



PHOTONICS Research

Automatic synthesis of light-processing functions for programmable photonics: theory and realization

ZHENGQI GAO,^{1,*}  XIANGFENG CHEN,^{2,3} ZHENGXING ZHANG,¹ UTTARA CHAKRABORTY,¹ 
WIM BOGAERTS,^{2,3}  AND DUANE S. BONING¹

¹Microsystems Technology Laboratories, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

²Ghent University-IMEC, Department of Information Technology, Ghent, Belgium

³Center of Nano- and Biophotonics, Ghent University, Ghent, Belgium

*Corresponding author: zhengqi@mit.edu

Received 2 September 2022; revised 6 February 2023; accepted 7 February 2023; posted 9 February 2023 (Doc. ID 474606); published 30 March 2023

Linear light-processing functions (e.g., routing, splitting, filtering) are key functions requiring configuration to implement on a programmable photonic integrated circuit (PPIC). In recirculating waveguide meshes (which include loop-backs), this is usually done manually. Some previous results describe explorations to perform this task automatically, but their efficiency or applicability is still limited. In this paper, we propose an efficient method that can automatically realize configurations for many light-processing functions on a square-mesh PPIC. At its heart is an automatic differentiation subroutine built upon analytical expressions of scattering matrices that enables gradient descent optimization for functional circuit synthesis. Similar to the state-of-the-art synthesis techniques, our method can realize configurations for a wide range of light-processing functions, and multiple functions on the same PPIC simultaneously. However, we do not need to separate the functions spatially into different subdomains of the mesh, and the resulting optimum can have multiple functions using the same part of the mesh. Furthermore, compared to nongradient- or numerical differentiation-based methods, our proposed approach achieves 3× time reduction in computational cost. © 2023 Chinese Laser Press

<https://doi.org/10.1364/PRJ.474606>

1. INTRODUCTION

Photonic integrated circuits (PICs) have drawn increasing attention over the past two decades. Their primary goal is to integrate complex manipulation of light (such as routing, filtering, coupling, interfering) onto a single chip [1–3]. Today, a PIC is usually designed for one specific application, so that it can be compact and power-efficient [4]. The design methodology for these chips is similar to that of application-specific integrated circuits (ASICs) in the electronic domain, and thus this kind of PIC is usually referred to as an application-specific PIC (ASPIC).

In contrast, another mainstream type of electronic circuit is the field-programmable gate array (FPGA). These circuits are generic in concept, and their functionality is programmed by configuring the on-chip connectivity of the logical building blocks. The photonic counterpart of FPGA, the programmable photonic integrated circuit (PPIC) [4–15], has been introduced recently based on the idea of run-time manipulation of light after a chip has been fabricated. Such reconfigurability is usually

made available by controlling the active components (e.g., optical PSs [4]) with electrical/thermal signals. Due to its programmability, a PPIC is suitable for various applications such as fast prototyping of ASPICs [4], building optical neural networks (ONNs) [16], and processing quantum information [17,18].

A PPIC is composed of a mesh of tunable basic units (TBUs) [12], also called analog optical gates [4]. The most common implementation of a TBU is a 2 × 2 Mach-Zehnder interferometer (MZI) circuit [4,12]. Considering the interconnections of TBUs, PPICs can generally be classified into two categories: (i) forward-only topologies [6–11,19–21], and (ii) loop-back (recirculating) topologies [4,12–15]. In a forward-only PPIC, light propagates in one direction (e.g., from left to right). It has been proven that with particular forward-only structures, a PPIC can realize any unitary transformation [10,19,22]. When fixed-length delay lines are introduced, it is also possible to implement finite impulse response (FIR) digital filters [11]. Feed-forward PPICs are commonly used to implement ONNs for AI computing. The first notable experimental

realization of an ONN was published in 2017 [16]. Later works have considered *in situ* ONN training [20,21] via novel optical backpropagation techniques.

However, without loop-back connections, a forward-only PPIC cannot realize a ring resonator or an infinite impulse response (IIR) digital filter. Such shortcomings have motivated researchers to consider recirculating-based PPICs [15,23,24]. The most common recirculating configurations are triangular, square, or hexagonal close packing [25]. However, while these loop-back meshes offer the possibility of implementing more complex connectivities as well as FIR and IIR filters, the configuration of those functions is mostly done by manually assigning and configuring the optical gates in the mesh. Such an *ad hoc* method will not be applicable (i) when we want to synthesize several filters at the same time, and (ii) when the size of a recirculating PPIC increases substantially.

To address these issues, a few published results [13,14,26] have proposed methods to perform this task automatically. The authors in Ref. [26] proposed to use optimization techniques to synthesize optical ring resonators and MZIs on a hexagonal-mesh PPIC. In Ref. [14], the authors proposed an auto-routing method based on graph theory for a hexagonal-mesh PPIC, and multiobjective routing is demonstrated by Ref. [13]. However, these methods can be dramatically improved to overcome the following key limitations: (i) their application range is restricted and many light-processing functions are not considered; and (ii) since many optical PSs need to be optimized in a PPIC, this high-dimensional optimization problem is not efficient with current methods that rely on nongradient methods (e.g., particle swarm optimization (PSO) in Ref. [26]) or gradient methods with numerical differentiation (e.g., Eqs. (4) and (5) in the supplementary material of Ref. [26]).

In this paper, we address these two main points by relying on scattering matrix theory, together with efficient calculation of analytical gradients. Specifically, we propose an efficient method that can realize configurations for many different light-processing functions on a square-mesh PPIC, without requiring *a priori* human design guidance. We start with the compact model of a TBU and derive the analytical transfer functions of the entire circuit according to scattering matrix theory. Built upon this, we implement an automatic differentiation subroutine that can analytically calculate the mean squared error (or other cost functions) between the target frequency responses and the configured circuit responses, and the cost function derivative, with respect to all tunable parameters inside the PPIC. This enables us to efficiently perform gradient descent optimization that realizes a variety of light-processing functions with different magnitude or/and phase responses. Our work has a close relationship with Ref. [24], where the authors derive a system-level analytical scattering matrix for a hexagonal mesh. However, our approach goes beyond that work by calculating and utilizing gradients for functional synthesis. In overview, our major contributions include the following.

- In Section 2, we propose a TBU compact model appropriate for the task of optical filter synthesis, and analytically derive the TBU transfer functions using scattering matrix theory.

- In Section 3, we consider a simplified case where all the horizontal TBUs in a PPIC are fixed to bar states, from which several useful observations can be made.

- In Section 4, we demonstrate our efficient synthesis method based on automatic differentiation and gradient descent optimization. We also develop a logarithmic cost function suitable to the case when we want to optimize both the stop band and passband of a wavelength filter response.

- In Section 5, we demonstrate that our proposed method can be applied to a wide range of light-processing functions at run-time scales of minutes. We also show that our method can synthesize multiple light-processing functions simultaneously in the same waveguide mesh.

- Finally, in Section 6, we discuss the limitations of our method and future considerations, such as how to extend it to suit a PPIC containing hundreds or even thousands of TBUs with arbitrary connections.

2. THEORY OF SCATTERING MATRICES

Following Refs. [4,9,15], we consider the TBU structure as shown in Fig. 1 throughout this paper. As shown in the top row of Fig. 1, we assume that two time-harmonic optical inputs $\{a_1^{(I)}e^{j\omega t}, a_2^{(I)}e^{j\omega t}\}$ are provided, respectively, at the two left ports $\{A_1, A_2\}$. Then the outputs can be calculated based on the transfer matrix \mathbf{F} ,

$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \end{bmatrix} = \mathbf{F} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \end{bmatrix}, \quad (1)$$

where we use the superscripts “*I*” and “*O*” in parentheses to represent the direction of light going into and coming out of the TBU, respectively. The transfer matrix \mathbf{F} is given as [4,9]

$$\mathbf{F} = \underbrace{\frac{\sqrt{2}}{2} \begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix}}_{\text{right DC}} \underbrace{\begin{bmatrix} e^{-j\theta} & 0 \\ 0 & e^{-j\phi} \end{bmatrix}}_{\text{PSs}} \underbrace{\frac{\sqrt{2}}{2} \begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix}}_{\text{left DC}}, \quad (2)$$

where the optical phase shifts (PSs) are parameterized by θ and ϕ , and the DCs are fixed 50%:50% splitters. Here we emphasize that the signals $\{a_1^{(I)}, a_2^{(I)}, b_1^{(O)}, b_2^{(O)}\}$ are all complex scalar variables. In our notation, we choose $e^{j\omega t}$ dependence instead of $e^{-j\omega t}$, so that it is consistent with the conventional Fourier transform from the time domain to the frequency domain. This results in the minus signs ahead of the complex unit j

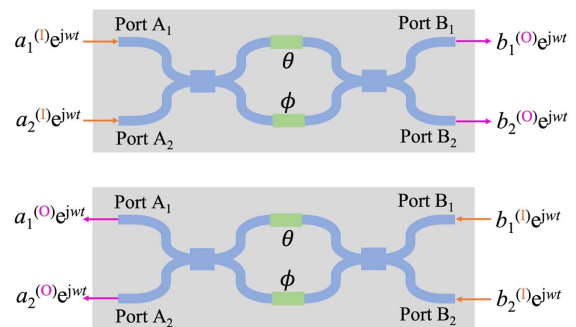


Fig. 1. Simplified schematic of a TBU. It is made up of two 50%:50% DCs on the left and right, and two optical PSs parameterized by $\{\theta, \phi\}$ in the middle. $\{\theta, \phi\}$ can be adjusted freely in $[0, 2\pi)$ by thermo- or electro-optic control of the two PSs.

on the right-hand side of Eq. (2); however, from the calculation perspective, the alternative representation can be equivalently employed.

If we reverse the direction of light propagation, as shown in the bottom row of Fig. 1, then the vector on the left-hand side of Eq. (1) will be $[a_1^{(O)}, a_2^{(O)}]^T$, while $[b_1^{(I)}, b_2^{(I)}]^T$ will be on the right. Combining the two propagation cases together, we have the scattering matrix relation,

$$\begin{bmatrix} b_1^{(O)} \\ b_2^{(O)} \\ a_1^{(O)} \\ a_2^{(O)} \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \begin{bmatrix} a_1^{(I)} \\ a_2^{(I)} \\ b_1^{(I)} \\ b_2^{(I)} \end{bmatrix}. \quad (3)$$

For our filter synthesis application, the model in Eq. (2) is insufficient: we will never obtain a frequency-dependent response using this model, because \mathbf{F} does not rely on the light frequency ω . To remedy this, we modify the previous transfer matrix by taking the role of the TBU waveguides into consideration,

$$\mathbf{F} = 0.5 \underbrace{\begin{bmatrix} e^{-j\theta} - e^{-j\phi} & -je^{-j\theta} - je^{-j\phi} \\ -je^{-j\theta} - je^{-j\phi} & -e^{-j\theta} + e^{-j\phi} \end{bmatrix}}_{\text{Eq. (2)}} \alpha e^{-j\omega \frac{n_{\text{eff}} L}{c}}, \quad (4)$$

where $n_{\text{eff}}(\omega)$ is the effective index of the propagating mode, L represents the length of the waveguide in the TBU, c is the speed of light in free space, and α represents the transmission loss introduced by the waveguides and couplers in the TBU. At the device level, there might be several waveguides in one TBU (e.g., between the left DC and the PSs, between the PSs and the right DC). Our compact model in Eq. (4) is valid as long as the waveguides are balanced in the upper and lower arms. See Appendix A for more details. Moreover, without considering dispersion (i.e., n_{eff} is a constant independent of ω), the $e^{-j\omega \frac{n_{\text{eff}} L}{c}}$ factor naturally corresponds to a time delay $\frac{n_{\text{eff}} L}{c}$ according to the Fourier transform, and thus PPICs can rely on digital filter theory to realize optical filter functions.

The circuit schematic of the recirculating PPIC waveguide mesh is shown in Fig. 2. In this paper, we ignore the TBUs in the right-most column. We also assume that the top and

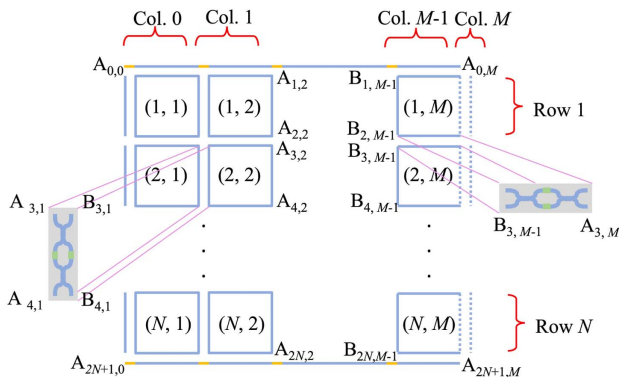


Fig. 2. Schematic of an $N \times M$ square-mesh PPIC. For derivation simplicity, we disable the TBUs at the right-most column marked by dashed lines and assume that the top and bottom connections (yellow lines) are ideal.

bottom connections (the yellow lines) are ideal connections, i.e., their transfer function is identity. These two assumptions are made for mathematical simplicity and the purpose of demonstration; note that our method is applicable without these assumptions.

