

Bi-Level Optimization Accelerated DRC-Aware Physical Design Automation for Photonic Devices

Hao Chen
HKUST(GZ)

Yuzhe Ma
HKUST(GZ)

Yeyu Tong
HKUST(GZ)

Abstract—Photonic integrated circuits (PICs) design has been challenged by the complex physics behind various integrated photonic devices. Inverse design offers an effective design automation solution for obtaining high-performance and compact photonic devices using computational algorithms and electromagnetic (EM) simulations. However, the challenge lies in transforming the fabrication-infeasible device geometries obtained from computational algorithms into reliable while optimal physical design. Incorporating fabrication constraints into the optimization iterations can extend running time and lead to performance compromise. In this work, we proposed a novel DRC-aware photonic inverse design framework, leveraging the bi-level optimization to enable end-to-end gradient-based device optimization. Our method can guarantee all intermediate devices on the optimization trajectory adhere to fabrication requirements and rules. The proposed workflow eliminates the need for a binarization process and fabrication constraint adaption, thus enabling a fast and efficient search for high-performance and reliable integrated photonic devices. Experimental results demonstrate the benefits of our proposed method, including improved device performance and reduced EM simulations and running time.

I. INTRODUCTION

Silicon photonic integrated circuits (PICs) have become a critical technology in the past decades, with the production of optical transceivers in data center communications, alongside the successful prototype demonstrations in computing and sensing [1]–[3]. The benefits of silicon photonics include its compatibility with complementary metal-oxide-semiconductor (CMOS) fabrication processes, and the ability to manipulate light across multiple dimensions with a compact footprint, such as optical mode, wavelength, polarization, and amplitude [4]. While electronic design automation (EDA) tools have facilitated the scaling of electronic integrated circuits, PIC design confronts distinct challenges, including the immature design automation ecosystem and the need for a diversity of photonic elements, each with complex interactions with light. Consequently, PIC design typically involves a tailored, hands-on process that demands a thorough understanding of photonics at the component level, often leading to a time-intensive development cycle [5].

Photonic inverse design has offered an effective solution for design automation of novel integrated photonic devices [6]–[10]. By specifying desired device function and performance metrics, advanced algorithms and electromagnetic (EM) simulations can be used to search for the optimal configuration of materials and geometries inside a given blank area. The use of the adjoint method in recent years has facilitated a more efficient device optimization process [6]–[10], which is based on gradient information obtained from two simulations: the forward simulation

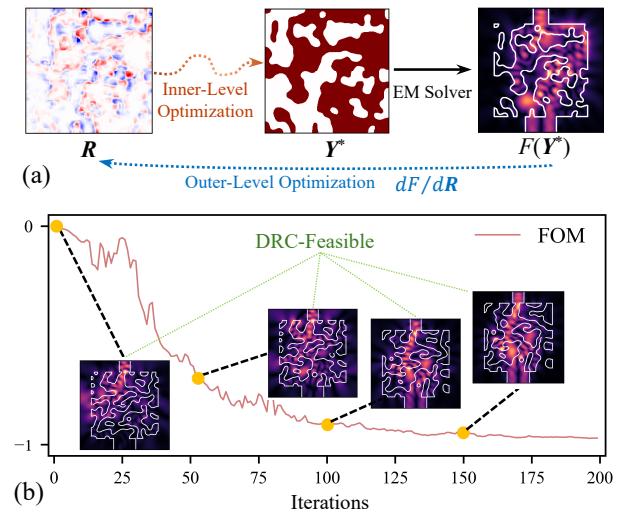


Fig. 1 (a) Photonic integrated device inverse design with the proposed bi-level optimization framework. (b) Optimization trajectory of an on-chip photonic MC device, *every* design on the optimization trajectory adheres to the DRC requirement.

predicts the behavior of light within a given device structure, while the adjoint simulation involves perturbing the EM fields to assess the impact of small changes in the device geometry on the performance metrics [9], [11]. While gradient-based inverse design algorithms can search for advanced device topology, the resulting structures often feature geometries that may not pass the design rule check (DRC) [8]. Two of the most challenging rules include minimal feature size and minimal spacing, since the optimized structures can manifest exceedingly small features and tight spacings beyond fabrication limits [7], [8], [10]. Currently, the majority of silicon photonics foundries rely on the use of 130nm to 180nm CMOS node toolset [1]. GlobalFoundries has recently offered 45nm CMOS-silicon photonics monolithic technology [12]. Fabrication beyond the lithography limits can result in pattern collapses, performance degradation, and a low fabrication yield.

To include fabrication limitations in the previous methods, unconstrained continuous optimization with binarization scheduling is employed [7], [9], [10]. A high-performance yet infeasible design is obtained first, followed by applying the fabrication constraints into the design region to guarantee its manufacturability. However, these methods generally suffer from extensive iterations. This is largely due to its continuous optimization approach, where the material density in the design region is

iteratively adjusted to converge towards an optimal solution. Secondly, another issue causing long iterations arises from the need for applying binarization after the completion of continuous optimization. Finally, a significant challenge arises from the inherent contradiction among binarization functions, fabrication constraints, and device performance. As a clear device contour is necessary for fabrication, however, the binarization function that transfers a continuous topology into a discrete and binary structure can introduce a saturation zone. Such saturation zone can hinder the device contour from evolving towards a design that is both performant and fabrication feasible. Always-feasible photonic inverse design framework has thus been proposed to utilize an always-feasible design generator [13]. However, back-propagating the adjoint gradient becomes difficult due to the non-differentiable generation process. Thus a gradient estimator is needed in the scheme. An optimization-based feasible design generator is firstly demonstrated with optimization unrolling [14]. However, this scheme is not suitable for devices with large footprints, as the memory and computational costs increase with the size of the footprint.

