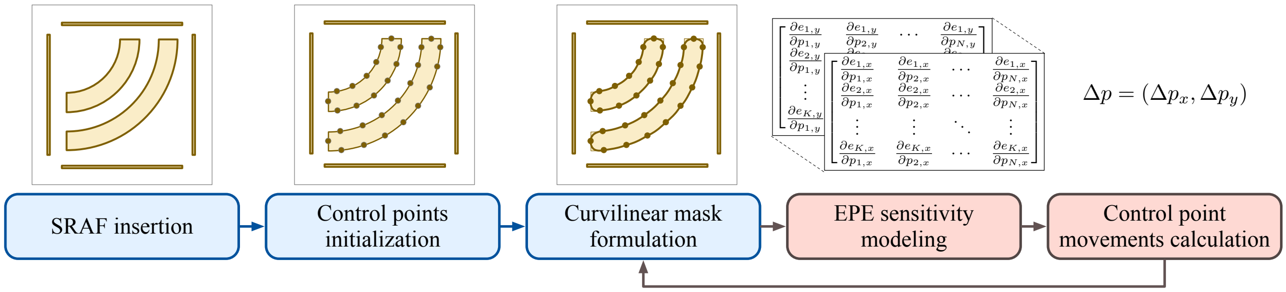

Fig. 2 The workflow of BEAM. After rule-based SRAF insertion, the control points are iteratively updated by modeling the EPE sensitivity and calculating the appropriate movements. The optimized movements are then updated to the mask, and the next iteration is launched until a satisfying mask quality is achieved.

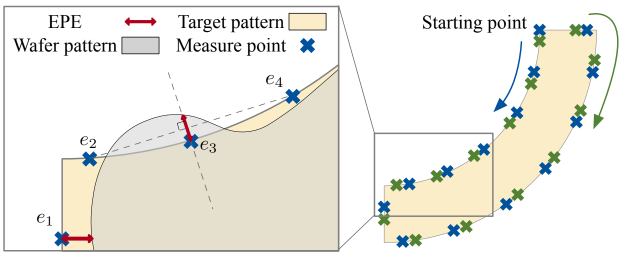

Fig. 3 Illustration of EPE measurement. Two sets of EPE measure points are evenly placed on the target boundary in two directions, starting from the same location. e_1 to e_4 are four successive EPE measure points. e_1 falls on a line-end segment, the others are on the curvy edge. e_1 adopts the standard measuring strategy. EPEs at e_2 to e_4 follow the measurement for the curvy edges we propose.

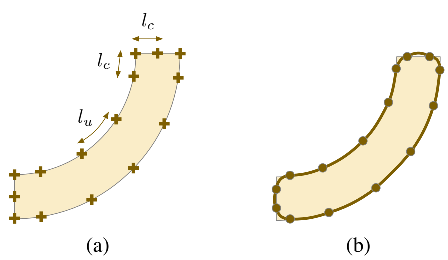

Fig. 4 The mask formulation steps in BEAM. (a) The segmentation process, where corner segments are shorter and others are evenly split. (b) The initial mask. Midpoints of the segments are considered as control points. B-spline interpolation is used for connection.

where $p(0) = p_i$ and $p(1) = p_{i+1}$. This formulation ensures smooth boundaries and transitions among control points.

C. BEAM Optimization Framework

In this section, we introduce the optimization flow of BEAM. First, each component in building the optimization framework is involved, including the optimization instance definition and the correction objective formulation. The optimization strategy is then illustrated.

Regarding the definition of the optimization instance in OPC, conventionally, MBOPC solvers fracture the mask pattern edges into segments and optimize their movement. One key difference between conventional MBOPC solvers and ours is the moving direction of the optimization instances, i.e., segments or control points. Most existing MBOPC solvers only conduct unidirectional movement inwards or outwards of the pattern. In BEAM, we propose to optimize the control point movements along two basis directions for more versatile correction. $\boldsymbol{\Delta P}$ defines the movement of control points bi-directionally, in the form of:

$$\boldsymbol{\Delta P} = [\boldsymbol{\Delta p}_x, \boldsymbol{\Delta p}_y], \quad \boldsymbol{\Delta P} \in \mathbb{R}^{2\times N}, \tag{2}$$

where $\boldsymbol{\Delta p}_x$ and $\boldsymbol{\Delta p}_y$ refer to the horizontal and vertical movement of all control points, respectively.