Next, we introduce naming conventions for the ports and propagation directions. As shown in Fig. 3, we adopt the following conventions for the ports in this PPIC: (i) the letters “A” and “B” are used to denote the ports on the left and right edge of a vertical TBU, respectively; (ii) the subscript (n, m) is used to express that the port is on the n th row and m th column, where $n = 0, 1, \dots, 2N + 1$ and $m = 0, 1, \dots, M$. As shown in Fig. 3, for any port, the light can propagate in two directions. We define going into and out of the vertical TBU device as “I” (i.e., orange arrows) and “O” (i.e., purple arrows), respectively. One minor subtlety arises when applying this direction naming convention to the top line shown in Fig. 2, since this top line does not associate with any vertical device. In this case, we consider there to be virtual vertical TBUs above this top line, and then apply our direction naming convention. Similarly, we consider there to be virtual vertical TBUs beneath the bottom line in Fig. 2 for the purpose of notation consistency.

Following Eq. (3) and applying the scattering matrix to the two propagating directions shown in the right figure in Fig. 3, we have

$$\begin{bmatrix} a_{2i,j}^{(O)} \\ b_{2i,j}^{(O)} \\ a_{2i-1,j}^{(O)} \\ b_{2i-1,j}^{(O)} \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \begin{bmatrix} a_{2i-1,j}^{(I)} \\ b_{2i-1,j}^{(I)} \\ a_{2i,j}^{(I)} \\ b_{2i,j}^{(I)} \end{bmatrix}. \quad (5)$$

If we expand all terms in Eq. (5) and rearrange the terms to locate those related to b and a at the left-hand and right-hand side, respectively, we obtain

$$\begin{bmatrix} b_{2i-1,j}^{(I)} \\ b_{2i-1,j}^{(O)} \\ b_{2i,j}^{(I)} \\ b_{2i,j}^{(O)} \end{bmatrix} = \mathbf{V} \begin{bmatrix} a_{2i-1,j}^{(I)} \\ a_{2i-1,j}^{(O)} \\ a_{2i,j}^{(I)} \\ a_{2i,j}^{(O)} \end{bmatrix}, \quad (6)$$

where \mathbf{V} is of size 4×4 , and with some algebra, we have

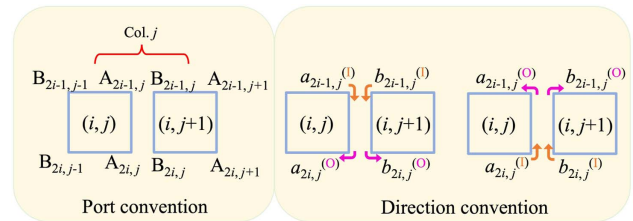


Fig. 3. Naming conventions for port and direction. Capitalized “A” and “B” should be regarded as port names, and lowercases “a” and “b” are the complex magnitudes ahead of $e^{j\omega t}$. For conciseness, we have omitted the $e^{j\omega t}$ dependence.

$$\begin{aligned}
V_{11} &= V_{33} = -F_{11}/F_{12}, \\
V_{22} &= V_{44} = F_{22}/F_{12}, \\
V_{41} &= V_{32} = 1/F_{12}, \\
V_{23} &= V_{41} = F_{21} - F_{11}F_{22}/F_{12},
\end{aligned} \quad (7)$$

and other entries of \mathbf{V} are all zero. Here we use F_{kl} to denote the entry on the k th ($k = 1, 2$) row and l th ($l = 1, 2$) column of matrix \mathbf{F} . Similar notations are applied to \mathbf{V} as well as all later occurring matrices. It is important to note that Eq. (6) holds for the index $i = 1, 2, \dots, N$ and $j = 0, 2, \dots, M - 1$, which covers all vertical TBUs in the middle, except the top and bottom lines in Fig. 2. The top and bottom lines correspond to row index 0 and $2N + 1$ under our naming convention, and the following relations hold on these two lines because we assume the yellow lines in Fig. 2 are ideal,

$$\begin{bmatrix} b_{k,j}^{(I)} \\ b_{k,j}^{(O)} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_{k,j}^{(I)} \\ a_{k,j}^{(O)} \end{bmatrix}, \quad k = 0 \quad \text{or} \quad 2N + 1. \quad (8)$$

Recall that in writing Eq. (5), we apply the scattering matrix method to the two propagation directions of a vertical TBU. We can do the same thing for a horizontal TBU, which gives the following equation:

$$\begin{bmatrix} a_{2i,j+1}^{(I)} \\ a_{2i+1,j+1}^{(I)} \\ b_{2i,j}^{(I)} \\ b_{2i+1,j}^{(I)} \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \begin{bmatrix} b_{2i,j}^{(O)} \\ b_{2i+1,j}^{(O)} \\ a_{2i,j+1}^{(O)} \\ a_{2i+1,j+1}^{(O)} \end{bmatrix}. \quad (9)$$

Similarly, we now move all terms related to b and a to the right-hand and left-hand side, respectively, and obtain

$$\begin{bmatrix} a_{2i,j+1}^{(I)} \\ a_{2i,j+1}^{(O)} \\ a_{2i+1,j+1}^{(I)} \\ a_{2i+1,j+1}^{(O)} \end{bmatrix} = \mathbf{H} \begin{bmatrix} b_{2i,j}^{(I)} \\ b_{2i,j}^{(O)} \\ b_{2i+1,j}^{(I)} \\ b_{2i+1,j}^{(O)} \end{bmatrix}, \quad (10)$$

where \mathbf{H} is of size 4×4 , and with some algebra, we have

$$\begin{aligned}
H_{12} &= F_{11}, & H_{14} &= F_{12}, \\
H_{32} &= F_{21}, & H_{34} &= F_{22}, \\
H_{21} &= F_{22}/\det(\mathbf{F}), & H_{23} &= -F_{12}/\det(\mathbf{F}), \\
H_{41} &= -F_{21}/\det(\mathbf{F}), & H_{43} &= F_{11}/\det(\mathbf{F}),
\end{aligned} \quad (11)$$

and all other entries of \mathbf{H} are zero. Here $\det(\mathbf{F}) = F_{11}F_{22} - F_{12}F_{21}$ represents the determinant of \mathbf{F} . Again note that Eq. (10) holds for the index $i = 0, 2, \dots, N$ and $j = 0, 2, \dots, M - 1$.

For a specific column index j , if we vary the row index i and k in Eqs. (6) and (8), and next stack all the resulting equations in one column, we obtain a scattering matrix for the mapping: $\{a_{n,j}^{(I)}, a_{n,j}^{(O)}\} \rightarrow \{b_{n,j}^{(I)}, b_{n,j}^{(O)}\}$, where $n = 0, 1, \dots, 2N + 1$. Similarly, if we vary the row index i in Eq. (10), we can write

down the scattering matrix for the mapping: $\{b_{n,j}^{(I)}, b_{n,j}^{(O)}\} \rightarrow \{a_{n,j+1}^{(I)}, a_{n,j+1}^{(O)}\}$. Combining these two steps gives us the scattering matrix representing the mapping: $\{a_{n,j}^{(I)}, a_{n,j}^{(O)}\} \rightarrow \{a_{n,j+1}^{(I)}, a_{n,j+1}^{(O)}\}$. Mathematically, that is to say,

$$\begin{bmatrix} a_{0,j+1}^{(I)} \\ a_{0,j+1}^{(O)} \\ \vdots \\ a_{2N+1,j+1}^{(I)} \\ a_{2N+1,j+1}^{(O)} \end{bmatrix} = \mathbf{T}^j \begin{bmatrix} a_{0,j}^{(I)} \\ a_{0,j}^{(O)} \\ \vdots \\ a_{2N+1,j}^{(I)} \\ a_{2N+1,j}^{(O)} \end{bmatrix}, \quad (12)$$

where \mathbf{T}^j is of size $(4N + 4) \times (4N + 4)$ and can be expressed as the product of two block diagonal matrices,

$$\mathbf{T}^j = \text{Diag}(\overbrace{\mathbf{H}, \dots, \mathbf{H}}^{N+1}) \text{Diag} \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \underbrace{\mathbf{V}, \dots, \mathbf{V}}_N, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right). \quad (13)$$

We emphasize that the first block diagonal matrix on the right-hand side of Eq. (13) is constructed via putting $(N + 1)\mathbf{H}$ matrices along the main diagonal. Here readers should be aware that the $(N + 1)\mathbf{H}$ matrices correspond to $(N + 1)$ different horizontal TBUs from top to bottom, and that to keep the notation uncluttered, we have not introduced subscripts or superscripts on \mathbf{H} to distinguish them. Each \mathbf{H} might be different. The second matrix on the right-hand side of Eq. (13) is constructed by putting a 2×2 matrix at the front and end, while the middle is filled with $N\mathbf{V}$ matrices corresponding to N different vertical TBUs from top to bottom. Similarly, each \mathbf{V} can be different.

If we repeat Eq. (12) M times for different column indices j , then we can obtain the overall scattering matrix for the mapping from $\{a_{n,0}^{(I)}, a_{n,0}^{(O)}\}$ to $\{a_{n,M}^{(I)}, a_{n,M}^{(O)}\}$,

$$\begin{bmatrix} a_{0,M}^{(I)} \\ a_{0,M}^{(O)} \\ \vdots \\ a_{2N+1,M}^{(I)} \\ a_{2N+1,M}^{(O)} \end{bmatrix} = \mathbf{T} \begin{bmatrix} a_{0,0}^{(I)} \\ a_{0,0}^{(O)} \\ \vdots \\ a_{2N+1,0}^{(I)} \\ a_{2N+1,0}^{(O)} \end{bmatrix}, \quad (14)$$

where

$$\mathbf{T} = \mathbf{T}^{M-1} \dots \mathbf{T}^1 \mathbf{T}^0. \quad (15)$$

If we rearrange the order of entries to put the values related to the direction I in the first several rows, and those related to the direction O in the last several rows, we obtain

$$\begin{bmatrix} a_{0,M}^{(I)} \\ \vdots \\ a_{2N+1,M}^{(I)} \\ a_{0,M}^{(O)} \\ \vdots \\ a_{2N+1,M}^{(O)} \end{bmatrix} = \mathbf{P}^T \mathbf{T} \mathbf{P} \begin{bmatrix} a_{0,0}^{(I)} \\ \vdots \\ a_{2N+1,0}^{(I)} \\ a_{0,0}^{(O)} \\ \vdots \\ a_{2N+1,0}^{(O)} \end{bmatrix}, \quad (16)$$

where \mathbf{P} is a row permutation matrix of size $(4N + 4) \times (4N + 4)$. Note that \mathbf{P} has a known structure, and its entries are either 0 or 1. For later simplicity, we will introduce the symbol $\mathbf{a}_M^{(I)} = [a_{0,M}^{(I)}, \dots, a_{2N+1,M}^{(I)}]^T$ and $\mathbf{a}_M^{(O)} = [a_{0,M}^{(O)}, \dots, a_{2N+1,M}^{(O)}]^T$ for the left-hand side of Eq. (16). Similar notations are also used for the right-hand side of Eq. (16), so that it can be simplified as

$$\begin{bmatrix} \mathbf{a}_M^{(I)} \\ \mathbf{a}_M^{(O)} \end{bmatrix} = \mathbf{T}^* \begin{bmatrix} \mathbf{a}_0^{(I)} \\ \mathbf{a}_0^{(O)} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{11}^* & \mathbf{T}_{12}^* \\ \mathbf{T}_{21}^* & \mathbf{T}_{22}^* \end{bmatrix} \begin{bmatrix} \mathbf{a}_0^{(I)} \\ \mathbf{a}_0^{(O)} \end{bmatrix}, \quad (17)$$

where

$$\mathbf{T}^* = \mathbf{P}^T \mathbf{T} \mathbf{P}, \quad (18)$$

and we adopt the block matrix notation in the last equality.

Thus far, we have obtained a relation between the input and the output. The ultimate scattering matrix \mathbf{T}^* is related to the individual \mathbf{V} (or \mathbf{H}) matrix of a vertical (or horizontal) TBU device via Eqs. (18), (15), and (13), in sequence. Furthermore, the relations from \mathbf{V} and \mathbf{H} matrices to the individual PSs $\{\theta, \phi\}$ are also clear via Eqs. (11), (7), and (4). Thus, we have obtained an analytical expression of \mathbf{T}^* defined by all PSs $\{\theta, \phi\}$. Although it is difficult to explicitly write down the expression for every entry in the \mathbf{T}^* matrix, we do know the sequential operations to construct it. Most importantly, all of the operations involved (e.g., matrix-vector multiplication) are differentiable, so that we can easily calculate $\frac{\partial \mathbf{T}^*}{\partial \theta}$ or $\frac{\partial \mathbf{T}^*}{\partial \phi}$ for any $\{\phi, \theta\}$ of any TBU device. As demonstrated later, this will form the basis for our synthesis method.