In this paper, we present a DRC-aware optimization framework for integrated photonic device inverse design. As shown in Fig. 1(a), our method formulates the DRC-aware photonic inverse design problem into a bi-level optimization problem, which consists of an inner-level and an outer-level optimization problem. The inner-level optimization serves as a conditional design generator, which can generate a DRC-feasible device according to a given reward matrix. The reward matrix that controls the feasible design generation is then optimized at the outer-level. We then show under the bi-level optimization framework, the gradient of the device's figure-of-merit (FOM) with respect to the reward matrix can be derived mathematically. The reward matrix is then updated using implicit differentiation. An example optimization trajectory is demonstrated in Fig. 1(b). The main contributions of this paper are summarized as follows:

- We propose a new prior condition for DRC-aware photonic devices topology generation, which can be used to implement an optimization-based conditional feasible design generator.
- We propose a bi-level optimization framework for photonic device inverse design using implicit differentiation, which ensures every device on the optimization trajectory is binary as well as adheres to the DRC requirements.
- We design and perform experiments to compare our method with the state-of-the-art photonic inverse design methods in terms of device performance and optimization efficiency. Experimental results show that our framework can reduce more than 53% required EM simulations and running time, as well as maintain better device performance.

II. PRELIMINARIES

Throughout this paper, we focus on the gradient-based photonic device design. For PIC devices, a 2D binary image \mathbf{Y} represents a design with two materials like silicon and silica. The FOM of a PIC device can be defined with its scattering coefficients (S -parameters). Physically, scattering coefficients describe the power transmission on multiple optical dimensions, such as

wavelength and optical modes. With a defined scalar FOM F , the gradient of FOM with respect to the device design $\frac{dF}{d\mathbf{Y}}$ can be obtained through two EM simulations, which is also known as the adjoint method [7]–[9]. Hence, for a specific photonic device design task, we can treat the EM simulator as a differentiable function. The simulator calculates the FOM F in the forward pass, which is $F = \text{simulator}(\mathbf{Y})$ and the gradient $\frac{dF}{d\mathbf{Y}}$ will be returned in the backward pass. With the gradient information, the photonic device design problem can be fully automated with optimization algorithms.

Photonic Device Inverse Design. Given a FOM F , the problem of photonic inverse design aims to find a physical device topology \mathbf{Y} such that the scalar FOM F is optimized. Fig. 2(a)–(d) show the EM simulator setup for the four typical on-chip photonic devices for benchmark [13], which are waveguide bend (WB), mode converter (MC), power splitter (PS) and wavelength multiplexer (WM). The FOM definition and design region size for these devices are shown in TABLE I. For WB, MC, and PS, the optimized devices are expected to operate at the wavelength $\lambda_0 = 1550\text{nm}$. The WM can direct light of $\lambda_1 = 1520\text{nm}$ and $\lambda_2 = 1580\text{nm}$ to the two output ports respectively. Under these settings, FOM for the 4 devices is normalized in the range of $[0, 1]$, where 0 indicates no power transmission (worst) and 1 indicates no power loss (best).

TABLE I Photonic device design tasks.

Device	Design region	FOM F
WB	3um \times 3um	$S_{21}(\lambda_0)$
MC	3um \times 3um	$S_{TM_1 \rightarrow TM_0}(\lambda_0)$
PS	3um \times 2.4um	$S_{21}(\lambda_0) + S_{31}(\lambda_0)$
WM	3um \times 3um	$0.5(S_{21}(\lambda_1) + S_{31}(\lambda_2))$

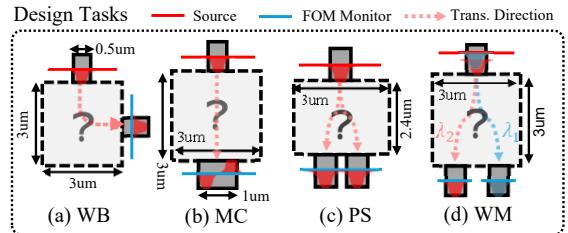


Fig. 2 Simulator setup. (a) WB, (b) MC, (c) PS, (d) WM.

DRC violations. Photonic integrated devices often exhibit a complex topology, tailored for manipulating light at micro and nano scales. Occasionally, these complex shapes exceed the limits of current fabrication capabilities. To ensure manufacturability, Design Rule Checking (DRC) is employed prior to fabrication for most PIC foundry. Typical DRC violations, illustrated in Figure 3, include issues related to feature size, spacing, sharp angles, and area.

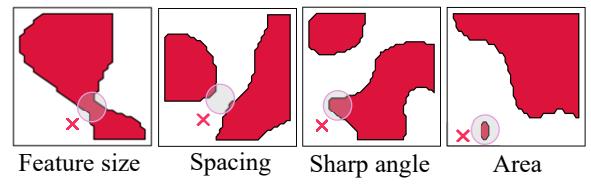


Fig. 3 Example of DRC violation in PIC device physical layout.