With the well-defined optimization instance $\boldsymbol{\Delta P}$, we then formulate the optimization objective. The ultimate objective of OPC task is to minimize the EPE. With Equation (1), the EPE change on the wafer image in response to any mask modification $\boldsymbol{\Delta P}$ can be linearly approximated. If we regard the mapping from the movements on N control points to the resulting EPE change on K measuring sites as a multiple-in-multiple-out system, this relationship can be characterized by a tensor $\mathcal{J} \in \mathbb{R}^{2\times K\times N}$, where the two channels $\boldsymbol{J}_x = \mathcal{J}_{0,:,:}$ and $\boldsymbol{J}_y = \mathcal{J}_{1,:,:}$ each represent a direction. Both $\boldsymbol{J}_x$ and $\boldsymbol{J}_y$ are in the form of a Jacobian matrix, where each entry reflects the sensitivity of a given measuring site's EPE to a specific control point movement in the horizontal or vertical direction. Tensor $\mathcal{J}$ serves as a linear surrogate that encodes how localized mask modifications influence proximity effects across the layout. With the formulation of $\boldsymbol{\Delta P}$ and $\mathcal{J}$, minimizing the overall EPE can be expressed as seeking a control point movement vector $\boldsymbol{\Delta P}$ that reduces the predicted EPE as close to zero as possible. The optimization objective finally becomes:

$$\min_{\boldsymbol{\Delta P}} \|\boldsymbol{J}_x\boldsymbol{\Delta p}_x + \boldsymbol{J}_y\boldsymbol{\Delta p}_y + f(M)\|_2^2, \tag{3}$$

where M is the current mask. The tensor $\mathcal{J}$ is formulated as:

$$\mathcal{J}_{d,:,:} = \begin{bmatrix} \frac{\partial e_{1,d}}{\partial p_{1,d}} & \frac{\partial e_{1,d}}{\partial p_{2,d}} & \cdots & \frac{\partial e_{1,d}}{\partial p_{N,d}} \\ \frac{\partial e_{2,d}}{\partial p_{1,d}} & \frac{\partial e_{2,d}}{\partial p_{2,d}} & \cdots & \frac{\partial e_{2,d}}{\partial p_{N,d}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{K,d}}{\partial p_{1,d}} & \frac{\partial e_{K,d}}{\partial p_{2,d}} & \cdots & \frac{\partial e_{K,d}}{\partial p_{N,d}} \end{bmatrix}, \tag{4}$$
$$d = 0, 1, \quad \mathcal{J} \in \mathbb{R}^{2\times K\times N},$$

where d indicates the horizontal or vertical direction, and $\frac{\partial e_{i,d}}{\partial m_{j,d}}$ means the rate of change in the estimated EPE at the measuring site e_i triggered by the movement of control point p_j on direction d. With this formulation, each entry in $\mathcal{J}$ quantifies the EPE sensitivity, i.e., how the EPE at a site changes in response to a small movement of a control point. While this provides the general structure of $\mathcal{J}$, in this section, we focus on outlining the optimization workflow. The detailed construction of $\mathcal{J}$ is discussed in the next section.

Once $\mathcal{J}$ is established, the next step is to minimize the EPE, which involves solving Equation (3). A straightforward approach would be to set the expected EPE to zero and solve the equation using the inverse matrices of $\boldsymbol{J}_x$ and $\boldsymbol{J}_y$. However, in the OPC context, local mask modifications only affect EPEs within a limited neighborhood due to the proximity effect, making the tensor $\mathcal{J}$ generally very sparse. As a result, directly solving with inverse matrices can lead

Algorithm 1: BEAM Workflow

```
   Input: Set of control points 𝒫 = {p_i}_{i=1}^N, maximum OPC
          iterations T, sampling batch size n, number of sampled
          batches k
   Output: Optimized mask M*
1  M_0 ← MaskInit(𝒫) by Equation (2)
2  M* ← M_0
3  EPE_best ← LithoSim(M_0)
4  for t ← 0 to T do
5      J ← CalcTensor(M_t, 𝒫, n, k)
6      ΔP ← GradientDescent(J) by Equation (5)
7      M_{t+1} ← Apply ΔP to M_t
8      EPE_tmp ← LithoSim(M_{t+1})
9      if EPE_tmp < EPE_best then
10         M* ← M_{t+1}
11         EPE_best ← EPE_tmp
12 return M*
```