Without loss of generality, we assume that our desired forward light propagation is from left to right in the PPIC shown in Fig. 4. Then we can regard the forward input $\mathbf{a}_0^{(I)}$ at the left

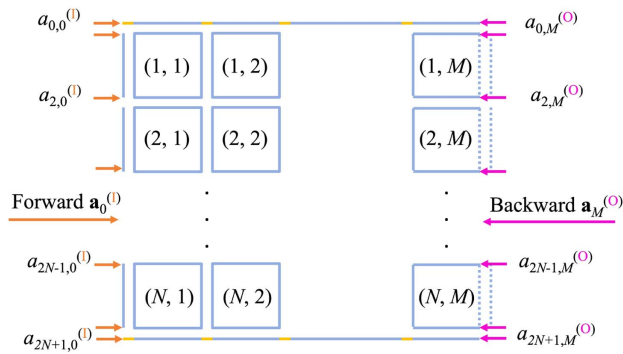


Fig. 4. Illustration of the forward input at the left and the backward input at the right. Note that the directions I and O in the parenthesized superscripts are defined according to going into or coming out of the associated vertical TBU, as defined in Fig. 3.

and the backward input $\mathbf{a}_M^{(O)}$ at the right both as given constant vectors. Based on Eq. (17), we can now express the forward output at the right $\mathbf{a}_M^{(I)}$ and the backward output at the left $\mathbf{a}_0^{(O)}$ as

$$\begin{aligned} \mathbf{a}_M^{(I)} &= (\mathbf{T}_{11}^* - \mathbf{T}_{12}^* \mathbf{T}_{22}^{*-1} \mathbf{T}_{21}^*) \mathbf{a}_0^{(I)} + \mathbf{T}_{12}^* \mathbf{T}_{22}^{*-1} \mathbf{a}_M^{(O)}, \\ \mathbf{a}_0^{(O)} &= \mathbf{T}_{22}^{*-1} \mathbf{a}_M^{(O)} - \mathbf{T}_{22}^{*-1} \mathbf{T}_{21}^* \mathbf{a}_0^{(I)}, \end{aligned} \quad (19)$$

where we use \mathbf{T}_{22}^{*-1} to represent the inverse of the matrix \mathbf{T}_{22}^* . In reality, the backward input at right $\mathbf{a}_M^{(O)}$ will usually be set to a zero vector, and the forward output at right is regarded as the final response of the PPIC,

$$\mathbf{a}_M^{(I)} = (\mathbf{T}_{11}^* - \mathbf{T}_{12}^* \mathbf{T}_{22}^{*-1} \mathbf{T}_{21}^*) \mathbf{a}_0^{(I)} = \mathbf{G}^* \mathbf{a}_0^{(I)}, \quad (20)$$

where for simplicity, we have denoted $\mathbf{G}^* = \mathbf{T}_{11}^* - \mathbf{T}_{12}^* \mathbf{T}_{22}^{*-1} \mathbf{T}_{21}^*$.

Several points are worth noting. First, both $\mathbf{a}_0^{(I)}$ and $\mathbf{a}_M^{(I)}$ are of size $(2N + 2) \times 1$. This provides us with some flexibility to synthesize multiple light-processing functions simultaneously. For instance, we can feed an input wave from the top port of the first vertical TBU, i.e., $a_{1,0}^{(I)}$ equal to 1 and all other entries of $\mathbf{a}_0^{(I)}$ equal to 0. Then the outputs at the second and third entries of $\mathbf{a}_M^{(I)}$ can be used to synthesize two different light-processing functions. Second, $\mathbf{a}_0^{(O)}$ might not be zero in Eq. (19) even if $\mathbf{a}_M^{(O)}$ is zero, because the information brought by $\mathbf{a}_0^{(I)}$ can recirculate back. This is revealed by the term $\mathbf{T}_{22}^{*-1} \mathbf{T}_{21}^* \mathbf{a}_0^{(I)}$ at the second line in Eq. (19). Third, recall we assume that the yellow lines in Fig. 2 are ideal connections, leading to the zero-one matrix in Eqs. (8) and (13). If the yellow lines are instead not ideal, we just need to revise the 2×2 zero-one matrix, while our derivation (as well as the later synthesis method) still holds.

We make two additional remarks related to the yellow direct connections in the top and bottom rows. First, from the application perspective, the yellow direct connections in the top and bottom rows of the mesh introduce a peculiarity. These connections break the connection symmetry of the mesh, and in particular break the clockwise/counterclockwise degeneracy of the square waveguide mesh. Normally, when injecting light in a square waveguide mesh, light will either circulate in a clockwise or counterclockwise direction inside a unit cell, but these circulations are not coupled. This means that in the scattering matrix of a square waveguide mesh, at least half of the elements are zero. By adding the connections in the top and bottom rows, these clockwise/counterclockwise circulations can be coupled and more generic mesh functions can be defined.

Second, from the calculation perspective, introducing the yellow direct connections lets us only need to provide the forward input $\mathbf{a}_0^{(I)}$ (i.e., $2N + 2$ scalars) if we assume the backward input $\mathbf{a}_M^{(O)} = \mathbf{0}$. However, without these yellow direct connections, we would have to set input values for those floating ports in the top and bottom rows; otherwise, the conditions are insufficient to determine the circuit response. Analytical gradients can still be calculated in such a case, but our derivation will need substantial modification.

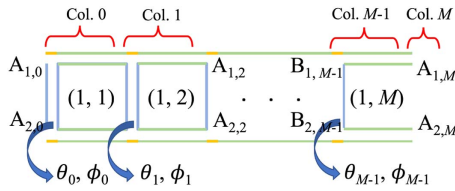


Fig. 5. Schematic of a $1 \times M$ square PPIC. The PSs in green horizontal TBUs are fixed to $\theta = 0$ and $\phi = \pi$.

3. HORIZONTAL RELAXATION

In the previous section, we derive the scattering matrix for a square-mesh PPIC in a general form. In this section, we consider a simplified case under the assumption of horizontal relaxation: all horizontal TBUs are configured with $\theta = 0$ and $\phi = \pi$, and thus operate in the bar state [4]. This implies that in Fig. 1, the light propagates from Port A_1 to B_1 , A_2 to B_2 , or in reverse, but does not go from A_1 to B_2 . Namely, when passing a horizontal TBU, the light is confined in the upper or lower arm.

As a starting point, we consider a $1 \times M$ square-mesh PPIC under this horizontal relaxation. Its schematic is shown in Fig. 5. In this case, the \mathbf{G}^* matrix defined in Eq. (20) is of size 4×4 . When M is odd, its expression is

$$\mathbf{G}^* = \begin{bmatrix} e^{-jM\omega\tau} & 0 & 0 & 0 \\ 0 & 0 & -\xi_M & 0 \\ 0 & \xi_M & 0 & 0 \\ 0 & 0 & 0 & -e^{-jM\omega\tau} \end{bmatrix} \quad (M = 1, 3, \dots), \quad (21)$$

and when M is even, its expression is

$$\mathbf{G}^* = \begin{bmatrix} e^{-jM\omega\tau} & 0 & 0 & 0 \\ 0 & \xi_M & 0 & 0 \\ 0 & 0 & \xi_M & 0 \\ 0 & 0 & 0 & e^{-jM\omega\tau} \end{bmatrix} \quad (M = 2, 4, \dots). \quad (22)$$

The $e^{-jM\omega\tau}$ term inside the matrix corresponds to the output at the top or bottom line in Fig. 5. Intuitively, this makes sense, since we have M horizontal TBUs at the top line, and under horizontal relaxation, these M TBUs function as M time delay elements. This in turn implies that the absolute value of ξ_M in Eqs. (21) and (22) is more interesting and can be utilized to synthesize light-processing functions. $|\xi_M|$ can be proven to have the following form:

$$|\xi_M| = \left| \frac{\prod_{m=0}^{M-1} (c_m f_m - d_m e_m)}{[0 \quad 1] \cdot \prod_{m=0}^{M-1} \begin{bmatrix} c_m & d_m \\ e_m & f_m \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}} \right|, \quad (23)$$

where $\{c_m, d_m, e_m, f_m\}$ are scalar values associated with the m th vertical TBU ($m = 0, 1, \dots, M-1$). Specifically, as shown in Fig. 5, if we denote the PSs inside the m th vertical TBU as $\{\theta_m, \phi_m\}$, we have the following relations:

$$\begin{aligned} c_m &= je^{-j2\omega\tau} \frac{p_m^2 - q_m^2}{2q_m}, \\ d_m &= -je^{-j\omega\tau} \frac{p_m}{q_m}, \\ e_m &= je^{j\omega\tau} \frac{p_m}{q_m}, \\ f_m &= -2je^{j2\omega\tau} \frac{1}{q_m}, \end{aligned} \quad (24)$$

where for simplicity, we have denoted $\tau(\omega) = \frac{n_{\text{eff}}(\omega)L}{c}$, and

$$\begin{aligned} p_m &= e^{-j\theta_m} - e^{-j\phi_m}, \\ q_m &= -e^{-j\theta_m} - e^{-j\phi_m}. \end{aligned} \quad (25)$$

With Eq. (24), the numerator in Eq. (23) can be proven to be 1, and the denominator is a polynomial in $e^{j2\omega\tau}$. As an example, we have

$$\begin{aligned} |\xi_1| &= \left| \frac{1}{2e^{j2\omega\tau}/q_0} \right| = \left| \frac{q_0}{2} e^{-j2\omega\tau} \right|, \\ |\xi_2| &= \left| \frac{1}{(-4e^{j4\omega\tau} + p_0 p_1)/q_0 q_1} \right| = \left| \frac{q_0 q_1 e^{-j4\omega\tau}}{-4 + p_0 p_1 e^{-j4\omega\tau}} \right|. \end{aligned} \quad (26)$$

Thus, a $1 \times M$ square PPIC, under horizontal relaxation, can be used to synthesize an IIR filter with zeros at the origin and poles generally complex.

A natural thought would be to extend the $1 \times M$ square PPIC under horizontal relaxation to an $N \times M$ square PPIC. Fortunately, due to the assumptions of our horizontal relaxation, this is straightforward. Specifically, in Eqs. (21) and (22), we have a \mathbf{G}^* matrix with a size of 4×4 corresponding to $N = 1$. For $N > 1$, we will have a \mathbf{G}^* matrix with a size of $(2N + 2) \times (2N + 2)$. Its first and last entries on the main diagonal will still be $e^{-jM\omega\tau}$, as in Eqs. (21) and (22). The middle part of \mathbf{G}^* will be filled with N different ξ_M in a similar way to in Eqs. (21) and (22), where the n th ξ_M corresponds to the n th row in the PPIC. To intuitively understand this, notice that the horizontal relaxation actually confines the horizontal propagation of signals in the same arm. Thus, the light propagating in the first row will never go to the second row, which means the transfer functions of two different rows are decoupled.

An important implication from this example is that even under this simplifying horizontal relaxation, the final transfer function shown in Eq. (23), though it has an analytical form, does not provide a direct solution—that is, it does not provide us with a direct analytical filter synthesis method. This motivates our optimization-based synthesis method proposed in the next section.

4. REALIZATION OF LIGHT-PROCESSING FUNCTIONS

In this section, we explain how we utilize our derivation to efficiently synthesize light-processing functions on an $N \times M$ square-mesh PPIC. Assume that we want to attain N light-processing functions represented by the complex transfer functions $\{U_n(\omega) | n = 1, 2, \dots, N\}$ specifying the magnitude and

phase responses in a range $[\omega_{\min}, \omega_{\max}]$. We choose N_{grid} frequency points $\{\omega_1 = \omega_{\min}, \omega_2 = \omega_{\min} + \Delta\omega, \dots, \omega_{N_{\text{grid}}} = \omega_{\max}\}$ in this desired angular frequency range with incremental step equal to $\Delta\omega$. Then we can define an error or cost function,

$$\text{Cost} = \sum_{k=1}^{N_{\text{grid}}} \sum_{n=1}^N |a_{2n,M}^{(I)}(\omega_k) - U_n(\omega_k)|^2. \quad (27)$$

Note that here we have made the dependence of $a_{2n,M}^{(I)}$ on the angular frequency explicit. If we can make the cost in Eq. (28) sufficiently small by adjusting all PSs $\{\theta, \phi\}$ of all vertical and horizontal TBUs, then we succeed in synthesizing the n th light-processing functions at the bottom port of the n th row (i.e., $A_{2n,M}$). This can be done by using an optimization technique: we minimize the cost in Eq. (28) with respect to all PSs $\{\theta, \phi\}$,

$$\min_{\mathbf{x}} \text{Cost} = \sum_{k=1}^{N_{\text{grid}}} \sum_{n=1}^N |a_{2n,M}^{(I)}(\omega_k, \mathbf{x}) - U_n(\omega_k)|^2, \quad (28)$$

where we use \mathbf{x} to collectively represent all PSs $\{\theta, \phi\}$ and make the dependence of \mathbf{x} explicit in the cost function.

However, the difficulty lies in the fact that this optimization problem is extremely high-dimensional. For an $N \times M$ square-mesh PPIC as shown in Fig. 2, it has $2(2N + 1)M$ PSs in total. Considering a fairly small 10×10 PPIC, there are already 420 PSs to tune. To the best of our knowledge, such a high-dimensional optimization problem is inefficient to solve unless using a gradient descent method with analytical gradients. Specifically, nongradient methods take a long time to converge, and gradient descent methods based on numerical differentiation require many function evaluations to calculate the gradient once. Importantly, in our case, we do have the analytical derivative $\partial \text{Cost} / \partial \theta$ or $\partial \text{Cost} / \partial \phi$ for any θ and ϕ based on our previous derivations, because the operations that relate θ (or ϕ) to the variable Cost are all differentiable. As a result, we can use gradient descent optimization to minimize Eq. (28) to perform the synthesis task. For details about how to calculate the gradient, please refer to Appendix B.