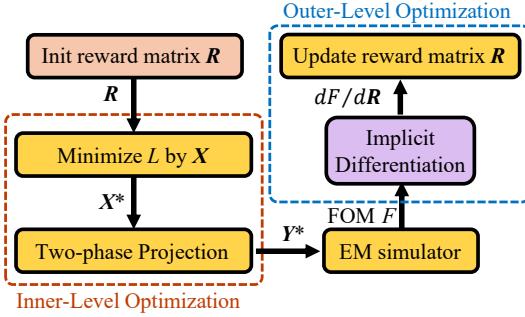


Fig. 4 The workflow of the proposed DRC-aware integrated photonic device inverse design framework.

III. PROPOSED METHOD

The overall workflow of our framework is illustrated in Fig. 4. Our framework comprises two optimization problems: an inner-level optimization problem and an outer-level optimization problem. The inner-level optimization is essentially a process for generating DRC-feasible designs. The generated feasible design, denoted as \mathbf{Y}^* , is governed by the reward matrix \mathbf{R} . Utilizing gradient information from the device simulator, implicit differentiation is employed to calculate the gradient of the device's FOM with respect to the reward matrix. Subsequently, the reward matrix is updated to enhance the device FOM.

A. DRC-Feasible Design Generation

As illustrated in Fig. 5(a), an effective strategy for achieving a design compatible with the DRC involves utilizing two circular brushes of solid and void types to paint a design, each with specified diameters d_s and d_v , respectively. This approach is grounded in two fundamental guidelines: first, ensuring the design is comprehensively filled; and second, maintaining a clear separation between solid and void elements to prevent overlap. Under these parameters, the finished design must adhere to minimum feature sizes and spacing of at least d_s and d_v , correspondingly. Additionally, the design's minimal curvature radius should not be less than the smaller value between $d_s/2$ and $d_v/2$. Here we show such a "painting" process can be equivalently converted to an optimization problem. Our method is based on the parametrization method called two-phase projection (TPP) [15], where a device design \mathbf{Y} is parameterized by a design auxiliary variable \mathbf{X} . The detailed procedure is shown in Fig. 5(b). Firstly, the design auxiliary variable \mathbf{X} is activated element-wisely by the void selectivity function $w_v(\cdot)$ and solid selectivity function $w_s(\cdot)$ to obtain \mathbf{X}_v and \mathbf{X}_s , respectively. As the name indicates, $w_v(\cdot)$ keeps void pixels (< 0), and $w_s(\cdot)$ keeps solid pixels (> 0) in design auxiliary variable \mathbf{X} . The two selectivity functions are formulated as [15]:

$$w_s(x) = \frac{1 + \alpha_s}{1 + \alpha_s e^{2n_s(1-x)}}, \quad w_v(x) = -\frac{1 + \alpha_v}{1 + \alpha_v e^{2n_v(1+x)}}, \quad (1)$$

where $\alpha_v = \alpha_s = 0.002$ and $n_s = -\ln(\alpha_s)$, $n_v = -\ln(\alpha_v)$. \mathbf{X}_v and \mathbf{X}_s are then convoluted with two 2D circular kernels $\mathbf{h}_v = \mathbf{h}(x, y; \frac{d_v}{2})$ and $\mathbf{h}_s = \mathbf{h}(x, y; \frac{d_s}{2})$ to obtain $\boldsymbol{\mu}_v$ and $\boldsymbol{\mu}_s$ respectively. $\mathbf{h}(x, y; r)$ can be formulated as:

$$\mathbf{h}(x, y; b) = \frac{1}{a} \frac{(b - \sqrt{x^2 + y^2})}{b}, \quad x^2 + y^2 \leq b^2, \quad (2)$$

where b is the radius of the kernel and a is a normalization factor such that $\sum_x \sum_y \mathbf{h}(x, y; b) = 1$. $\boldsymbol{\mu}_v$ and $\boldsymbol{\mu}_s$ are then mapped to ρ_v and ρ_s through two thresholding functions. Finally, ρ_v and ρ_s are summed together to obtain the device design \mathbf{Y} . The final device design \mathbf{Y} is bounded by $0 \leq \mathbf{Y} \leq 1$, where zero denotes the void material (silica), one denotes the solid material (silicon), and intermediate values near 0.5 denote undetermined blank regions. The whole procedure is summarized in Algorithm 1.

Algorithm 1: `twoPhaseProjection (\mathbf{X}, d_s, d_v)`

```

1  $\boldsymbol{\mu}_s = w_s(\mathbf{X}) \otimes \mathbf{h}(x, y; \frac{d_s}{2});$ 
2  $\boldsymbol{\mu}_v = w_v(\mathbf{X}) \otimes \mathbf{h}(x, y; \frac{d_v}{2});$ 
3  $\rho_s = 1 - e^{-\beta \boldsymbol{\mu}_s} + \boldsymbol{\mu}_s e^{-\beta}; // \text{ thresholding, } \beta = 8$ 
4  $\rho_v = -1 + e^{\beta \boldsymbol{\mu}_v} - \boldsymbol{\mu}_v e^{-\beta};$ 
5  $\mathbf{Y} = \frac{\rho_s + (1 + \rho_v)}{2};$ 
6 return  $\mathbf{Y};$ 
```

