Gradient-Based Power Efficient Functional Synthesis for Programmable Photonic Circuits

Zhengqi Gao, *Graduate Student Member, IEEE*, Xiangfeng Chen, Zhengxing Zhang, Uttara Chakraborty, *Graduate Student Member, IEEE*, Wim Bogaerts, *Fellow, IEEE*, and Duane S. Boning, *Fellow, IEEE*

***Abstract*—We propose an automatic approach for implementing light processing functions on programmable photonic integrated circuits. Our approach offers two unprecedented and significant advantages. Firstly, it is generalizable to any topology, including triangular, square, and hexagonal meshes, and any linear light processing function with a frequency domain representation. Secondly, it achieves power efficiency by minimizing power consumption, even when taking into account non-ideal thermal crosstalk. Our key is to employ a differentiable scattering matrix simulation and formulate functional synthesis as an L1-regularized optimization problem. Our proposed approach demonstrates superior power efficiency and synthesized result quality compared to several common baselines, with a slight and manageable increase in algorithm runtime.**

***Index Terms*—Power efficiency, functional synthesis, programmable photonic integrated circuit.**

I. Introduction

Silicon photonic circuit design traditionally entails crafting a photonic circuit tailored to a specific application [1], [2], necessitating a full design restart when new customer requirements arise. Programmable photonic integrated circuits (PPICs) have emerged as a promising alternative design paradigm that leverages reconfigurability to avoid such redundancy [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. Analogous to the principles underpinning modern field-programmable gate arrays (FPGAs), users can program PPICs' configurable blocks with software to achieve desired optical functions in run-time. These configurable building blocks are typically implemented as active Mach-Zehnder interferometer (MZI) structures, and are known as *tunable basic units* (TBUs) in the literature [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. More specifically, each TBU comprises two actively controlled optical tuners, such as thermal/electro-optic phase shifters, whose properties can be dynamically altered via electric signals.

Depending on the interconnectivity of TBUs, PPIC can be broadly classified into pure feedforward topology [13], [14], [15], [16], [17], [18] and recirculating topology [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [19]. Special feedforward topologies, such as Reck's [17] and Clements's [14], have been shown to possess the capability of realizing any unitary matrix. Furthermore, the advantages of optical computing, such as high frequency and low power consumption, make feedforward PPIC architecture an ideal solution for accelerating matrix-vector multiplication, a frequently occurring operation in modern deep learning. Consequently, numerous studies have explored the use of feedforward PPICs as hardware accelerators for deep learning, which are commonly referred to as optical neural networks in the literature [13], [14], [15], [16], [17]. In contrast, recirculating PPICs (e.g., triangular, square, and hexagonal mesh) feature loop-back connections, which allow light to propagate in various directions throughout the circuit. Therefore, they are primarily used in optical applications [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. As a concrete example, by utilizing the loop-backs in a recirculating PPIC, optical signals can circulate indefinitely, enabling the creation of resonant modes in a ring resonator or the realization of feedback in an infinite impulse response (IIR) filter. However, implementing an optical ring resonator or an IIR filter is not possible with a feedforward PPIC due to the absence of loops.

Along with the versatility advantage brought by the loop-backs in a recirculating PPIC, configuring an optical function (e.g., routing, splitting, filtering) on a recirculating PPIC becomes complicated, and is usually performed manually by experts. In the literature, some works investigate automatic approaches to resolve this functional synthesis task. For example, particle swarm optimization has been employed to design optical ring resonators and Mach-Zehnder interferometers on hexagonal-mesh PPICs, as discussed in [6]. Others have proposed an auto-routing method based on graph theory for a hexagonal-mesh PPIC, while multi-objective routing has also been demonstrated in another study in [11]. The authors in [10], [12] have proposed deriving the system-level transfer function analytically using mathematical induction, allowing analytical gradients to be obtained, and gradient descent to be applied to

Manuscript received 25 December 2023; revised 14 April 2024 and 3 May 2024; accepted 10 May 2024. Date of publication 14 May 2024; date of current version 2 September 2024. The work of Xiangfeng Chen and Wim Bogaerts was supported by European Research Council under Grant 725555 (PhotonicSWARM). *(Corresponding author: Zhengqi Gao.)*

Zhengqi Gao, Zhengxing Zhang, Uttara Chakraborty, and Duane S. Boning are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: zhengqi@mit.edu; zhxzhang@mit.edu; uttara@mit.edu; boning@mit.edu).