We note that in some applications, the desired light-processing functions only have requirements on the magnitude, but with no constraints on the phase. In such cases, we can choose $\{U_n(\omega) | n = 1, 2, \dots, N\}$ to be real functions representing the desired magnitude response and revise the cost in Eq. (28) as

$$\text{Cost}_{\text{Linear Mag}} = \sum_{k=1}^{N_{\text{grid}}} \sum_{n=1}^N r_k |a_{2n,M}^{(I)}(\omega_k, \mathbf{x}) - U_n(\omega_k)|^2, \quad (29)$$

where r_k ($k = 1, 2, \dots, N_{\text{grid}}$) is a user-defined positive real scalar controlling the weight ratio. As will be demonstrated in our numerical results, we find that building upon Eq. (29) and using logarithm magnitude works even better, especially for synthesizing an optical filter where stop band and passband have very different magnitude requirements. This logarithm cost is

$$\text{Cost}_{\text{Log Mag}} = \sum_{k=1}^{N_{\text{grid}}} \sum_{n=1}^N r_k |\ln |a_{2n,M}^{(I)}(\omega_k, \mathbf{x})| - \ln U_n(\omega_k)|^2. \quad (30)$$

5. NUMERICAL RESULTS

In all our numerical experiments, we choose $n_{\text{eff}} = 2.35$, $L = 250 \mu\text{m}$, $c = 3 \times 10^8 \text{ m/s}$, and $\alpha = 0.99$. We do not take dispersion effects into account (i.e., n_{eff} is considered to be constant and independent of ω , which means that $n_g = n_{\text{eff}} = 2.35$). Real waveguides do have dispersion, but this does not affect the method, as long as the dispersion of n_{eff} can be described by an analytically derivable function (e.g., with the help of n_g). Moreover, we emphasize that in high refractive index contrast platforms, n_{eff} usually depends on frequency, and the dispersion effect causes a narrow free spectral range (FSR) in the PPIC. Before moving on, we define a value for later simplicity,

$$\Delta f = \frac{c}{n_g L} = \frac{3 \times 10^8}{2.35 \times 250 \times 10^{-6}} \approx 510.638 \text{ GHz} \quad (31)$$

When plotting the figures of frequency response, we will normalize the frequency x axis following the rule,

Table 1. Detailed Information for All Our Experiments^a

	Input Port	Output Port	Target (s)	Cost	Results	Run Time	Phase Acc/FSR ^b
No. 1, routing	A _{1,0}	A _{2,5}	Mag, phase	Eq. (28)	Fig. 6	0.27 min	8L
No. 2, splitting	A _{1,0}	A _{2,5} , A _{4,5} , A _{6,5}	Mag	Eq. (29)	Fig. 7	1.09 min	8L, 10L, 17L
No. 3, splitting (c)	A _{1,0}	A _{3,5} , A _{7,5}	Mag, phase	Eq. (28)	Fig. 8	0.63 min	10L, 10L
No. 4, splitting (c)	A _{5,0}	A _{1,5} , A _{3,5} , A _{7,5} , A _{9,5}	Mag, phase	Eq. (28)	Fig. 9	4.48 min	$\frac{\Delta f}{2}, \frac{\Delta f}{2}, \frac{\Delta f}{4}, \frac{\Delta f}{4}$
No. 5, filtering	A _{1,0}	A _{2,5}	Mag	Eq. (30)	Fig. 10	81.59 min	$\frac{\Delta f}{2}$
No. 6, WDM	A _{5,0}	A _{3,5} , A _{7,5}	Mag	Eq. (30)	Fig. 11	108.25 min	$\frac{\Delta f}{12}, \frac{\Delta f}{12}$
No. 7, WDM and filtering	1 at A _{1,0} , 1j at A _{10,0}	{A _{2,5} , A _{6,5} }, A _{10,5}	Mag	Eq. (30)	Fig. 12	110.34 min	$\frac{\Delta f}{12}, \frac{\Delta f}{12}, \frac{\Delta f}{2}$

^aAll are performed on a 5×5 square mesh. “c” is short for “coherent.” “Mag” is short for “magnitude.” When using the logarithm cost Eq. (30), we set r_k to 10 and 1 for frequency points in the passband and stop band, respectively. If there is no stop band (e.g., Case 2), r_k is set to 1.0 for all k .

^bIn Cases 1, 2, and 3, the synthesized results have no interference; we use phase accumulation to depict how many TBUs the light path passes through (e.g., 8L). In Cases 4, 5, 6, and 7, interference occurs and it becomes less clear that the light path goes through a specific number of TBUs. Thus, we use the metric FSR (e.g., $\Delta f/2$) in these cases.

$$f_{\text{norm}} = \frac{2}{\Delta f} (f - f_{\text{center}}), \quad (32)$$

where f_{norm} and f represent the frequency value after and before normalization, respectively. Here f_{center} represents the center frequency,

$$f_{\text{center}} = \frac{c}{\lambda_{\text{center}}} = \frac{3 \times 10^8}{1550 \times 10^{-6}} \approx 193.548 \text{ THz} \quad (33)$$

For instance, Eq. (32) will map $\{f_{\text{center}} - 0.5\Delta f, f_{\text{center}}, f_{\text{center}} + 0.5\Delta f\}$ to $\{-1, 0, 1\}$, respectively. Recall that FSR represents the periodicity of the frequency response when interference occurs. Provided our introduced notation Δf , these statements are equivalent: (i) $\text{FSR} = c/n_g(KL) = \Delta f/K$; and (ii) in the normalized frequency figure, a range of $[-1, 1]$ corresponds to K periods, or one period has the length $2/K$. Defining the value Δf and plotting the frequency response on a normalized frequency axis give us a consistent way to visualize the results in different examples.

Our algorithm is implemented in Python, and all our numerical experiments are performed on the same RedHat Linux server with 16 Intel Xeon E7-4850 CPUs working at 2.1 GHz. The initial guess required by the gradient descent optimization is randomly generated, consistent with our claim that our synthesis method does not require human design knowledge. However, we emphasize that in most of our examples, we are optimizing an interferometric system with many phase variables, and thus the cost function for most configurations will have many local peaks and valleys. The specific configuration coming out of the optimization algorithm will therefore depend strongly on the initial condition. Table 1 comprehensively lists the detailed information of all our experiments. In the following paragraphs, we comment on each case.

For Case 1, we consider routing the input light to an output port with minimum cost over the entire frequency band. Results are shown in Fig. 6. The synthesized path shown in Fig. 6(e) has gone through eight TBUs. Thus, according to Eq. (4), we know that the synthesized configuration has a phase accumulation corresponding to $8L$, or more specifically, that the output port has an $e^{-j\frac{\omega n_g 8L}{c}}$ dependence. This implies that we should witness a phase change of 2π over a frequency range of $c/n_g(8L) = \Delta f/8$, i.e., an interval with length 0.25 in the normalized frequency figure. This is indeed the case, as shown in Fig. 6(h). Also, if zooming in, Fig. 6(h) is exactly the same as Fig. 6(b). Since we have considered a loss term $\alpha = 0.99$ in our compact TBU model, the synthesized normalized power transmission shown in Fig. 6(g) cannot reach 0 dB. We see that the synthesized light path shown in Fig. 6 relies on the top line and passes through eight TBUs, and a quick calculation shows $20 \log 0.99^8 \approx -0.70$, consistent with Fig. 6(g). Last, but not least, Fig. 6(d) also demonstrates that only the first few rows have been adjusted by the optimization routine. This is as expected, since our input and output ports are both located at the top part of the mesh.

For Case 2, we consider equal power splitting to three output ports. Results are shown in Fig. 7. We note that due to reciprocity, combining three light inputs can also be readily solved. As shown in Fig. 7(d), the three light paths pass through 8, 10, and 17 TBUs, respectively, implying the three output

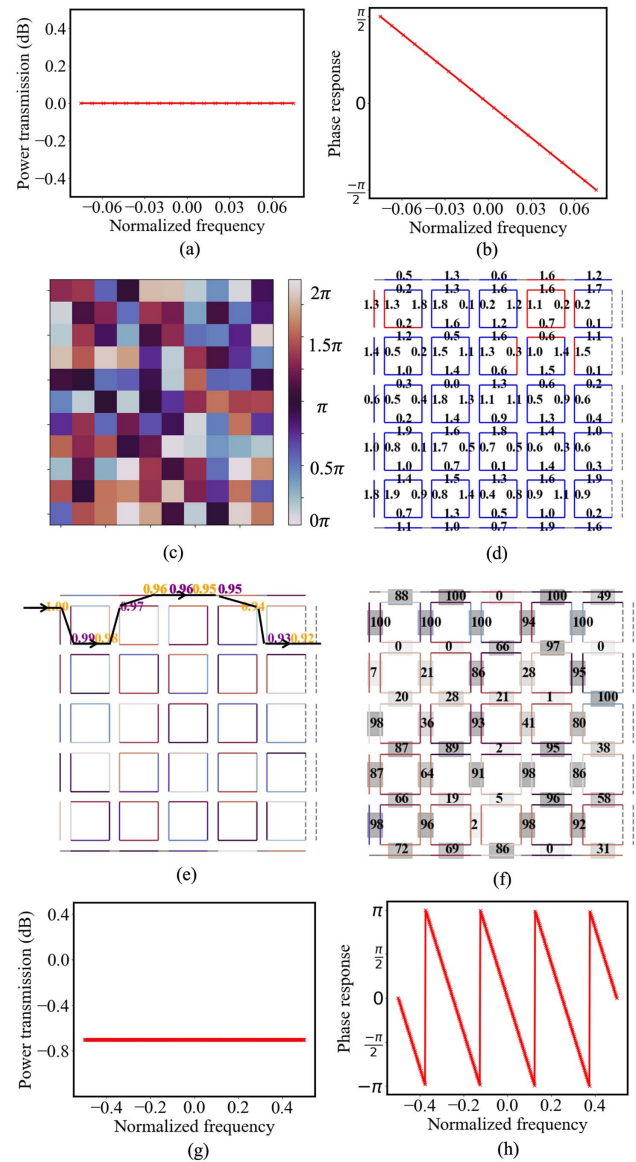


Fig. 6. Case 1, routing. (a) and (b) show the target response $U(\omega)$ with magnitude normalized to input, and phase, respectively, used in the cost function. (c) shows a heat map of all optimized PS values (see Appendix C for colored cell ordering details). (d) shows the resulting optimized configuration (π omitted). Red lines are those PS changes larger than 0.2π before and after optimization, while blue lines are those with changes smaller than 0.2π . (e) shows the port magnitude at frequency f_{center} —orange for inward direction and purple for outward direction (refer to Fig. 3 for definition). Port magnitudes less than 0.2 are not drawn. The light path is plotted in black. (f) shows the power coupling ratio (i.e., $\cos^2 \frac{\phi-\theta}{2}$) of each TBU with a percentage in a shaded bounding box. The edge color of the TBU shows common PS $\frac{\pi-\phi-\theta}{2}$; see Appendix D; (g) synthesized magnitude response; (h) synthesized phase response; (g) and (h) show the frequency response that the optimized configuration is able to achieve. The square meshes in (e) and (f) share the same color bar, shown in (c).

responses should have phase accumulations corresponding to $8L$, $10L$, and $17L$. Namely, we will see a phase change of 2π over a frequency range of $\Delta f/8$, $\Delta f/10$, and $\Delta f/17$, respectively, corresponding to an interval with length $2/8$, $2/10$,

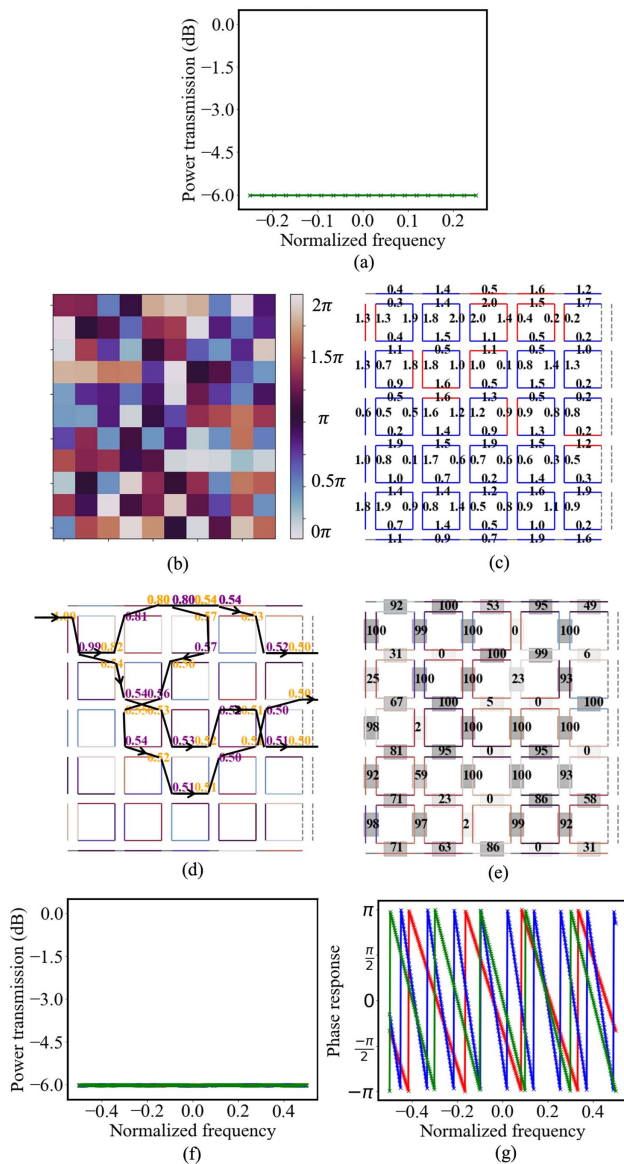


Fig. 7. Case 2, splitting. (a) Target equal three-way split magnitude response (normalized to the input); (b) heat map of all optimized PS values; (c) optimized configuration (π omitted); (d) port magnitude at frequency f_{center} ; (e) power coupling ratio and common PS; (f) synthesized magnitude response; (g) synthesized phase response; (f) and (g) show the frequency response that the optimized configuration is able to achieve. There are three lines colored in red, blue, and green in (a), and they overlap here and in (f).