In the parametrization example shown in Fig. 5(b), solid and void pixels in the design auxiliary variable \mathbf{X} are situated at a distance $> (d_s + d_v)/2$. As a result, both solid and void phases in the final device design \mathbf{Y} are able to maintain the size defined by the kernel diameter d_s and d_v . However, if two pixels of different phases in the design auxiliary variable \mathbf{X} projecting different phases are spaced too closely, which is the case shown in Fig. 5(c), phase mixing in the final element density \mathbf{Y} will occur. In this case, phase clarity is lost and minimal feature size and spacing are not preserved. Noticing that when phase mixing happens, an extra blank region near the value 0.5 will form in \mathbf{Y} due to the cancellation of the solid and void phase. Hence, by optimizing \mathbf{X} such that there is no blank region left in \mathbf{Y} , phase clarity can be enforced. To penalize the blank region in \mathbf{Y} , the following function can be used:

$$g(x) = 1 - |2x - 1|. \quad (3)$$

The function graph of $g(x)$ is given in Fig. 5(c). With $g(x)$, the minimal feature size and spacing conditions in \mathbf{Y} can be satisfied if the following inequality holds:

$$c(\mathbf{Y}) = \text{mean}(g(\mathbf{Y})) \leq \epsilon, \quad (4)$$

where $\text{mean}(\cdot)$ represents the reduced mean operation and ϵ denotes a small positive number. To randomly generate a DRC-feasible design \mathbf{Y} that satisfies minimal feature size d_s and spacing d_v , we can randomly sample \mathbf{X} as the initial point to minimize $c(\mathbf{Y})$ by \mathbf{X} . Minimizing $c(\mathbf{Y})$ actually means two facts. Firstly, each pixel in the final design \mathbf{Y} should be pushed to either solid or void. Secondly, minimal feature size and spacing defined by d_s and d_v should be preserved, otherwise extra blank regions will form due to the cancellation of solid and void phases, which is just the case shown in Fig. 5(c).

B. Conditional Feasible Design Generation

To optimize a feasible photonic device, controlled feasible design generation is needed. Here we define the conditional feasible design generation loss L and show a feasible design can be generated according to a control variable by minimizing the loss:

$$L = -\text{mean}(\mathbf{R} \odot \mathbf{Y}) + \tau c(\mathbf{Y}). \quad (5)$$

In Equation (5), \odot denotes the element-wise matrix multipli-

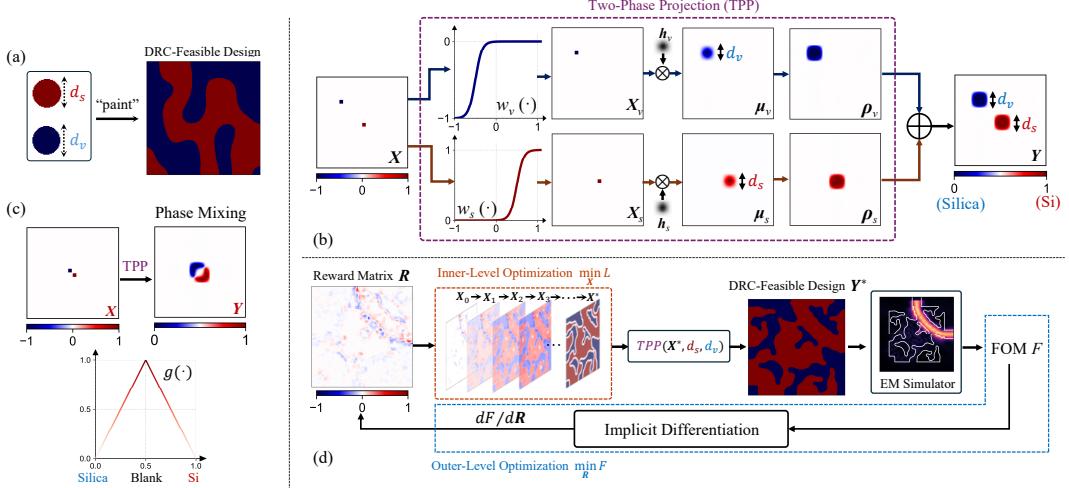


Fig. 5 (a) Demonstration of painting a DRC-feasible design with two circular brushes; (b) Two-phase projection (TPP) parameterization; (c) An example when phase mixing occurs and function graph of $g(x)$ for blank region penalization; (d) Bi-level optimization framework for DRC-aware photonic integrated device inverse design.

cation and τ is the penalty weight, \mathbf{R} is the control variable, referred to as the reward matrix. The first term of L ensures the similarity between the reward matrix \mathbf{R} and device design \mathbf{Y} and minimizing $c(\mathbf{Y})$ guarantees the design is feasible to the fabrication conditions. Thus, minimizing the loss L can generate a device design that both keeps faithful to the reward matrix \mathbf{R} and satisfy the DRC requirements. Hence, conditional feasible design generation can be equivalently converted to solving an optimization problem:

$$\mathbf{X}^* = \arg \min_{\mathbf{X}} L. \quad (6)$$

With \mathbf{X}^* , the feasible device design \mathbf{Y}^* conditioned on a given reward matrix \mathbf{R} can be obtained by:

$$\mathbf{Y}^* = \text{twoPhaseProjection}(\mathbf{X}^*, d_s, d_v). \quad (7)$$

An intuitive example of the proposed optimization-based conditional feasible design generator is shown in Fig. 5(d).