Xiangfeng Chen and Wim Bogaerts are with the Photonics Research Group, Department of Information Technology, Ghent University-IMEC, 9052 Ghent, Belgium (e-mail: xiangfeng.chen@ugent.be; wim.bogaerts@ugent.be).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JLT.2024.3400942.

Digital Object Identifier 10.1109/JLT.2024.3400942

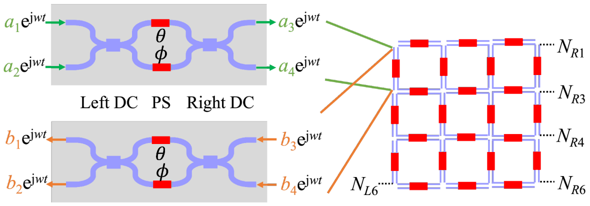


Fig. 1. Left: A simplified schematic for a TBU. The green and orange color indicates that TBU is a bi-directional and reciprocal device. Right: A 3-by-3 square-mesh recirculating PPIC comprises TBUs placed vertically and horizontally. We name the nodes at the left side and the right side from top to bottom using N_{Lx} and N_{Rx}, respectively, where $x = 1, 2, \ldots, 6$.

realize linear light-processing functions on a square mesh. A similar idea was utilized by Perez et al. [20] for a hexagonal mesh. However, these methods have their limitations, as most of them are specific to a single topology, such as the square mesh in Gao et al. [10], [12] and the hexagonal mesh in Perez et al. [6], [20]. Despite these advances, the approaches tend to be topology-specific and are not power-aware, often overlooking the energy consumption associated with controlling tuners on TBUs.

In this paper, we present a unified approach for functional synthesis that can be applied to various topologies, such as triangular, square, hexagonal, and even their mixtures. To begin, we adopt a differentiable frequency-domain scattering matrix representation. Next, we observe and justify that L1-regularized gradient descent is naturally the optimal choice for automating power-efficient functional synthesis based on the thermal cross-talk model in [21]. Finally, we utilize the proposed method to synthesize a few common linear light-processing functions (e.g., routing, splitting, filtering, wavelength division multiplexing) on various recirculating meshes. It surpasses several common baselines in both power efficiency and synthesized result quality, with a slight and manageable increase in algorithm run-time. One of our contributions is to mathematically justify that it is sufficient to consider the functional synthesis problem in the phase space, as long as the thermal cross-talk effect is reasonably small.

II. Differentiable S-Matrix Simulation

As shown in Fig. 1, a tunable basic unit (TBU) consists of two directional couplers (DCs) with splitting ratios ideally both equal to 50% and two active optical phase shifters (PSs). The phase shifts (PSs) are parameterized by θ and ϕ and their values could be controlled varying in the full $[0, 2\pi]$ range when imposing thermal/electric signals. In our paper, we adopt the following scattering matrix model for a TBU [10], [12]:

$$\mathbf{F} = \frac{\alpha}{2}\begin{bmatrix} \mathrm{e}^{-\mathrm{j}\theta} - \mathrm{e}^{-\mathrm{j}\phi} & -\mathrm{j}\mathrm{e}^{-\mathrm{j}\theta} - \mathrm{j}\mathrm{e}^{-\mathrm{j}\phi} \\ -\mathrm{j}\mathrm{e}^{-\mathrm{j}\theta} - \mathrm{j}\mathrm{e}^{-\mathrm{j}\phi} & -\mathrm{e}^{-\mathrm{j}\theta} + \mathrm{e}^{-\mathrm{j}\phi} \end{bmatrix} \mathrm{e}^{-\mathrm{j}\omega\frac{n_{\text{eff}}L}{c}} \tag{1}$$

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. We emphasize that θ and ϕ are the design variables that we could adjust in run-time. In the right of Fig. 1, we show a 3-by-3 square-mesh PPIC made up of a few TBUs.