and $2/17$ in the normalized frequency figure. As shown in Fig. 7(g), these correctly reflect the 4, 5, and 8.5 periods in the interval of $[-0.5, 0.5]$. One subtlety here is that when designing the target function $U(\omega)$, we consider the power loss due to $\alpha = 0.99$ and provide some margin in advance. Namely, the target magnitude chosen here is 0.5 on a linear scale [i.e., about -6.0 dB in Fig. 7(a)], such that $0.5^2 \times 3 = 0.75 < 1.0$. Alternatively, choosing all three target magnitudes to be $\sqrt{1/3}$ on a linear scale would be problematic. From a numerical perspective, the optimization routine would seek

to push all three output magnitudes to $\sqrt{1/3}$, but since this is unattainable simultaneously due to the loss term α , it could happen that the resulting three outputs would be unequal (e.g., $\sqrt{0.30}, \sqrt{0.31}, \sqrt{1/3}$). Using a target magnitude that is attainable, as we do here, can prevent this issue. However, one side effect of a preprovided power loss margin is that it might encourage the light path to go through more TBUs. For instance, the zigzag light path with $17L$ in this example is only one possible solution. It is obvious from Fig. 7(d) that this light path could propagate to the right bottom direction at the port with magnitude 0.57 in the middle, instead of going to the left bottom as it currently does.

For Case 3, we consider coherent splitting. Namely, we want to split the input light to two output ports but now with identical phase. Results are shown in Fig. 8. As seen in Fig. 8(e), both light paths pass through 10 TBUs, implying that a frequency range of $\Delta f/10$ (i.e., an interval with length 0.2 in the normalized frequency figure) is required for a phase change of 2π . This is also confirmed by Fig. 8(h). Moreover, we note that in a 5×5 square mesh, without using the top or bottom line, the minimum number of TBUs required to propagate light from a port at left to a port at right is 10. Moreover, the optimization routine obtains a synthesized result that seems natural and readily understandable. Namely, we chose the output port row indices to be 3 and 7 in this case, while the input port row index was 1 (see Table 1). The resulting synthesized light path first goes from top left to the bottom right direction without any splitting, and then approximately stops at the middle between the output ports. Then it performs a 50%:50% power splitting and the resulting two light paths keep propagating without further splitting all the way to the output ports. This approach of first propagating to the middle followed by a 50%:50% splitting is a generic strategy to synthesize one-input to two-output coherent splitting and is automatically found by the optimization.

For Case 4, we consider a more complicated version of Case 3. Now, we attempt to do coherent splitting to four output ports. Results are shown in Fig. 9. Due to the structure of the square mesh, it is actually impossible to find four light paths all with the same length, meaning that the goal in this case is unachievable. Specifically, because the four output ports do not belong to the same clockwise/counterclockwise sub-mesh, there will be at least one L light path difference. As shown in Fig. 9(e), it seems that the optimization attempts to utilize interference to approach this unattainable goal as closely as possible. From Fig. 9(g), we see that the red and blue curves both have an FSR equal to $0.5\Delta f$, while the green and cyan curves both have an FSR equal to $0.25\Delta f$. The difference of FSRs also indicates that we cannot achieve coherent splitting at an arbitrary frequency point, since these paths have a periodicity mismatch. This is also verified in Figs. 9(g) and 9(h). Our synthesized results do satisfy the given targets shown in Figs. 9(a) and 9(b): the optimization achieves coherent splitting in the normalized range $[-0.05, 0.05]$, which corresponds to around a 25 GHz range in reality. However, we also notice that outside this range, the optimization cannot always achieve coherent splitting. An important note is that there are several rings in the synthesized configuration in this case and explains why

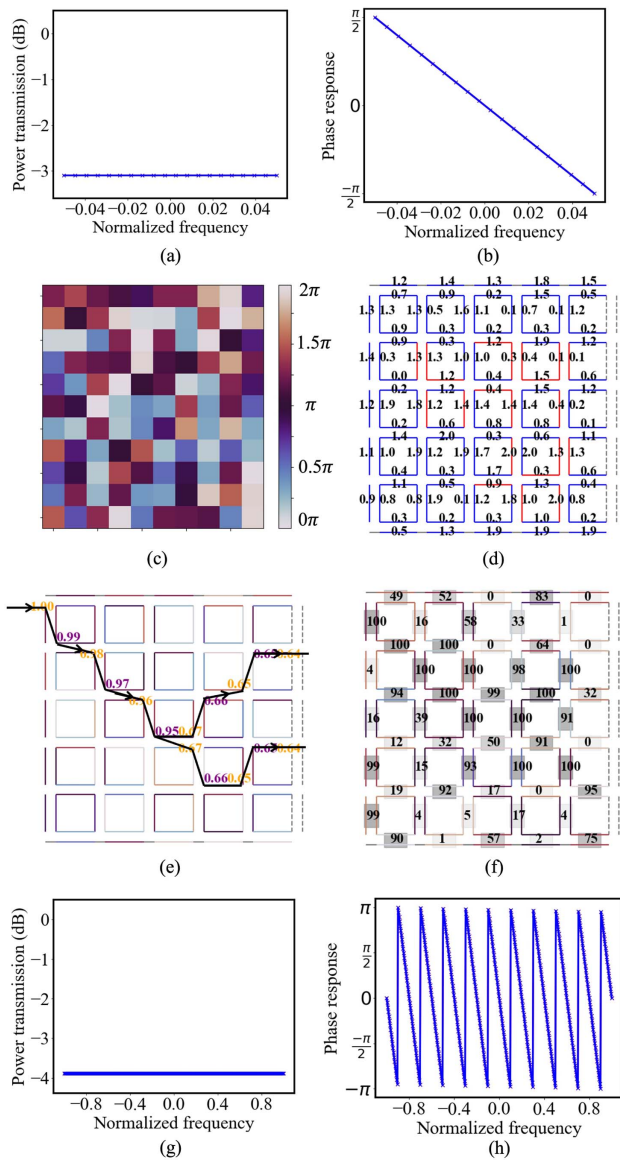


Fig. 8. Case 3, coherent two-way splitting. (a) and (b), respectively, show the target equal magnitude split with equal phase response. (c) Heat map of all optimized PS values; (d) optimized configuration (π omitted); (e) port magnitude at frequency f_{center} ; (f) power coupling ratio and common PS; (g) synthesized magnitude response; (h) synthesized phase response; (g) and (h) show the frequency response that the optimized configuration is able to achieve. There are two lines colored in red and blue in (a), and they overlap here and in (b), (g), and (h).

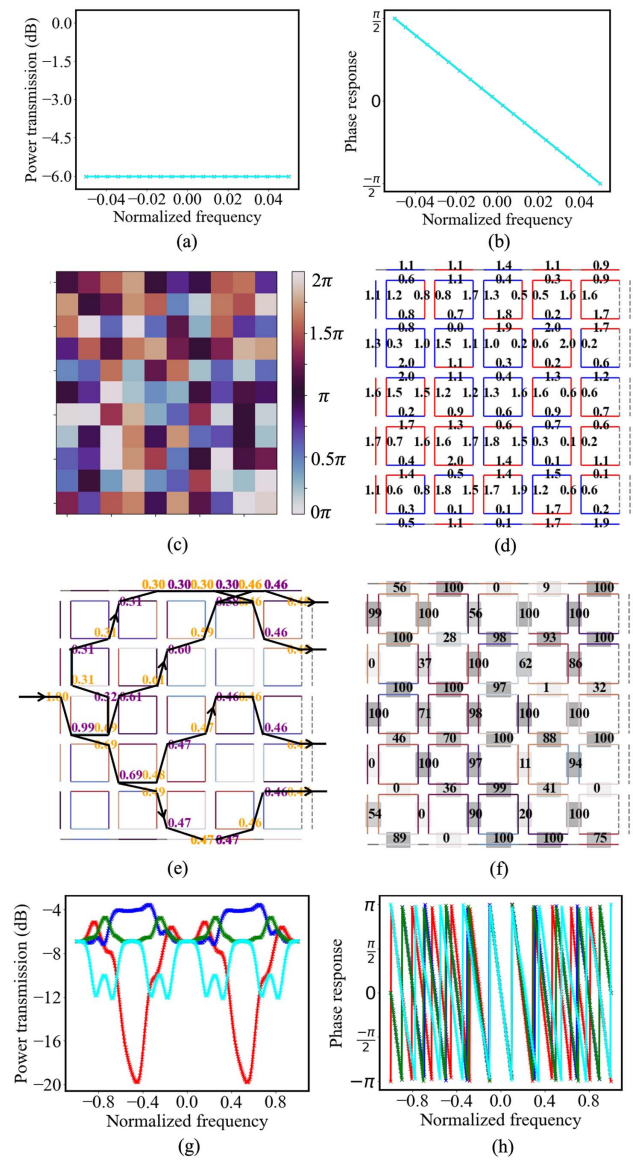


Fig. 9. Case 4, coherent four-way splitting. (a) and (b), respectively, show the target equal four-way magnitude split and phase response. (c) Heat map of all optimized PS values; (d) optimized configuration (π omitted); (e) port magnitude at frequency f_{center} ; (f) power coupling ratio and common PS; (g) synthesized magnitude response; (h) synthesized phase response; (g) and (h) show the frequency response that the optimized configuration is able to achieve. There are four lines colored in red, blue, green, and cyan in (a), and they overlap here and in (b).

we obtain a frequency-dependent response in Fig. 9(g). However, port magnitudes associated with some of the rings are smaller than 0.2, and thus are not drawn.

For Case 5, we consider optical filtering. Results are shown in Fig. 10. As seen in Fig. 10(d), many rings have formed in the obtained configuration. We successfully achieve near 0 dB in the passband, and about -70 dB in the stop band. The FSR is about $0.5\Delta f$, as depicted in Fig. 10(f).

As Case 6, we consider two-way wavelength division multiplexing (WDM), also called an optical interleaver, where the

spectrum is separated into even and odd frequency channels over two outputs. From the results in Fig. 11(d), it is clear that many rings have formed in the optimized configuration. Moreover, Fig. 11(d) is plotted at the central frequency, and thus the other output port magnitudes are less than 0.3 and not drawn.

For Case 7, we consider synthesizing two light-processing functions (WDM and optical filtering) at the same time, given two in-phase inputs. Namely, we provide a complex input $1.0 + 0.0j$ at $A_{1,0}$ and a complex input $0.0 + 1.0j$ at $A_{10,0}$.

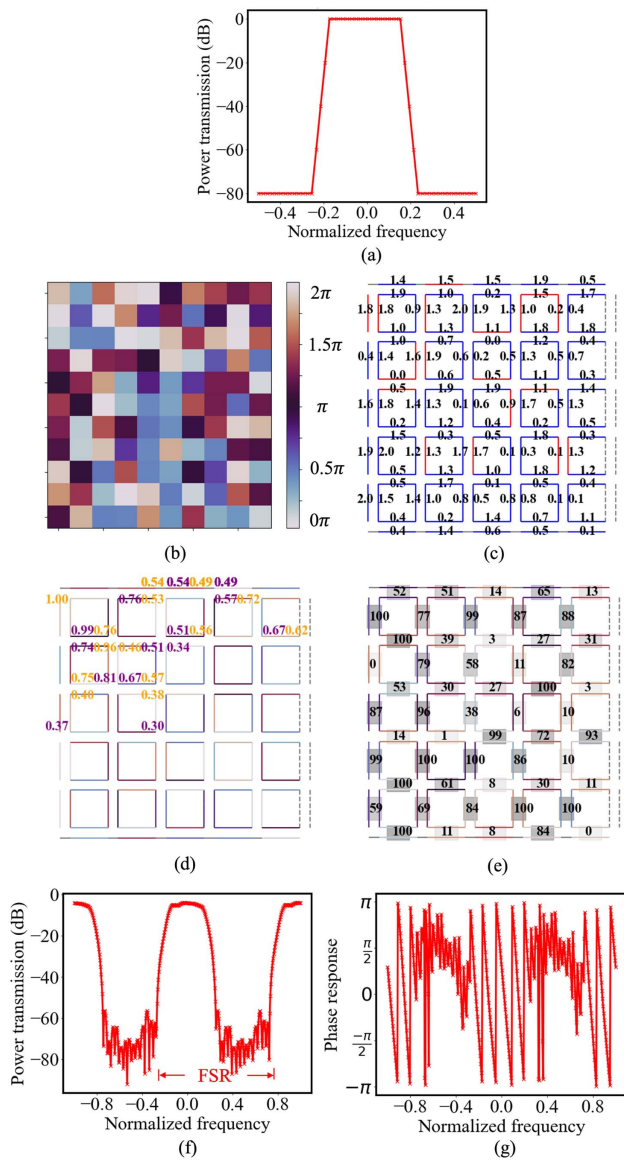


Fig. 10. Case 5, optical filtering. (a) Target magnitude response; (b) heat map of all optimized PS values; (c) optimized configuration (π omitted); (d) port magnitude at frequency f_{center} ; (e) power coupling ratio and common PS; (f) synthesized magnitude response; (g) synthesized phase response; (f) and (g) show the frequency response that the optimized configuration is able to achieve. Note that in this case, only port magnitudes over 0.3 are plotted in (d) for clarity.