Algorithm 2: Sensitivity Tensor Calculation

```
Input: Input mask M, set of control points 𝒫 = {p_i}_{i=1}^N,
       sampling batch size n, number of sampled batches k
Output: 𝒥: Sensitivity tensor
function CalcTensor(M, 𝒫, n, k):
    EPE_M ← LithoSim(M)
    Randomly sample k batches {ℬ_i}_{i=1}^k from 𝒫
    for i ← 1 to k do
        for d ← 0 to 1 do
            M_tmp ← Perturb(ℬ_i, Δp_d)
            EPE_tmp ← LithoSim(M_tmp)
            Δe ← EPE_tmp − EPE_M
            w ← CalcWeight(Δe), by Equation (6)
            J_d^(sub) ← GetSubMat(w, Δe), by Equation (8)
    J_x ← Sum(J_0^(sub))
    J_y ← Sum(J_1^(sub)), by Equation (9)
    𝒥_{0,:,:}, 𝒥_{1,:,:} ← J_x, J_y
```

to an ill-posed solution. To address this, we apply a gradient descent step to iteratively solve the objective. The gradient descent step is mathematically formulated as:

$$\text{grad} = \begin{bmatrix} \boldsymbol{J}_x^T \cdot (\boldsymbol{J}_x \boldsymbol{\Delta P_x} + \boldsymbol{J}_y \boldsymbol{\Delta P_y} + f(M)) \\ \boldsymbol{J}_y^T \cdot (\boldsymbol{J}_x \boldsymbol{\Delta P_x} + \boldsymbol{J}_y \boldsymbol{\Delta P_y} + f(M)) \end{bmatrix}, \qquad (5)$$
$$\boldsymbol{\Delta P} = \boldsymbol{\Delta P} - \alpha \cdot \text{grad},$$

where α is the learning rate. The other OPC iteration launches until the quality of the updated mask reaches a satisfying result.

Algorithm 1 demonstrates an overview of the optimization framework. The optimization process launches after initialization, and each iteration starts with calculating tensor $\mathcal{J}$, (line 5). Using this, a gradient descent step is performed to compute the mask adjustment ΔP(line 6). The updated mask M_{t+1} is then generated by applying ΔP to the current mask M (line 7), with which the next iteration is launched. This process continues for multiple iterations, and the algorithm eventually returns the optimized mask M^* with the best EPE.

D. EPE Sensitivity Estimation

As previously discussed, each iteration of the correction process begins with the construction of the tensor $\mathcal{J}$, which captures the EPE sensitivity by observing how perturbations to control points affect the wafer's

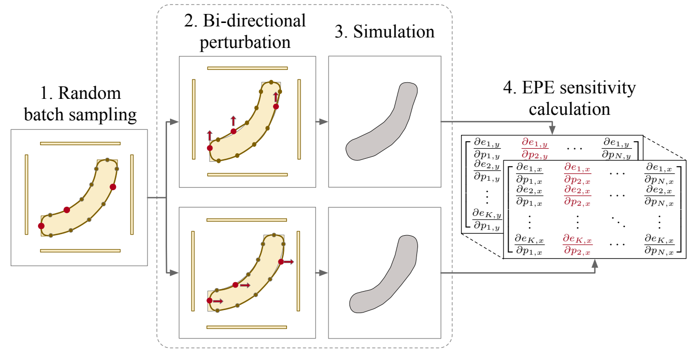

Fig. 5 The EPE sensitivity modeling process. Several subsets of control points are randomly sampled and perturbed in two orthogonal directions. After the simulation, the EPE change on the wafer image is captured for the EPE sensitivity estimation.