C. Gradient-based Bi-level Optimization

With the conditional feasible design generator, the feasible photonic device optimization problem can be equivalently converted to finding a reward matrix \mathbf{R} such that the FOM of the generated feasible design \mathbf{Y}^* is maximized. The optimization problem can be formed as follows:

$$\min_{\mathbf{R}} F(\mathbf{Y}^*), \quad (8)$$

$$\text{s.t. } (5), (6), (7) \quad (9)$$

In Equation (8), F is the FOM used to evaluate the device performance, which is obtained from a device simulator. Mathematically, this is called a bi-level optimization problem [16], [17], where the inner-level problem defined in Equation (6) is used for the conditional feasible design generation and the reward matrix \mathbf{R} controlling the design generation is optimized at the outer-level to improve device FOM. To enable gradient-based optimization, the gradient of F with respect to \mathbf{R} needs

to be obtained, which can be expanded by the chain rule:

$$\frac{dF}{d\mathbf{R}} = \frac{dF}{d\mathbf{Y}^*} \times \frac{d\mathbf{Y}^*}{d\mathbf{X}^*} \times \frac{d\mathbf{X}^*}{d\mathbf{R}}. \quad (10)$$

By using the first-order optimality condition and implicit function theorem [16], [18]–[20], the gradient of F with respect to \mathbf{R} can be formulated as:

$$\frac{dF}{d\mathbf{R}} = \frac{dF}{d\mathbf{Y}^*} \times \frac{d\mathbf{Y}^*}{d\mathbf{X}^*} \times \underbrace{-\left(\frac{d^2L}{d\mathbf{X}^{*2}} \right)^{-1} \times \frac{d^2L}{d\mathbf{R}d\mathbf{X}^*}}_{\text{implicit gradient}}. \quad (11)$$

Equation (11) indicates once a feasible design parameterized by \mathbf{X}^* is obtained by solving the inner-level function L to a local minimal, the gradient of the FOM F with respect to the reward matrix \mathbf{R} can be immediately obtained. Some issues need to be addressed here. Firstly, it is obvious that our inner-level objective L is a non-convex function, thus suffers from the problem called non-singleton inner-level solutions in the bi-level optimization problem [21]. The problem can be simply explained by when the reward matrix \mathbf{R} is updated according to the gradient, we may not reach the same local minimal of L in the next time. The problem can be alleviated by reducing the required iterations when solving the inner-level objective L to a local minima. In our problem, we always use $\mathbf{X}_0 = \mathbf{R}$ as the initial point when solving the inner-level objective. The second problem lies in the computation of implicit gradient in Equation (11). For a $M \times N$ design, directly computing the Hessian matrix inversion leads to the time complexity $O((MN)^3)$. Such a complexity is not acceptable even if optimizing a small device. Hence, approximation is needed when computing the implicit gradient. Our method to update the reward matrix \mathbf{R} is implicit differentiation (ID), where the inverse Hessian is approximated with the Neumann series [19]:

$$\left(\frac{d^2L}{d\mathbf{X}^{*2}} \right)^{-1} \approx \alpha \sum_{j=0}^T \left(\mathbf{I} - \alpha \frac{d^2L}{d\mathbf{X}^{*2}} \right)^j. \quad (12)$$

$dF/d\mathbf{R}$ in Equation (11) can then be obtained by using efficient

vector-Hessian products (VHP) and vector-Jacobian product (VJP) with automatic differentiation frameworks. Detailed procedures are summarized in Algorithm 2. With the Neumann series approximation, the space complexity is reduced to $O(MN)$, which can be used to deal with photonic device design with large footprints. The complete framework is illustrated in Fig. 5(d).

Algorithm 2: Implicit Differentiation (ID)

```

1 Initialize reward matrix  $\mathbf{R}$ ;
2 while not converged do
3    $\mathbf{X} = \mathbf{R}$ ;
4   for  $i = 1$  to  $T$  do
5      $\mathbf{X} = \mathbf{X} - \gamma \frac{dL}{d\mathbf{X}}$ ;
6   end
7    $\mathbf{R} = \mathbf{R} - \beta \cdot \text{gradApprox}(\mathbf{R}, \mathbf{X})$ ;
8 end
9 return  $\mathbf{R}$ ;

```

Algorithm 3: $\text{gradApprox}(\mathbf{R}, \mathbf{X})$

```

1  $\mathbf{Y}^* = \text{twoPhaseProjection}(\mathbf{X}, d_s, d_v)$ ;
2  $\mathbf{q} = \text{grad}(\text{simulator}(\mathbf{Y}^*), \mathbf{X})$ ; //  $\mathbf{q} = \frac{dF}{d\mathbf{X}}$ 
3  $\mathbf{v} = \text{grad}(f(\mathbf{X}, \mathbf{R}), \mathbf{X})$ ; //  $\mathbf{v} = \frac{dL}{d\mathbf{X}}$ 
4  $\mathbf{p} = \mathbf{q}$ ;
5 for  $j = 1$  to  $T$  do
6    $\mathbf{v} = \mathbf{v} - \alpha \cdot \text{grad}(\mathbf{v}, \mathbf{X}, \text{grad\_outputs} = \mathbf{q})$ ;
7    $\mathbf{p} = \mathbf{p} + \mathbf{v}$ ;
8 end
9  $\mathbf{p} = -\alpha * \mathbf{p}$ ; //  $\mathbf{p} \approx \frac{dF}{d\mathbf{X}^*} \times -\left(\frac{d^2 L}{d\mathbf{X}^{*2}}\right)^{-1}$ 
10  $\mathbf{p} = \text{grad}(\mathbf{v}, \mathbf{R}, \text{grad\_outputs} = \mathbf{p})$ ;
11 return  $\mathbf{p}$ ;