The two output ports for WDM are $A_{2,5}$ and $A_{6,5}$, while that for filtering is $A_{10,5}$. Results are shown in Fig. 12. Note that in Fig. 12(d), we see that some inner port magnitudes are larger than 1.0. This is possible because (i) the total input power is 2.0, and (ii) when a ring is formed, it can lead to the “intensity buildup” phenomenon [27,28] near resonance.

To better quantify the performance of our method, we implement two baseline methods for comparison: (i) differential evolution, a population-based gradient-free global optimization approach; and (ii) gradient descent optimization with numerical differentiation. Table 2 summarizes the run time of our method and the two baselines. We see that our proposed

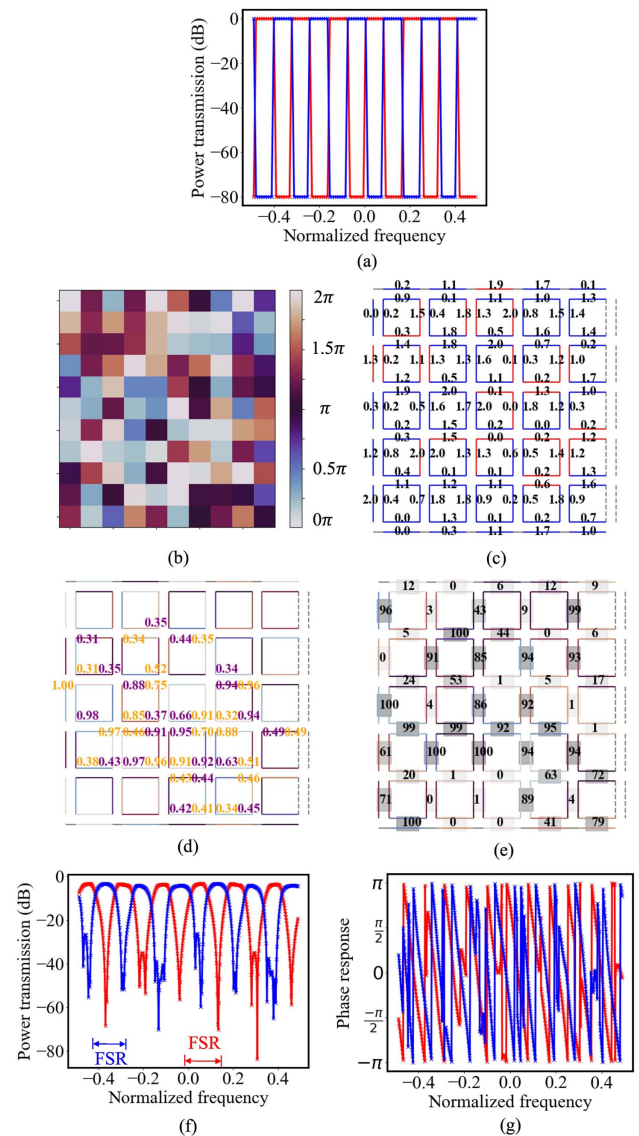


Fig. 11. Case 6, WDM. (a) Target magnitude response; (b) heat map of all optimized PS values; (c) optimized configuration (π omitted); (d) port magnitude at frequency f_{center} ; (e) power coupling ratio and common PS; (f) synthesized magnitude response; (g) synthesized phase response; (f) and (g) show the frequency response that the optimized configuration is able to achieve. Note that in this case, only port magnitudes over 0.3 are plotted in (d) for clarity.

method achieves about $3\times$ computation time cost reduction compared with the implemented baseline methods.

We emphasize that gradient descent optimization (with potentially nonconvex cost functions such as ours) is known to be only able to find local minima, and thus the specific configuration coming out of the optimization algorithm will depend strongly on the initial condition. To justify the practical utility of the proposed method, we also need to show that even with different initializations, the optimization routine can always yield a good result. Due to space limitations, we take Cases 1 and 5 as examples. We run our method with different initializations and plot the results in Figs. 13 and 14. These

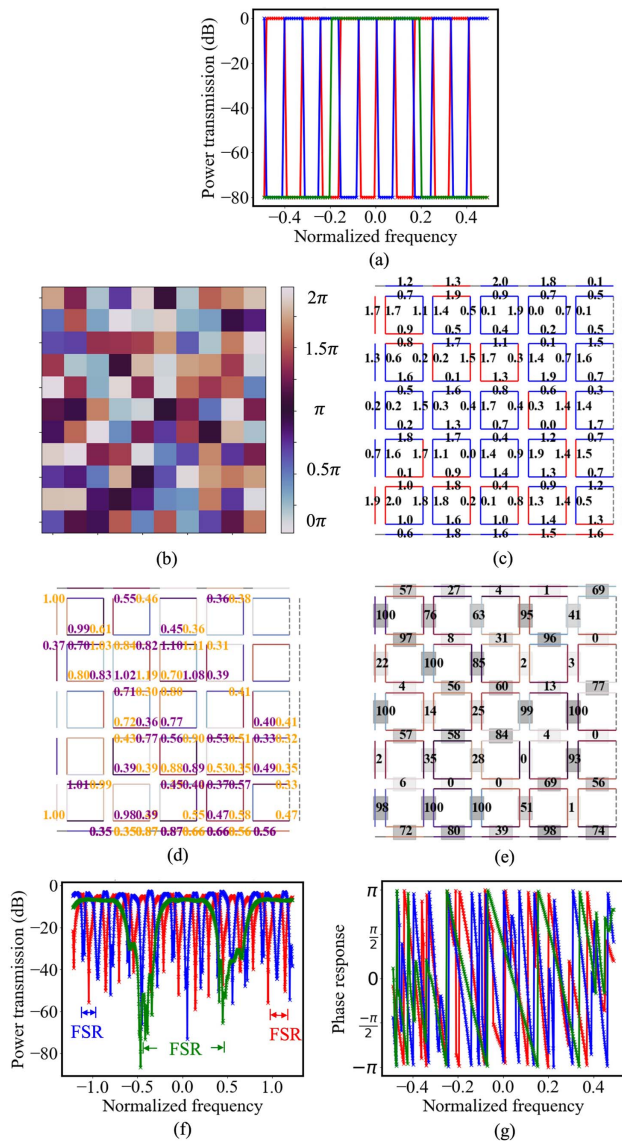


Fig. 12. Case 7, simultaneously synthesizing two light-processing functions for two in-phase inputs. Figure caption is similar to that of Fig. 6, except that in this case, only port magnitudes over 0.3 are plotted in (d) for clarity.

Table 2. Run Time (in min) Comparison of Our Method with DE and ND^a

	Ours	DE	ND
No. 1, routing	0.27	>100	≈22.44
No. 2, splitting	1.09	>100	≈78.02
No. 3, splitting (c)	0.63	>100	≈61.72
No. 4, splitting (c)	4.48	>100	≈80.26
No. 5, filtering	81.59	>400	>400
No. 6, WDM	108.25	>400	>400
No. 7, WDM and filtering	110.34	>400	>400

^aDE is short for differential evolution, a population-based gradient-free global optimization approach. ND is short for gradient descent optimization with numerical differentiation. We stop DE/ND when the synthesized results attain similar cost values to our method or similar curve shapes in the magnitude or/and phase response figures. The “>” sign indicates that the corresponding algorithm’s result is not comparable to ours within the specified time.

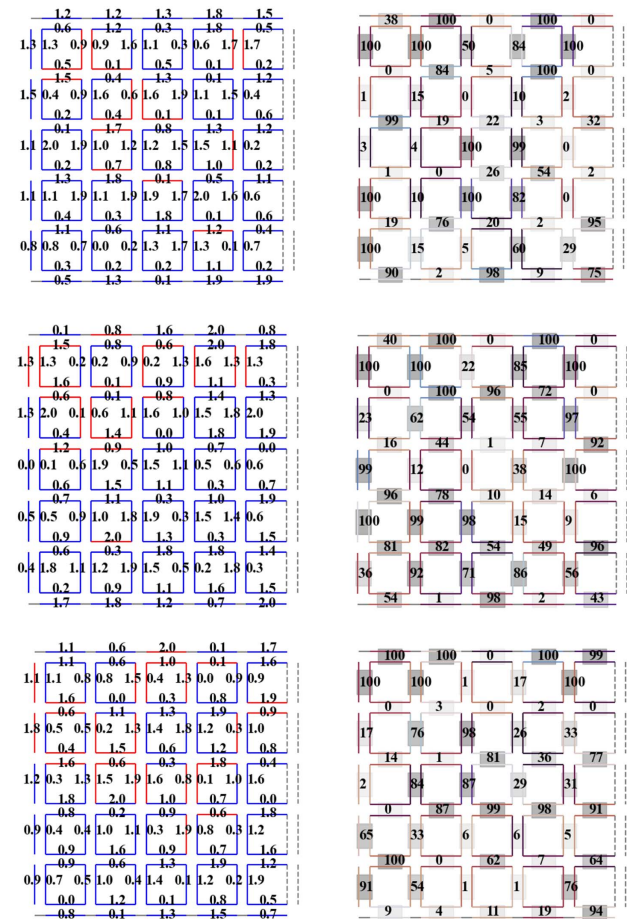


Fig. 13. Three different configurations are obtained under three random initializations; all satisfy the goal of routing in Case 1. Each row represents one synthesized configuration. Figure 6 is not included here. Left column, the optimized configuration (π omitted); right column, power coupling ratio (%) and common PS; all synthesized magnitude responses are identical to those in Fig. 6(g), hence not shown.

demonstrate the robustness of our method to random initialization. Note that for our applications, we do not necessarily need a global optimum, while a locally optimal configuration is already sufficient. Note that when the PPIC size further scales up, we would expect the optimization result to be more strongly impacted by the initialization, because more local optima might exist for a higher dimensional optimization problem.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we propose an efficient synthesis method that can be applied to realize configurations for a wide range of light-processing functions on a square-mesh PPIC. The key property that makes our method efficient is that we analytically derive the gradients of the mean squared error, or the log ratio, between target and realized circuit response with respect to the tunable PSs based on scattering matrix theory. Then, a gradient

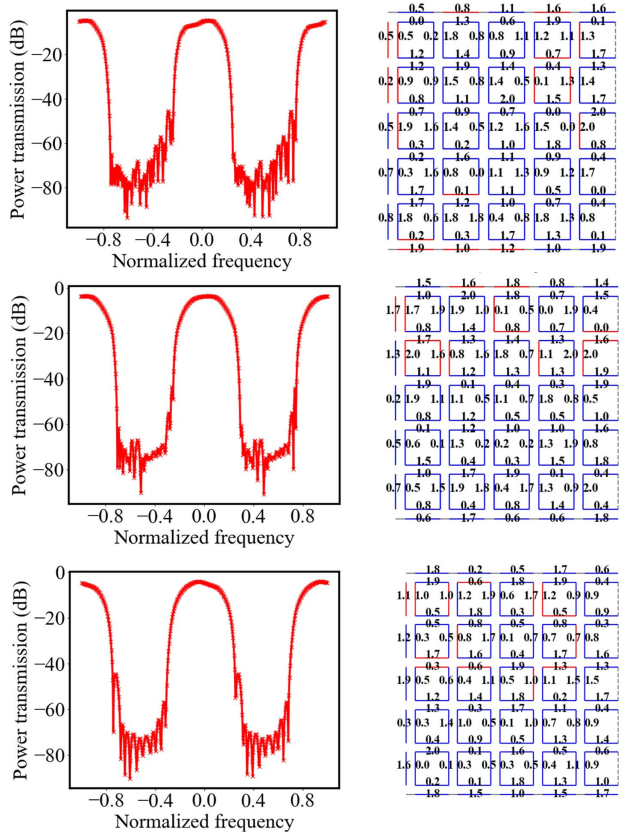


Fig. 14. Three different configurations are obtained under three random initializations; all satisfy the goal of filtering in Case 5. Each row represents one synthesized configuration. Figure 10 is not included here. Left column, synthesized magnitude response; right column, optimized configuration (π omitted).

descent optimization can be carried out to synthesize the desired light-processing functions at time scales of minutes.

Other PPIC connection topologies: We consider a square mesh in this paper because it provides the clearest derivation of the scattering matrix elements, compared to triangular and hexagonal meshes, due to the fact that the TBUs are placed either vertically or horizontally. Nevertheless, we emphasize that even though identifying column 1, 2, ..., M as shown in Fig. 2 is harder for a triangular or hexagonal mesh, it is still possible, and thus our method is also applicable to these topologies. For example, the authors in Ref. [24] have successfully derived a system-level transfer function for a hexagonal mesh using an approach similar to ours. However, when a mixture of triangular, square, and hexagonal mesh is used, or an arbitrary connection of TBUs is adopted, the current implementation of our method or Ref. [24] can fail because it may not be possible to divide the circuit into columns, and we both build the scattering matrix iteratively going through column by column. In the future, we will expand our approach to arbitrary connection topologies.