EPE. To construct $\mathcal{J}$, the process involves four key steps: applying small perturbations to selected control points, conducting lithography simulations to observe the resulting EPE changes, normalizing by the perturbation amplitudes to capture the rate of change, and updating the corresponding columns of $\mathcal{J}$. However, several challenges arise in this process. First, determining the exact perturbation to apply is non-trivial. Second, balancing the informativeness of the EPE change vector with computational cost is a challenge. While involving more control points per perturbation reduces the specificity of the EPE change vector for individual points, it also reduces the number of simulations required. Conversely, fewer control points require more simulations to sufficiently capture the impact of each control point. Lastly, measuring the individual contribution of each control point from a single EPE change vector becomes complicated when multiple points are involved. In solving these problems, we propose a bidirectional perturbation strategy, a batch-sampling-based method for modeling $\mathcal{J}$, and a distance-based contribution decomposition approach, balancing between high performance and efficiency. The following section explains the techniques we propose, respectively, and the workflow of constructing $\mathcal{J}$ is demonstrated in Algorithm 2.

Bidirectional Perturbation for Curvilinear Mask. Recall that in BEAM, control points are optimized to move bi-directionally in horizontal and vertical directions. To support this flexibility in movement, the EPE sensitivities must be modeled in both directions. Therefore, we adopt a bi-directional perturbation strategy in constructing $\mathcal{J}$. For each control point p_i, we apply two separate perturbations $\epsilon_{i,x}$ and $\epsilon_{i,y}$. Perturbations are applied independently in the two directions, resulting in two EPE change vectors $\boldsymbol{\Delta e}_x \in \mathbb{R}^K$ and $\boldsymbol{\Delta e}_y \in \mathbb{R}^K$, both include the EPE change values calculated across all K measuring sites. They are then separately used for updating the two channels of $\mathcal{J}$, and specifically, $\boldsymbol{J}_x$ and $\boldsymbol{J}_y$.

Batch Sampling-based Sensitivity Analysis. A brute-force way of modeling $\mathcal{J}$ involves perturbing one control point at a time and observing the EPE change. Specifically, in calculating $\mathcal{J}_{:,:,i}$, perturbations are applied separately in the horizontal and vertical directions to control point p_i, followed by lithography simulation to evaluate the EPE change at all measuring sites. This process is repeated for all N control points, requiring $2N$ separate lithography simulations per iteration. While effective, this method is computationally expensive.

To address this, we propose a batch-sampling-based method for EPE sensitivity estimation. Fig. 5 demonstrates an overview of this strategy. Instead of perturbing control points one by one, we randomly sample k batches of control points from the control points set $\mathcal{P}$, denoted by $\mathcal{B}_i$, where $i = 1, ..., k$. Each batch contains n points. For each batch, the sampled control points are perturbed separately in the two directions.

The resulting temporary mask is then input into the lithography simulator to assess the impact of the perturbations. The sample-perturb-simulate cycle is performed k times, allowing parallel processing. After simulation, $2k$ EPE change vectors can be captured for sensitivity estimation. Conceptually similar to Monte Carlo sampling, this batch-sampling method approximates an unknown distribution by randomly sampling subsets of control points, significantly reducing the number of simulations and speeding up the construction of tensor $\mathcal{J}$.

Distance-based Contribution Decomposition From the batch-sampling step, $2k$ EPE change vectors can be collected for tensor $\mathcal{J}$ estimation. Since every vector integrates the impact of perturbing a batch of control points, it is necessary to decompose each vector into the individual contributions from the participating control points. Since the decomposition strategy is identical for all vectors to both channels of $\mathcal{J}$, we focus on decomposing one vector to one channel for simplicity.

Let $\boldsymbol{\Delta e} \in \mathbb{R}^K$ represent the EPE change vector, and $\boldsymbol{J}$ denote the target channel of $\mathcal{J}$. If horizontal perturbations are applied, the target channel is $\mathcal{J}_{0,:,:}$, and otherwise $\mathcal{J}_{1,:,:}$. The corresponding perturbed control point batch is denoted by B. Hence, the j^{th} entry of $\boldsymbol{\Delta e}$, denoted by Δe_j, represents the EPE change at the j^{th} measuring site caused by perturbing control points in $\mathcal{B}$ with perturbation vector $\boldsymbol{\Delta p}$.