```

IV. EXPERIMENTAL RESULTS

The proposed framework is implemented with Pytorch [22] and TorchOpt [18] with GPU acceleration. Other algorithms for fabrication-constrained photonic device inverse design are also implemented for comparison, including geometry constraints (GC) [9], analytical constraints (AC) [7], strictly feasible design generator (SFDG) [13] and adaptive projection (AP) [10]. All the methods are evaluated by four silicon photonics device design tasks, including WB, MC, PS, and WM shown in TABLE I. 70nm, 110nm, and 150nm feature size and spacing conditions are tested in the experiment. For each fabrication condition and algorithm, 20 devices are optimized for each design task using different algorithms. We use the open-source EM simulator Meep [23] to build device simulators. 2D finite-difference time-domain (FDTD) EM simulation is used for all design tasks. We use the simulation mesh precision of 25 points per um and material mesh precision of 100 points per um in the device

simulator setup. The EM simulation is run on the CPU. All the experiments are conducted on a computer with AMD EPYC 7763 CPU and NVIDIA GeForce RTX 3090 GPU.

A. Implementation

In our approach, we adjust the penalty weight for the blank region in the inner-level objective function L to $\tau = 0.5$. To deal with the inner-level optimization, we employ the Adam optimizer, maintaining a constant learning rate of $\gamma = 0.05$ and setting $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to manage the gradient decay and second-moment exponential moving averages, respectively. The optimization process is iterated $T = 50$ times. Additionally, we incorporate $\alpha = 0.05$ into the Neumann series to facilitate an approximation of the inverse Hessian in implicit differentiation. For optimizing the outer level, we opt for the stochastic gradient descent (SGD) method. The learning rate has been meticulously optimized for various device design tasks to achieve the best outcomes. We compared our methods with four existing fabrication-constrained photonic inverse design algorithms, which are shown in Table II. There are three different methods to parameterize a design in previous works. The density pixels and binary pixels methods both use a 2D image to represent a design. The density method allows the pixel values to change continuously while only binary pixels are allowed in binary pixel methods. For the density method, generally, a binarization function (also called the projection function) is needed during the optimization to binarize the design [9], [10]. For the level-set parametrization, the device's counter is implicitly represented by the intersection between the zero-plane and the level-set function [7].

B. Algorithm Comparison

In this section, we compare different design algorithms in terms of the optimized device performance and efficiency.

Device performance: We investigate the average and best performance of the four device design tasks to evaluate the performance of our proposed algorithm. The FOM of each device is within $[0, 1]$, where 0 indicates no power transmission (worst) and 1 indicates no power loss (best). The results are revealed in TABLE III. Our method surpasses the performance of four baseline methods, both in terms of average and peak device performance. Specifically, in average device performance, it achieves improvements of 36.0%, 0.4%, 6.8%, and 21.7% over the GC, AC, SFDG, and AP methods, respectively. Similarly, peak device performance demonstrates enhancements of 5.2%, 0.1%, 1.1%, and 15.5% compared to the same methods.

Efficiency: The iteration curve examples of the MC design task using different algorithms are shown in Fig. 6(a). For GC, AC, and AP methods, ripples on the iteration curve are due to the strength adjustment of the binarization function in continuous optimization. It is worth noting that for the two methods, binary design can only be achieved near the end of optimization, potentially increasing the number of iterations required. For GC and AC methods, in the final stage of binarization, the fabrication constraints are introduced to the optimizer, where an abrupt upward trend in the convergence curve of the FOM can be observed. This phenomenon typically arises when the newly introduced constraints limit the feasible solution space, forcing the optimization algorithm to explore areas that were

TABLE II Comparison to existing methods.

Methods	Parametrization	Optimizer
GC [9]	Density pixels	MMA
AC [7]	Density pixels,Level-set	L-BFGS-B,SGD
SFDG [13]	Binary pixels	Adam
AP [10]	Density pixels	L-BFGS-B
Ours	Binary pixels	Adam,SGD

TABLE III Comparision in terms of average FOM, best FOM and required simulations (RS) to reach a successful device.