Dealing with nonideality: Two assumptions used in this paper (i.e., omitting the last column and assuming ideal yellow lines in Fig. 2) are only for ease of mathematical notation. These assumptions can be relaxed, consistent with our method.

In a real-world scenario, dispersion effects can exist either due to the frequency-dependent effective index of the waveguide, or due to a frequency-dependent power coupling ratio in the 50%:50% DCs. Furthermore, each building block TBU might be slightly different due to process variations in manufacturing. All of these nonidealities can be addressed, as long as we can describe them as differentiable functions of frequency. Indeed we can do so, for example by using a Taylor expansion and introducing a group index for dispersion.

Another major source of nonideality comes from the thermal cross talk of heaters, which we have not considered in the main text. However, by using thermal eignemode decomposition [29], our proposed method remains applicable under a change of optimized variables and can account for thermal cross talk. A similar treatment can be adopted for other nonidealities of actuators in PPICs. Please refer to Appendix E for details.

Last, but not least, we emphasize that when nonidealities (e.g., process variation, dispersion effect, or beam-splitting error) are considered, a more complex variant of the proposed compact model might be needed. Please refer to Appendix F for details.

Numerical considerations: As shown in our numerical results, choosing an appropriate cost function is of crucial importance, especially in a case when both a stop band and a passband are present at the same time. In our paper, we use a weighted logarithm cost function for such cases, but note that other options also exist, such as a combination of linear and logarithm cost [30]. The choice of cost function can substantially impact the optimized results, and it will be interesting to consider if better cost functions exist. When doing so, it will be beneficial to explore differentiable cost functions, since gradient descent optimization is preferred in this problem. Exploration could also overcome another limitation of our current cost functions. Ideally, we want the cost to be zero if the achieved rejection ratio (e.g., $|-80|$ dB) is already larger than the target (e.g., $|-70|$ dB) in the stop band. However, implementing this threshold strategy might degrade the performance of gradient descent optimization, as it will introduce nondifferentiable points in the optimization search space.

We also note that the matrix \mathbf{V} shown in Eq. (6) is not invertible when the two PSs $\{\theta, \phi\}$ have a π difference (i.e., $\theta = \pi + \phi$ or vice versa). In this case, our approach will fail. This is consistent with our intuition: when the PSs have a π difference, the vertical TBU is in the bar state, and knowing all port magnitudes related to “A” does not confer any knowledge on the port magnitudes related to “B” [see Fig. 3 and Eq. (6)]. In a real numerical implementation, this means that if the phase difference $|\phi - \theta|$ is close to π , then the associated \mathbf{V} will be ill-conditioned, and our simulated frequency response at ports and the gradients might be inaccurate. Fortunately, in our optimized results, we have not encountered this issue. Observant readers might find that, for example, in Fig. 11(e), there exist a few vertical TBUs with a power coupling ratio reported as 0, implying that they are in bar states. However, this is because we only display up to integer percentage when drawing the figure for space reasons. To support the claim made in the main text, we have printed the condition number of \mathbf{T}^* defined in Eq. (18) as a way to examine numerical stability

when running the program. But we warn of the need to pay attention to this case. In the future, this problem should readily be solved when we expand our approach to support any connections, since at that time, our scattering matrix will be set up based on graph theory, without requiring inverting \mathbf{V} .

APPENDIX A: JUSTIFICATION FOR THE COMPACT MODEL

Here we justify the validity of the compact TBU model given in Eq. (2). Considering all possible locations of waveguides in a TBU, the most general form of the compact model is

$$\mathbf{F} = \frac{1}{2} \mathbf{\Lambda}_1 \begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix} \mathbf{\Lambda}_2 \begin{bmatrix} e^{-j\theta} & 0 \\ 0 & e^{-j\phi} \end{bmatrix} \mathbf{\Lambda}_3 \begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix} \mathbf{\Lambda}_4, \quad (\text{A1})$$

where we have defined $\mathbf{\Lambda}_i$ ($i = 1, 2, 3, 4$),

$$\mathbf{\Lambda}_i = \begin{bmatrix} \alpha_i e^{-j\omega \frac{n_{\text{eff}} L_i}{c}} & 0 \\ 0 & \alpha'_i e^{-j\omega \frac{n_{\text{eff}} L'_i}{c}} \end{bmatrix}, \quad (\text{A2})$$

and $\{L_i, L'_i\}$ represent the length of waveguide in the upper and lower arm, respectively. For instance, here $\mathbf{\Lambda}_2$ represents the waveguides between the right DC and the middle PSs. As long as the waveguides are balanced in the upper and lower arms (i.e., $L_i = L'_i$ and $\alpha_i = \alpha'_i$ for all $i = 1, 2, 3, 4$), then Eq. (A1) can be simplified as

$$\mathbf{F} = \frac{1}{2} \begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix} \begin{bmatrix} e^{-j\theta} & 0 \\ 0 & e^{-j\phi} \end{bmatrix} \begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix} \alpha e^{-j\omega \frac{n_{\text{eff}} L}{c}}, \quad (\text{A3})$$

where L equals the sum of all L_i , and α equals the product of all α_i ,

$$L = \sum_{i=1}^4 L_i, \quad \alpha = \prod_{i=1}^4 \alpha_i. \quad (\text{A4})$$

Fortunately, the assumption of balanced waveguides is only a mild one: imbalanced waveguides are seldom used in PPICs, so our compact model remains applicable.

APPENDIX B: EXAMPLE OF CALCULATING THE GRADIENT

According to Eq. (20) of the main text, we can calculate the derivative of $\mathbf{a}_M^{(l)}$ with respect to one PS θ ,

$$\begin{aligned} \frac{\partial \mathbf{a}_M^{(l)}}{\partial \theta} &= \frac{\partial}{\partial \theta} (\mathbf{G}^* \mathbf{a}_0^{(l)}) = \frac{\partial \mathbf{G}^*}{\partial \theta} \mathbf{a}_0^{(l)} = \frac{\partial (\mathbf{T}_{11}^* - \mathbf{T}_{12}^* \mathbf{T}_{22}^{*-1} \mathbf{T}_{21}^*)}{\partial \theta} \mathbf{a}_0^{(l)} \\ &= \left(\frac{\partial \mathbf{T}_{11}^*}{\partial \theta} - \frac{\partial \mathbf{T}_{12}^*}{\partial \theta} \mathbf{T}_{22}^{*-1} \mathbf{T}_{21}^* \right. \\ &\quad \left. - \mathbf{T}_{12}^* \frac{\partial \mathbf{T}_{22}^{*-1}}{\partial \theta} \mathbf{T}_{21}^* - \mathbf{T}_{12}^* \mathbf{T}_{22}^{*-1} \frac{\partial \mathbf{T}_{21}^*}{\partial \theta} \right) \mathbf{a}_0^{(l)} \\ &= \left(\frac{\partial \mathbf{T}_{11}^*}{\partial \theta} - \frac{\partial \mathbf{T}_{12}^*}{\partial \theta} \mathbf{T}_{22}^{*-1} \mathbf{T}_{21}^* \right. \\ &\quad \left. + \mathbf{T}_{12}^* \mathbf{T}_{22}^{*-1} \frac{\partial \mathbf{T}_{22}^*}{\partial \theta} \mathbf{T}_{22}^{*-1} \mathbf{T}_{21}^* - \mathbf{T}_{12}^* \mathbf{T}_{22}^{*-1} \frac{\partial \mathbf{T}_{21}^*}{\partial \theta} \right) \mathbf{a}_0^{(l)}, \end{aligned} \quad (\text{B1})$$

where in the last line we have used the property of derivative of matrix inverse. Namely, for a matrix \mathbf{R} , we have

$$\frac{\partial \mathbf{R}^{-1}}{\partial \theta} = -\mathbf{R}^{-1} \frac{\partial \mathbf{R}}{\partial \theta} \mathbf{R}^{-1}. \quad (\text{B2})$$

Due to the block matrix definition,

$$\mathbf{T}^* = \begin{bmatrix} \mathbf{T}_{11}^* & \mathbf{T}_{12}^* \\ \mathbf{T}_{21}^* & \mathbf{T}_{22}^* \end{bmatrix}, \quad (\text{B3})$$

we also have

$$\frac{\partial \mathbf{T}^*}{\partial \theta} = \begin{bmatrix} \frac{\partial \mathbf{T}_{11}^*}{\partial \theta} & \frac{\partial \mathbf{T}_{12}^*}{\partial \theta} \\ \frac{\partial \mathbf{T}_{21}^*}{\partial \theta} & \frac{\partial \mathbf{T}_{22}^*}{\partial \theta} \end{bmatrix}. \quad (\text{B4})$$

In other words, if we know \mathbf{T}^* and $\frac{\partial \mathbf{T}^*}{\partial \theta}$, then the derivative $\frac{\partial \mathbf{a}_M^{(l)}}{\partial \theta}$ is known. \mathbf{T}^* is straightforward, as depicted in Eq. (18).

As an example, if the θ we consider corresponds to the vertical TBU at Row 1 and Column 0, then only \mathbf{T}^0 depends on this θ , so that we have

$$\frac{\partial \mathbf{T}^*}{\partial \theta} = \mathbf{P}^T \mathbf{T}^{M-1} \dots \mathbf{T}^1 \frac{\partial \mathbf{T}^0}{\partial \theta} \mathbf{P}. \quad (\text{B5})$$

According to Eq. (13) in the main text, we know

$$\mathbf{T}^0 = \text{Diag}(\overbrace{\mathbf{H}, \dots, \mathbf{H}}^{N+1}) \text{Diag} \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \underbrace{\mathbf{V}, \dots, \mathbf{V}}_N, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right), \quad (\text{B6})$$

and that the considered θ of the vertical TBU at Row 1 and Column 0 occurs in the first \mathbf{V} matrix inside $\text{Diag}(\cdot)$. Thus, we have

$$\frac{\partial \mathbf{T}^0}{\partial \theta} = \text{Diag}(\overbrace{\mathbf{H}, \dots, \mathbf{H}}^{N+1}) \text{Diag} \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \underbrace{\frac{\partial \mathbf{V}}{\partial \theta}, \mathbf{0}, \dots, \mathbf{0}}_N, \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \right). \quad (\text{B7})$$

Since $\partial \mathbf{V} / \partial \theta$ is straightforward using Eqs. (7) and (4) in the main text, we can calculate $\partial \mathbf{a}_M^{(l)} / \partial \theta$ by using Eqs. (B7), (B5), (B4), and (B1) in sequence. With the derivative $\partial \mathbf{a}_M^{(l)} / \partial \theta$ available, it is easy to calculate the derivative of Cost defined in Eq. (28) in the main text with respect to θ ,

$$\frac{\partial \text{Cost}}{\partial \theta} = 2 \sum_{k=1}^{N_{\text{grid}}} \sum_{n=1}^N [a_{2n,M}^{(l)}(\omega_k) - U_n(\omega_k)] \frac{\partial a_{2n,M}^{(l)}(\omega_k)}{\partial \theta}. \quad (\text{B8})$$

Here $\frac{\partial a_{2n,M}^{(l)}(\omega_k)}{\partial \theta}$ is the entry at index $2n$ of $\frac{\partial \mathbf{a}_M^{(l)}}{\partial \theta}$.

APPENDIX C: COLOR CELL ORDER IN HEAT MAP

We replot Figs. 6(c) and 6(e) in Fig. 15 below to demonstrate how we draw the heat map. Starting from the top left corner of the heat map with a row-first order, the color cells represent θ of vertical TBUs (cells with gray number annotations 1 through 25), then ϕ of vertical TBUs (cells with annotations 26 through 50), then θ of horizontal TBUs (cells annotated from 51 to 80), and finally ϕ of horizontal TBUs (cells annotated from 81 to 110).

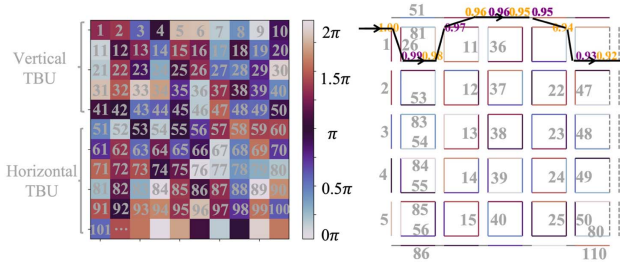


Fig. 15. Demonstration of how we plot the heat map, using Figs. 6(c) and 6(e) as an example.