For decomposition, we first regard Δe_j as the sum of the impacts made by the movement of each control point, and use w_{ij} to quantify the contribution percentage of p_i's perturbation to the measuring site e_j. To decide the individual weight properly, we recall the background of OPC. The wafer image distortions arise primarily due to the proximity effect when light passes through the mask features. Intuitively, the influence of a local mask modification on the wafer image attenuates with increasing distance between the modification and the measurement site. Therefore, we assume that the contribution of the control point p_i on the measure point e_j is inversely related to the Euclidean distance in between. Based on this assumption, the weight w_{ij} is formulated as:

$$w_{ij} = \frac{\exp(-dist_{ij})}{\sum_{k=1}^{n} \exp(-dist_{kj})} \quad \forall i = 1, 2, \cdots, n, \tag{6}$$

where $dist_{ij}$ is the Euclidean distance between p_i and e_j. This formulation ensures that the contribution of each control point is weighted by its proximity to the measuring site, with closer control points having a greater impact on the EPE at the site.

Once the decomposition weights w_{ij} are established, we commit the decomposed influence to $\boldsymbol{J}$. Since both EPE values and perturbations are discrete variables in the context, instead of computing partial derivatives, we estimate the sensitivity using the rate of EPE change to the applied perturbation, formulated as:

$$\frac{w_{ij}\Delta e_j}{\Delta p_i} = \text{Sensitivity of } e_j \text{ to the movement of } p_i. \tag{7}$$

Meanwhile, as only a subset of control points is sampled and perturbed each time, only the columns corresponding to the sampled control points will be updated in $\boldsymbol{J}$. Hence, one-time sampling only updates n columns of $\boldsymbol{J}$. This can be represented by a sub-matrix, $\boldsymbol{J}^{(sub)}$. The mathematical formulations are as follows:

$$J_{ij}^{(sub)} = \begin{cases} w_{ij} \cdot \frac{\Delta e_j}{\Delta p_i}, & \text{if } p_i \in \mathcal{B} \\ 0, & \text{otherwise,} \end{cases} \quad \boldsymbol{J}^{(sub)} \in \mathbb{R}^{K \times N}. \tag{8}$$

Summing all the sub-matrices from k batches results in $\boldsymbol{J}$:

$$\boldsymbol{J} = \sum_{i=1}^{k} \boldsymbol{J}^{(sub)}. \tag{9}$$

To avoid the accumulation of overlapping points, when adding up all the sub-matrices, each column in $\boldsymbol{J}$ will be normalized with the number of times the corresponding control point is sampled. Calculating $\boldsymbol{J}$ on both horizontal and vertical directions accomplishes the two channels of $\mathcal{J}$.

IV. Experimental Results

A. Experimental Setup

Benchmark. In our experiments, a set of fundamental components commonly encountered in photonic designs is utilized as the benchmark for validating the proposed approach. The designs are partly visualized in Fig. 6, including circular, Euler and Bezier bends, splitters, spirals, multiple kinds of couplers and rings, generated by GDSFactory [22].

Implementation Details. The proposed method is implemented in Python on a CentOS-7 machine with an Intel i7-5930K 3.50GHz CPU and Nvidia GeForce RTX 3090 GPU. The lithography model is from the ICCAD2013 contest [23], and we employ the simulator implementation from an open-source project named OpenILT [24]. As a proof-of-concept, we scale the designs to accommodate the input scale requirement of the simulator, and rescale the wafer images back to the original size. Both the correction process and image evaluation are performed at the original scale.

In our BEAM implementation, unless otherwise specified, five batches of control points are sampled in each iteration, and the sampling batch size is 25% of the total number of control points. The gradient descent step's learning rate α is 0.01. The results on all testcases are visualized in Fig. 6, including the photomasks, the wafer images under nominal conditions, and the PV band patterns when the exposure dose range is restricted to $1.00 \pm 2\%$

B. Comparison with Baselines

To validate the effectiveness of BEAM, we conduct experiments that compare BEAM against two ILT methods, MOSAIC [4] and Multi-level ILT [21], and the adaptation of MB-OPC solutions to curvilinear designs, revealed in TABLE II. The ILT methods are implemented by the open-source project OpenILT [24]. Besides, two MB-OPC solutions are compared. The first strategy corresponds to "Manhattanized OPC" in TABLE II. The Manhattanization step is applied to the initial masks before the optimization process, and the resulting rectilinear patterns are split into line segments. A local EPE correction heuristic is implemented in this method, which moves the local segments inwards when fixing an outer EPE, and vice versa. Another compared baseline is referred to as "Rigorous BEAM" in TABLE II. It adopts the same control point placement, movement, and curvilinear mask formulation strategy as ours. The only difference is that it rigorously constructs the EPE sensitivity matrix by enumerating all control points and perturbing one point at a time, followed by simulation instead of batch sampling. All methods adopt an early-stopping strategy, where the optimization ends when the average EPE per measuring site is no more than 15nm.