Task	GC			AC			SFDG			AP			Ours		
	Avg	Best	RS	Avg	Best	RS	Avg	Best	RS	Avg	Best	RS	Avg	Best	RS
70nm WB	0.636	0.940	153.0	0.975	0.993	177.7	0.928	0.977	52.5	0.946	0.957	159.0	0.976	0.987	39.4
70nm MC	0.932	0.974	142.4	0.981	0.995	98.3	0.963	0.998	82.1	0.944	0.963	118.4	0.982	0.998	29.0
70nm PS	0.582	0.910	160.0	0.996	0.999	99.5	0.848	0.999	82.0	0.980	0.998	110.4	0.996	0.999	41.6
70nm WM	0.777	0.925	151.0	0.957	0.977	115.7	0.957	0.978	201.8	0.905	0.926	89.5	0.962	0.982	48.6
110nm WB	0.498	0.915	134.0	0.962	0.983	183.6	0.927	0.970	73.3	0.790	0.848	-	0.970	0.991	38.0
110nm MC	0.843	0.956	137.2	0.982	0.997	200.7	0.945	0.998	90.3	0.778	0.873	-	0.983	0.995	44.3
110nm PS	0.622	0.952	148.0	0.993	0.998	101.2	0.932	0.999	75.6	0.968	0.984	111.6	0.995	0.999	48.4
110nm WM	0.848	0.913	178.7	0.953	0.980	127.2	0.911	0.974	239.2	0.718	0.729	-	0.959	0.982	98.5
150nm WB	0.730	0.995	165.7	0.966	0.996	191.3	0.932	0.968	86.6	0.573	0.658	-	0.972	0.996	64.6
150nm MC	0.931	0.968	162.4	0.967	0.996	101.2	0.900	0.997	105.3	0.529	0.667	-	0.970	0.992	91.4
150nm PS	0.540	0.959	172.5	0.990	0.998	110.6	0.856	0.998	136.9	0.967	0.971	102.4	0.990	0.999	61.3
150nm WM	0.679	0.906	172.1	0.945	0.977	137.7	0.872	0.911	354.0	0.530	0.730	-	0.958	0.978	130.0
Sum	8.618	11.313	1877	11.667	11.889	1644.7	10.971	11.767	1579.6	9.628	10.304	-	11.713	11.898	735.1

- indicates no successful device (FOM > 0.9) is obtained in all optimizations.

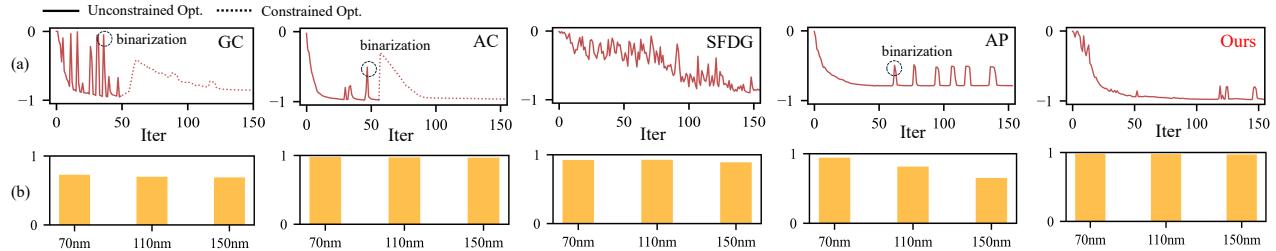


Fig. 6 (a) Convergence curves of optimizing 110nm MC device; (b) Average FOM under different DRC conditions.

previously deemed sub-optimal. The adaption of optimization constraints can greatly increase the required iteration. AC employs a two-stage approach, initially acquiring a design through continuous optimization with gradual binarization. The result of the continuous optimization phase acts as the starting point for the level-set evolution, where the constraints are subsequently introduced. Both SFDG and AP are constraint-free methods. The fabrication constraints are implicitly enforced by restricting the design space. The SFDG method employs a gradient estimator to update the design. This approach causes the iteration curve to exhibit characteristics like a noise process, potentially leading to convergence issues. The AP method simply uses the pixel size that is identical to the required feature size condition. Consequently, the possible design space is greatly reduced. This can be verified in Fig. 6(b), the average FOM achieved by the AP method drops fast when the feature size condition increases. Our method demonstrates a smoother convergence curve in comparison to other methods. Crucially, as all intermediate devices along the optimization path are binary and conform to fabrication conditions, our algorithm can be halted as soon as the target FOM is achieved. This capability significantly reduces the number of iterations required. In photonic device inverse design, the dominant computation overhead comes from the EM simulation. Hence, we compare the required simulations (RS) when a device with FOM > 0.9 can be obtained using different algorithms. For those constraint-based methods including GC and AC, we examine the FOM and constraint violation simultaneously such that the target FOM and fabrication conditions can be satisfied. For the algorithm AP, which uses density pixels with a size identical to the required minimal feature size and spacing, an extra requirement is that the degree of binarization should be larger than 99%. The required simulations under different

design tasks are also summarized in TABLE III. Among all the 12 design tasks, our method achieves the minimum RS to obtain the successful devices. Our method can reduce 60.8%, 55.3%, and 53.5% required EM simulations compared with GC, AC, and SFDG methods. In our device simulator setup, the average running time of one EM simulation is around 10.44s. Hence, our method can save running time of 992.9s, 791.3s, and 735.2s compared with GC, AC, and SFDG methods on average. We attribute the improvement to the fact that our method avoids the procedure of applying binarization and adaption to the optimization constraints, which are used in GC and AC methods. For the AP method, since the parametrization has restricted the density pixels to have the same size with the minimal feature size condition, the design freedom has been largely limited. Under the feature size conditions of 110nm and 150nm, the AP method can only successfully search for the PS device with a FOM > 0.9 while failing in the remaining design tasks.

V. CONCLUSION

In this work, we proposed a novel DRC-aware inverse design framework for fabrication-constrained photonic device design automation. Compared with the state-of-the-art methods, the experimental results show that our method can reduce more than 53% of required EM simulations, while achieving better optimization performance in benchmark device design tasks.