APPENDIX D: POWER COUPLING AND COMMON PS

Recall the compact model of a TBU as shown in Eq. (A3). If we only focus on the first several terms, we have

$$\begin{aligned}
 & 0.5 \times \begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix} \begin{bmatrix} e^{-j\theta} & 0 \\ 0 & e^{-j\phi} \end{bmatrix} \begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix} \\
 &= 0.5 \begin{bmatrix} e^{-j\theta} - e^{-j\phi} & -je^{-j\theta} - je^{-j\phi} \\ -je^{-j\theta} - je^{-j\phi} & -e^{-j\theta} + e^{-j\phi} \end{bmatrix} \\
 &= 0.5e^{-j\phi} \begin{bmatrix} e^{-j(\theta-\phi)} - 1 & -j[e^{-j(\theta-\phi)} + 1] \\ -j[e^{-j(\theta-\phi)} + 1] & -e^{-j(\theta-\phi)} + 1 \end{bmatrix} \\
 &= e^{-j\phi} \begin{bmatrix} j \sin \frac{\alpha}{2} e^{j\frac{\alpha}{2}} & -j \cos \frac{\alpha}{2} e^{j\frac{\alpha}{2}} \\ -j \cos \frac{\alpha}{2} e^{j\frac{\alpha}{2}} & -j \sin \frac{\alpha}{2} e^{j\frac{\alpha}{2}} \end{bmatrix} \\
 &= je^{-j\phi} e^{j\frac{\alpha}{2}} \begin{bmatrix} \sin \frac{\alpha}{2} & -\cos \frac{\alpha}{2} \\ -\cos \frac{\alpha}{2} & -\sin \frac{\alpha}{2} \end{bmatrix} \\
 &= \underbrace{e^{j(\frac{\alpha}{2} - \phi + \frac{\alpha}{2})}}_{\text{Common phase shift}} \times \underbrace{\begin{bmatrix} \sin \frac{\alpha}{2} & -\cos \frac{\alpha}{2} \\ -\cos \frac{\alpha}{2} & -\sin \frac{\alpha}{2} \end{bmatrix}}_{\text{Power coupling matrix}}, \quad (\text{D1})
 \end{aligned}$$

where for simplicity, we have denoted $\alpha = -(\theta - \phi)$, and used the following equations to do the simplification:

$$\begin{aligned}
 0.5(e^{j\alpha} - 1) &= j \sin \frac{\alpha}{2} e^{j\frac{\alpha}{2}}, \\
 0.5(e^{j\alpha} + 1) &= \cos \frac{\alpha}{2} e^{j\frac{\alpha}{2}}. \quad (\text{D2})
 \end{aligned}$$

The power coupling ratio is represented by $(\cos \frac{\alpha}{2})^2 = \cos^2 \frac{\phi - \theta}{2}$. This is shown by the shaded gray area in Fig. 6(f) of the main text. Based on Eq. (D1), we define $\frac{\alpha}{2} - \phi + \frac{\alpha}{2} = \frac{\alpha}{2} - \frac{\phi + \theta}{2}$ as the common PS and mark it using colors in the background of Fig. 6(f).

APPENDIX E: CONSIDERING THERMAL CROSS TALK AND OTHER NONIDEALITIES

Recall that in the main text, we solve Eq. (28) to obtain a configuration for a desired light-processing function. In one important real-world scenario (thermally controlled phase shifters), the PPIC delivers power to heaters at the phase shifters. However, this process involves nonideal thermal cross talk. This can be mathematically represented by [29]

$$\mathbf{h}(\Phi \mathbf{p}) = \mathbf{x}, \quad (\text{E1})$$

where Φ is the thermal coefficient matrix, usually with ones on the diagonals and small off-diagonal entries (i.e., diagonal-dominant), \mathbf{p} denotes the delivered power by the heaters at phase shifters, \mathbf{h} represents the function mapping from power to PS, and \mathbf{x} represents all PSs. As an example, an ideal case without thermal cross talk corresponds to $\Phi = \mathbf{I}$.

With the help of Eq. (E1), our proposed method remains applicable when considering thermal cross talk. Specifically, we can now optimize with respect to \mathbf{p} instead of \mathbf{x} ,

$$\begin{aligned}
 \min_{\mathbf{p}} \text{Cost} &= \sum_{k=1}^{N_{\text{grid}}} \sum_{n=1}^N |a_{2n,M}^{(I)}(\omega_k, \mathbf{x}) - U_n(\omega_k)|^2 \\
 \text{s.t.}, \quad \mathbf{h}(\Phi \mathbf{p}) &= \mathbf{x}, \quad (\text{E2})
 \end{aligned}$$

or, more densely,

$$\min_{\mathbf{p}} \text{Cost} = \sum_{k=1}^{N_{\text{grid}}} \sum_{n=1}^N |a_{2n,M}^{(I)}[\omega_k, \mathbf{h}(\Phi \mathbf{p})] - U_n(\omega_k)|^2. \quad (\text{E3})$$

Note that \mathbf{h} and Φ are known (or could be characterized) once a PPIC is fabricated. Thus, the above optimization is well defined with respect to \mathbf{p} . Since the extra involved operations, multiplying by Φ and mapping by \mathbf{h} , are differentiable, we can still obtain analytical gradients and adopt gradient descent optimization to solve Eq. (E3). A similar treatment can be adopted for other nonidealities.

APPENDIX F: THE COMPACT MODEL UNDER NONIDEALITY

Our compact model, shown in Eqs. (2) and (4), assumes that the DCs achieve 50%:50% splitting; however, this is not the case in many real-world scenarios. The proposed compact model should be refined if nonidealities (e.g., process variation, nonideal beam splitting) are important and need to be considered. Here we show how to refine the compact model for an example nonideality,

$$\mathbf{F} = \frac{1}{2} \cdot \mathbf{M}(\eta_1) \cdot \begin{bmatrix} e^{-j\theta} & 0 \\ 0 & e^{-j\phi} \end{bmatrix} \cdot \mathbf{M}(\eta_2) \cdot \alpha e^{-j\omega \frac{n_{\text{eff}} L}{c}}, \quad (\text{F1})$$

where $\{\eta_1, \eta_2\}$ represent the deviation of the DC from 50%:50%, and $\mathbf{M}(\eta)$ is defined as [9]

$$\mathbf{M}(\eta) = \begin{bmatrix} \cos(\pi/4 + \eta) & -j \sin(\pi/4 + \eta) \\ -j \sin(\pi/4 + \eta) & \cos(\pi/4 + \eta) \end{bmatrix}. \quad (\text{F2})$$

Note that $\{\eta_1, \eta_2, \alpha, L\}$ should be treated as random variables due to process variation. A commonly used assumption is that these random variables can be decomposed as the summation of deterministic variables and perturbation variables,

$$\begin{aligned}
 \eta_1 &= \eta_1^* + \Delta\eta_1, \\
 \eta_2 &= \eta_2^* + \Delta\eta_2, \\
 \alpha &= \alpha^* + \Delta\alpha, \\
 L &= L^* + \Delta L, \quad (\text{F3})
 \end{aligned}$$

where $\{\eta_1^*, \eta_2^*, \alpha^*, L^*\}$ denote the deterministic nominal design values, and $\{\Delta\eta_1, \Delta\eta_2, \Delta\alpha, \Delta L\}$ represent the probabilistic perturbation introduced in manufacturing. For different TBUs,

they share the values of $\{\eta_1^*, \eta_2^*, \alpha^*, L^*\}$, but their $\{\Delta\eta_1, \Delta\eta_2, \Delta\alpha, \Delta L\}$ values are different. Moreover, we can further combine Eq. (F1) with Eq. (A1) to generate a more comprehensive compact model spanning all waveguides in the TBU.

For any given instantiation of process variations (e.g., using random sampling), the compact model remains differentiable, and thus our proposed method can be used to generate a solution for that instance. Evaluation of performance degradation or yield (e.g., using the Monte Carlo method), becomes possible. Future research might consider the robust synthesis problem, building on the method proposed here.

Funding. European Research Council (725555).

Acknowledgment. Xiangfeng Chen and Wim Bogaerts received funding from the ERC.

Disclosures. The authors declare that there are no conflicts of interest related to this article.

Data Availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

REFERENCES

- L. Chrostowski and M. Hochberg, *Silicon Photonics Design: From Devices to Systems* (Cambridge University, 2015).
- W. Bogaerts and L. Chrostowski, "Silicon photonics circuit design: methods, tools and challenges," *Laser Photon. Rev.* **12**, 1700237 (2018).
- W. Bogaerts, M. Fiers, and P. Dumon, "Design challenges in silicon photonics," *IEEE J. Sel. Top. Quantum Electron.* **20**, 8202008 (2013).
- W. Bogaerts, D. Pérez, J. Capmany, D. A. Miller, J. Poon, D. Englund, F. Morichetti, and A. Melloni, "Programmable photonic circuits," *Nature* **586**, 207–216 (2020).
- Z. Gao, X. Chen, Z. Zhang, U. Chakraborty, W. Bogaerts, and D. S. Boning, "Automatic realization of light processing functions for programmable photonics," in *IEEE Photonics Conference* (IEEE, 2022), pp. 1–2.
- D. A. B. Miller, "Self-aligning universal beam coupler," *Opt. Express* **21**, 6360–6370 (2013).
- D. A. Miller, "Self-configuring universal linear optical component," *Photon. Res.* **1**, 1–15 (2013).
- C. Taballione, T. A. Wolterink, J. Lugani, A. Eckstein, B. A. Bell, R. Grootjans, I. Visscher, J. J. Renema, D. Geskus, C. G. Roeloffzen, and I. A. Walmsley, "8 × 8 programmable quantum photonic processor based on silicon nitride waveguides," in *Frontiers in Optics* (Optical Society of America, 2018), paper JTU3A.58.
- S. Bandyopadhyay, R. Hamerly, and D. Englund, "Hardware error correction for programmable photonics," *Optica* **8**, 1247–1255 (2021).
- W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley, "Optimal design for universal multiport interferometers," *Optica* **3**, 1460–1465 (2016).
- K. Jinguji and T. Yasui, "Synthesis of one-input m -output optical FIR lattice circuits," *J. Lightwave Technol.* **26**, 853–866 (2008).
- D. Pérez, I. Gasulla, L. Crudgington, D. J. Thomson, A. Z. Khokhar, K. Li, W. Cao, G. Z. Mashanovich, and J. Capmany, "Multipurpose silicon photonics signal processor core," *Nat. Commun.* **8**, 1 (2017).
- X. Chen, P. Stroobant, M. Pickavet, and W. Bogaerts, "Graph representations for programmable photonic circuits," *J. Lightwave Technol.* **38**, 4009–4018 (2020).
- A. López, D. Pérez, P. DasMahapatra, and J. Capmany, "Auto-routing algorithm for field-programmable photonic gate arrays," *Opt. Express* **28**, 737–752 (2020).
- L. Zhuang, C. G. Roeloffzen, M. Hoekman, K.-J. Boller, and A. J. Lowery, "Programmable photonic signal processor chip for radiofrequency applications," *Optica* **2**, 854–859 (2015).
- Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljačić, "Deep learning with coherent nanophotonic circuits," *Nat. Photonics* **11**, 441–446 (2017).
- L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins, and A. E. Lita, "Quantum computational advantage with a programmable photonic processor," *Nature* **606**, 75–81 (2022).
- J. M. Arrazola, V. Bergholm, K. Brádler, T. R. Bromley, M. J. Collins, I. Dhand, A. Fumagalli, T. Gerrits, A. Goussev, L. G. Helt, and J. Hundal, "Quantum circuits with many photons on a programmable nanophotonic chip," *Nature* **591**, 54–60 (2021).
- S. Pai, B. Bartlett, O. Solgaard, and D. A. B. Miller, "Matrix optimization on universal unitary photonic devices," *Phys. Rev. Appl.* **11**, 064044 (2019).
- S. Bandyopadhyay, A. Sludds, S. Krastanov, R. Hamerly, N. Harris, D. Bunandar, M. Streshinsky, M. Hochberg, and D. Englund, "Single chip photonic deep neural network with accelerated training," *arXiv*, arXiv:2208.01623 (2022).
- S. Pai, Z. Sun, T. W. Hughes, T. Park, B. Bartlett, I. A. D. Williamson, M. Minkov, M. Milanizadeh, N. Abebe, F. Morichetti, A. Melloni, S. Fan, O. Solgaard, and D. A. B. Miller, "Experimentally realized *in situ* backpropagation for deep learning in nanophotonic neural networks," *arXiv*, arXiv:2205.08501 (2022).
- M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental realization of any discrete unitary operator," *Phys. Rev. Lett.* **73**, 58–61 (1994).
- J. Capmany, I. Gasulla, and D. Pérez, "The programmable processor," *Nat. Photonics* **10**, 6–8 (2016).
- D. Pérez and J. Capmany, "Scalable analysis for arbitrary photonic integrated waveguide meshes," *Optica* **6**, 19–27 (2019).
- D. Pérez, I. Gasulla, J. Capmany, and R. A. Soref, "Reconfigurable lattice mesh designs for programmable photonic processors," *Opt. Express* **24**, 12093–12106 (2016).
- D. Pérez-López, A. López, P. DasMahapatra, and J. Capmany, "Multipurpose self-configuration of programmable photonic circuits," *Nat. Commun.* **11**, 1 (2020).
- W. Bogaerts, P. De Heyn, T. Van Vaerenbergh, K. De Vos, S. Kumar Selvaraja, T. Claes, P. Dumon, P. Bienstman, D. Van Thourhout, and R. Baets, "Silicon microring resonators," *Laser Photon. Rev.* **6**, 47–73 (2012).
- J. Heebner, R. Grover, and T. Ibrahim, *Optical Microresonator Theory* (Springer, 2008).
- M. Milanizadeh, D. Aguiar, A. Melloni, and F. Morichetti, "Canceling thermal cross-talk effects in photonic integrated circuits," *J. Lightwave Technol.* **37**, 1325–1332 (2019).
- M. Wang, X. Chen, U. Khan, and W. Bogaerts, "Programmable wavelength filter with double ring loaded MZI," *Sci. Rep.* **12**, 1 (2022).