We compare BEAM against the baselines in terms of mean EPE (denoted "MEPE" in TABLE II), L2 loss between the target pattern and the wafer image, PV-band, and runtime. Mean EPE refers to the average EPE per measurement point. In comparison, since Manhattanized OPC failed to exceed the EPE threshold for early stopping in most cases, the time when it first reaches the best EPE is recorded as its runtime. BEAM achieves the best EPE values and the fastest runtime on most testcases. Specifically, it reduces the EPE by 11% and L2 loss by 5% than MOSAIC, and outperforms Multi-level ILT in EPE and L2 loss by 13% and 26%. Competitive performance is reached in the PV band comparison. Moreover, BEAM notably outperforms two ILT methods in efficiency by 60% and 17%. However, Manhattanized OPC produces a sub-optimal result, as rectilinear masks

TABLE II Performance comparison against baselines. Mean EPE per measure point, PV band, and runtime are compared and measured in nanometers (nm), square nanometers (nm^2), and seconds (s), respectively. Testcases 1 to 4 correspond to Fig. 6(a) from left to right.

Design	# MP†	MOSAIC [4]				Multi-level ILT [21]				Manhattanized OPC				Rigorous BEAM				BEAM			
		MEPE	L2($\times 10^3$)	PVB($\times 10^3$)	RT	MEPE	L2($\times 10^3$)	PVB($\times 10^3$)	RT	MEPE	L2($\times 10^3$)	PVB($\times 10^3$)	RT	MEPE	L2($\times 10^3$)	PVB($\times 10^3$)	RT	MEPE	L2($\times 10^3$)	PVB($\times 10^3$)	RT
Case 1	196	15	525	1213	2.31	15	574	1266	2.33	22	753	1201	2.67	12	545	1237	32.45	15	615	1209	2.19
Case 2	126	15	285	1006	6.52	15	370	982	2.64	22	474	1126	2.78	14	341	1022	28.95	15	338	1006	4.6
Case 3	204	15	637	1827	1.83	15	680	1825	2.19	20	878	1884	4.92	15	531	1847	34.17	15	519	1842	3.77
Case 4	326	17	1124	4459	18.24	20	1366	4269	11.13	26	1547	4214	9.38	22	1402	4397	69.52	15	1018	4451	7.46
Case 5	160	15	576	1195	2.01	14	530	1171	3.48	15	514	1190	2.63	9	403	1198	5.24	7	352	1199	0.79
Case 6	102	15	262	699	2.35	14	301	736	1.38	22	442	732	2.49	12	257	687	7.25	13	288	695	0.82
Case 7	144	14	319	900	2.07	15	380	893	1.84	24	331	879	2.49	15	386	943	28.16	14	347	901	3.45
Case 8	72	15	208	574	1.91	14	242	597	1.72	19	298	576	2.47	15	235	561	7.84	13	212	578	1.07
Case 9	154	15	354	985	2.43	15	441	1042	1.91	21	542	1013	2.52	15	419	1042	17.48	15	417	1042	2.45
Case 10	106	15	272	692	2.12	15	497	719	1.28	21	414	686	2.52	13	272	670	4.88	14	282	677	0.54
Case 11	98	15	298	675	2.02	15	457	847	1.42	20	375	636	2.5	13	250	621	6.21	13	278	631	0.45
Case 12	72	14	207	562	1.81	15	239	563	2.15	18	289	595	2.61	14	214	549	4.47	11	166	562	0.95
Sum	1760	180	5067	14787	45.62	182	6077	14910	33.47	250	6857	14732	39.98	169	5255	14774	246.62	160	4832	14793	28.54
Ratio		1.11	1.05	1.00	1.60	1.13	1.26	1.01	1.17	1.56	1.42	1.00	1.40	1.06	1.09	1.00	8.64	1.00	1.00	1.00	1.00

† # MP means the number of EPE measure points.