ACKNOWLEDGMENT

This work is supported by the Natural Science Foundation of Guangdong Province (No.2024A1515012438), Nansha District Key Area S&T Scheme (No. 2024ZD007), CCITA Funding (20240105), and Guangzhou Municipal Science and Technology Project (Guangzhou EDA Key Laboratory, No.2023A03J0013).

REFERENCES

- [1] A. Rahim, J. Goyaerts, B. Szelag, J.-M. Fedeli, P. Absil, T. Aalto, M. Harjanne, C. Littlejohns, G. Reed, G. Winzer *et al.*, “Open-access silicon photonics platforms in europe,” *IEEE Journal of Selected Topics in Quantum Electronics*, 2019.
- [2] C. Li, F. Jiang, S. Chen, X. Lil, Y. Liu, L. Chen, X. Li, and J. Xu, “Ronet: Scaling gpu system with silicon photonic chiplet,” in *2023 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2023.
- [3] F. Sunny, A. Mirza, M. Nikdast, and S. Pasricha, “Crosslight: A cross-layer optimized silicon photonic neural network accelerator,” in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, 2021.
- [4] M. J. Deen and P. K. Basu, *Silicon photonics: fundamentals and devices*. John Wiley & Sons, 2012.
- [5] S. Y. Siew, B. Li, F. Gao, H. Y. Zheng, W. Zhang, P. Guo, S. W. Xie, A. Song, B. Dong, L. W. Luo *et al.*, “Review of silicon photonics technology and platform development,” *Journal of Lightwave Technology*, 2021.
- [6] J. Skarda, L. Su, K. Y. Yang, D. Vercruyse, N. V. Sapra, and J. Vučković, “From inverse design to implementation of practical photonics,” in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019.
- [7] D. Vercruyse, N. V. Sapra, L. Su, R. Trivedi, and J. Vučković, “Analytical level set fabrication constraints for inverse design,” *Scientific reports*, 2019.
- [8] A. Y. Piggott, E. Y. Ma, L. Su, G. H. Ahn, N. V. Sapra, D. Vercruyse, A. M. Netherton, A. S. Khope, J. E. Bowers, and J. Vuckovic, “Inverse-designed photonics for semiconductor foundries,” *Acs Photonics*, 2020.
- [9] A. M. Hammond, A. Oskooi, M. Chen, Z. Lin, S. G. Johnson, and S. E. Ralph, “High-performance hybrid time/frequency-domain topology optimization for large-scale photonics inverse design,” *Optics Express*, 2022.
- [10] H. Liang, Q. Wang, X. Yuan, H. Liu, J. Xu, Y. Zhang, K. Liu, Y. Huang, and X. Ren, “Topological inverse design of fabrication-constrained nanophotonic devices via an adaptive projection method,” *Optics Letters*, 2022.
- [11] S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A. W. Rodriguez, “Inverse design in nanophotonics,” *Nature Photonics*, 2018.
- [12] T. Hirokawa, Y. Bian, K. Giewont, A. Aboketaf, S. Chandran, J.-K. Cho, Z. Chowdhury, W. S. Lee, Q. Liu, P. Sharma *et al.*, “Latest progress and challenges in 300 mm monolithic silicon photonics manufacturing,” in *Optical Fiber Communication Conference*, 2024.
- [13] M. F. Schubert, A. K. Cheung, I. A. Williamson, A. Spyra, and D. H. Alexander, “Inverse design of photonic devices with strict foundry fabrication constraints,” *ACS Photonics*, 2022.
- [14] H. Chen, M. Zhang, and Y. Tong, “Always-feasible photonic inverse design with a differentiable conditional design generator,” *ACS Photonics*, 2024.
- [15] J. V. Carstensen and J. K. Guest, “Projection-based two-phase minimum and maximum length scale control in topology optimization,” *Structural and multidisciplinary optimization*, 2018.
- [16] C. Chen, X. Chen, C. Ma, Z. Liu, and X. Liu, “Gradient-based bi-level optimization for deep learning: A survey,” *arXiv preprint arXiv:2207.11719*, 2022.
- [17] L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, and M. Pontil, “Bilevel programming for hyperparameter optimization and meta-learning,” in *International conference on machine learning*, 2018.
- [18] J. Ren, X. Feng, B. Liu, X. Pan, Y. Fu, L. Mai, and Y. Yang, “Torchopt: An efficient library for differentiable optimization,” *Journal of Machine Learning Research*, 2023.
- [19] J. Lorraine, P. Vicol, and D. Duvenaud, “Optimizing millions of hyperparameters by implicit differentiation,” in *International conference on artificial intelligence and statistics*, 2020.
- [20] A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots, “Truncated back-propagation for bilevel optimization,” in *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019.
- [21] Y. Zhang, P. Khanduri, I. Tsaknakis, Y. Yao, M. Hong, and S. Liu, “An introduction to bilevel optimization: Foundations and applications in signal processing and machine learning,” *IEEE Signal Processing Magazine*, 2024.
- [22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.
- [23] A. F. Oskooi, D. Roundy, M. Ibanescu, P. Bermel, J. D. Joannopoulos, and S. G. Johnson, “Meep: A flexible free-software package for electromagnetic simulations by the fDTD method,” *Computer Physics Communications*, 2010.