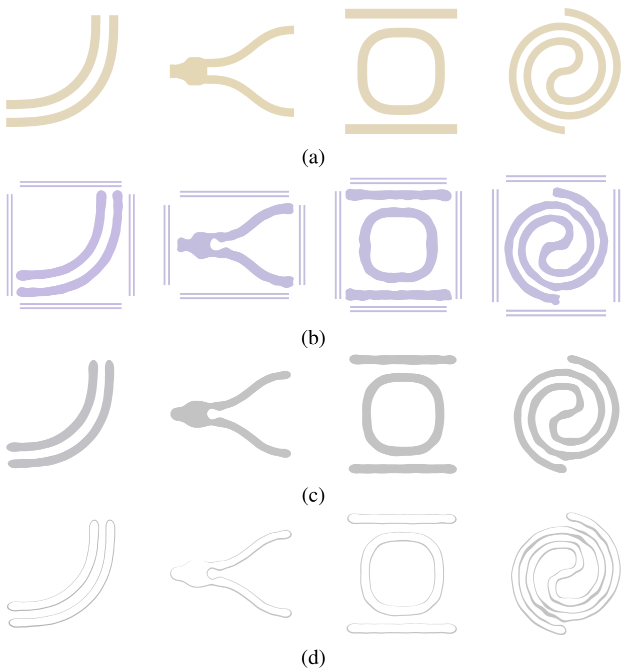

Fig. 6 Visualization of the benchmark used in the experiments, and the results generated by BEAM, including (a) the target patterns, (b) the optimized masks, (c) the wafer image in nominal condition, and (d) the PV band visualization.

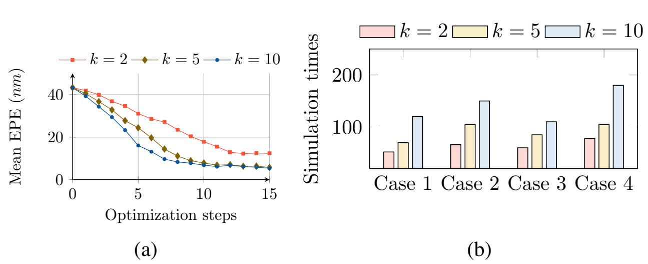

Fig. 7 (a) The EPE trajectories of Case 1 in TABLE II. k refers to the number of batches sampled in each OPC iteration. (b) The number of lithography simulation times. Testcases correspond to Case 1 to Case 4 in TABLE II.

are less effective at correcting EPEs at arbitrary angles. The time at which it first achieves its lowest EPE was recorded. Besides, Rigorous BEAM achieves parallel mask quality compared with BEAM due to the similar mask formulation strategy, but it is 8 times slower than BEAM due to the exhaustive EPE sensitivity modeling.

C. Ablation Study

To further study the trade-off between computational efficiency and mask fidelity, we conduct an ablation study toward the sampling ratio, and more specifically, the number of batches sampled k in each optimization step. We set k to 2, 5, 10 for comparison. Section IV-B shows the EPE trajectories in optimizing Case 1 with respect to optimization steps. It can be observed that for all sampling ratios, the EPE trajectories consistently decrease as the number of optimization steps increases. Ultimately, curves of $k = 5$ and $k = 10$ converge to better results than $k = 2$. This is because an increased number of batches sampled provides a more reliable estimation of EPE sensitivities due to the availability of more data. Therefore, as the sampling ratio increases, the EPE drops more significantly within fewer optimization steps, and also converges better. On the other hand, the need for data becomes saturated with $k = 5$, since the curves of $k = 5$ and $k = 10$ converge to a similar level after sufficient optimization steps.

However, increasing the number of sampled batches introduces a trade-off. While it enhances the reliability of EPE sensitivity estimation, the computational cost significantly increases revealed by the number of lithography simulations. Fig. 7(b) reveals the number of lithography simulation times needed in optimizing the testcases. The simulation cost significantly increases with a larger k. Therefore, while benefits correcting accuracy, the high demand for simulation in higher sampling ratios requires an increased computational time and cost.

V. Conclusion

In this work, we present BEAM, a unified model-based OPC framework tailored for curvilinear photonic designs. BEAM represents mask shapes using control points and B-spline interpolation that supports smooth and flexible mask boundaries. With an EPE measurement strategy introduced for curvilinear shapes, a bidirectional MEEF analysis approach is proposed, supporting control point movements along two orthogonal directions to ensure versatile corrections. Besides, we introduce a batch-based strategy to balance between correction flexibility and computational overhead. We validate BEAM on a set of fundamental curvilinear PIC components and demonstrate that it achieves outstanding correction performance with significantly reduced computational overhead.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China (No. 62204066), the Nansha District Key Area S&T Scheme (No. 2024ZD007), the Guangdong Science and Technology Department (No. 2025B1212150003), and The Research Grants Council of Hong Kong SAR (No. RFS2425-4S02).

References

[1] C. A. Mack, *Field guide to optical lithography*. SPIE press Bellingham, WA, 2006, vol. 6.

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

[3] W. Bogaerts and L. Chrostowski, "Silicon photonics circuit design: methods, tools and challenges," *Laser & Photonics Reviews*, vol. 12, no. 4, p. 1700237, 2018.

[4] J.-R. Gao, X. Xu, B. Yu, and D. Z. Pan, "MOSAIC: Mask optimizing solution with process window aware inverse correction," in *ACM/IEEE Design Automation Conference (DAC)*, 2014, pp. 52:1–52:6.

[5] 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*, vol. 39, no. 12, pp. 5069–5082, 2020.

[6] Z. Yu, G. Chen, Y. Ma, and B. Yu, "A GPU-enabled level set method for mask optimization," in *IEEE/ACM Proceedings Design, Automation and Test in Eurpoe (DATE)*, 2021, pp. 1835–1838.

[7] 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 *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2020.

[8] Y. Luo, X. Liang, and Y. Ma, "Enabling robust inverse lithography with rigorous multi-objective optimization," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2024, pp. 1–9.

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

[10] 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 (TCAD)*, vol. 35, no. 8, pp. 1345–1357, 2016.

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

[12] X. Liang, Y. Ouyang, H. Yang, B. Yu, and Y. Ma, "RL-OPC: Mask optimization with deep reinforcement learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 1, pp. 340–351, 2023.

[13] X. Liang, H. Yang, K. Liu, B. Yu, and Y. Ma, "CAMO: Correlation-aware mask optimization with modulated reinforcement learning," in *ACM/IEEE Design Automation Conference (DAC)*, 2024, pp. 1–6.

[14] L. Pang, "Inverse lithography technology: 30 years from concept to practical, full-chip reality," *Journal of Micro/Nanopatterning, Materials, and Metrology*, vol. 20, no. 3, pp. 030 901–030 901, 2021.

[15] N. Zeggaoui, B. Orlando, G. Kerrien, V. Farys, E. Yesilada, S. Cremer, A. Tritchkov, and V. Liubich, "Si-photonics waveguides manufacturability using advanced RET solutions," in *Optical Microlithography XXX*, A. Erdmann and J. Kye, Eds., vol. 10147, International Society for Optics and Photonics. SPIE, 2017, p. 1014711.

[16] J. Lei, Y. Yang, G. Lippincott, and X. Zhang, "Model-based OPC using the MEEF matrix III," in *DTCO and Computational Patterning III*, N. V. Lafferty, Ed., vol. 12954, International Society for Optics and Photonics. SPIE, 2024, p. 1295418.

[17] N. B. Cobb and Y. Granik, "Model-based opc using the meef matrix," in *22nd Annual BACUS Symposium on Photomask Technology*, vol. 4889. SPIE, 2002, pp. 1281–1292.

[18] J. Lei, L. Hong, G. Lippincott, and J. Word, "Model-based opc using the meef matrix ii," in *Optical Microlithography XXVII*, vol. 9052. SPIE, 2014, pp. 170–178.

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

[20] J. Choi, S. Ryu, S. Lee, M. Kim, J. Park, P. Buck, I. Bork, B. Durvasula, S. Gharat, N. Rao *et al.*, "Study on various curvilinear data representations and their impact on mask and wafer manufacturing," *Journal of Micro/-Nanopatterning, Materials, and Metrology*, vol. 20, no. 4, pp. 041 403–041 403, 2021.

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

[22] "The GDSFactory Project," https://gdsfactory.github.io.

[23] 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)*. IEEE, 2013, pp. 271–274.

[24] S. Zheng, B. Yu, and M. Wong, "Openilt: An open source inverse lithography technique framework," in *2023 IEEE 15th International Conference on ASIC (ASICON)*. IEEE, 2023, pp. 1–4.