When a pair of time-harmonic optical signals with complex coefficients $\{a_1, a_2\}$ are injected at the left nodes of a TBU as shown in Fig. 1, the output responses characterized by $\{a_3, a_4\}$ could be calculated by the scattering matrix. As TBU is bi-directional and reciprocal, similar relations exist between $\{b_1, b_2\}$ and $\{b_3, b_4\}$. These equations are summarized below:

$$\begin{bmatrix} a_3 \\ a_4 \end{bmatrix} = \mathbf{F}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \mathbf{F}\begin{bmatrix} b_3 \\ b_4 \end{bmatrix} \tag{2}$$

It is important to note that time-harmonic chronomatic optical signals possess a single frequency, leading to a common assumption that the signals can be represented in the frequency domain by a dependence on $\mathrm{e}^{\mathrm{j}\omega t}$ or $\mathrm{e}^{-\mathrm{j}\omega t}$. Thus, the complex coefficient preceding either $\mathrm{e}^{\mathrm{j}\omega t}$ or $\mathrm{e}^{-\mathrm{j}\omega t}$ (such as $\{a_i, b_i\}_{i=1}^{4}$ in our example) completely characterizes the optical signal's magnitude and phase. In our notation, we opt for the use of $\mathrm{e}^{\mathrm{j}\omega t}$ instead of $\mathrm{e}^{-\mathrm{j}\omega t}$; nevertheless, both choices yield equivalent results in calculations.

Frequency-domain simulation at various levels (e.g., FDFD, S-matrix simulation, Fourier optics) has been widely used in the optics and photonics domain in past works [22], [23], [24]. In our paper, we utilize the frequency-domain S-matrix simulation with a customized differentiable engine. Let us consider a toy example of connecting two TBUs in series as shown in Fig. 2. This toy circuit has six nodes, and each node is accompanied by two complex scalars (marked in green and orange), representing the circuit responses when excited by input sources, due to the bi-directional propagation of light. We can denote these scalars collectively with a column vector $\mathbf{x} = [a_1, \ldots, a_6, b_1, \ldots, b_6]^T$, which is unknown and needs to be solved. In the following, we will show that a matrix equation $\mathbf{A}\mathbf{x} = \mathbf{z}$ could be built, where $\mathbf{A}$ is a known 12-by-12 complex matrix and $\mathbf{z}$ is a known 12-dimensional complex column vector of a specific structure, so that $\mathbf{x}$ could be solved using matrix inversion $\mathbf{x} = \mathbf{A}^{-1}\mathbf{z}$.

Specifically, we can write down two matrix equations for TBU1 with scattering matrix $\mathbf{F}^1$ by specializing (2) to it. When expanding out the terms of these two matrix equations and moving all terms to the left-hand side (LHS), we obtain four linear equality constraints with right-hand sides (RHS) set to zero. These four LHS terms can be expressed with the first four rows of $\mathbf{A}$, and the four RHS terms can be expressed with the first four rows of $\mathbf{z}$ (set to zeros). Similarly, TBU2 with scattering matrix $\mathbf{F}^2$ imposes another four linear equality constraints about $\mathbf{x}$ and can be included in the second four rows of $\mathbf{A}$ and $\mathbf{z}$. Finally, to ensure the circuit has a physical meaning and can be solved, we need to specify the input sources that excite the circuit. Namely, we need to define a complex vector $\mathbf{u}$ (i.e., input sources) to set the values of $\{a_1, a_2, b_5, b_6\}$. Again, this could be regarded as four linear constraints about $\mathbf{x}$, and adopted into the last four rows of $\mathbf{A}$ and $\mathbf{z}$. Note that $\mathbf{A}$ must be invertible because this is a physical system and we know there is a unique circuit response $\mathbf{x}$.

In essence, the depicted approach converts all S-matrix relations and the source input requirement into a system of linear equalities, which could be further used to solve the circuit. For

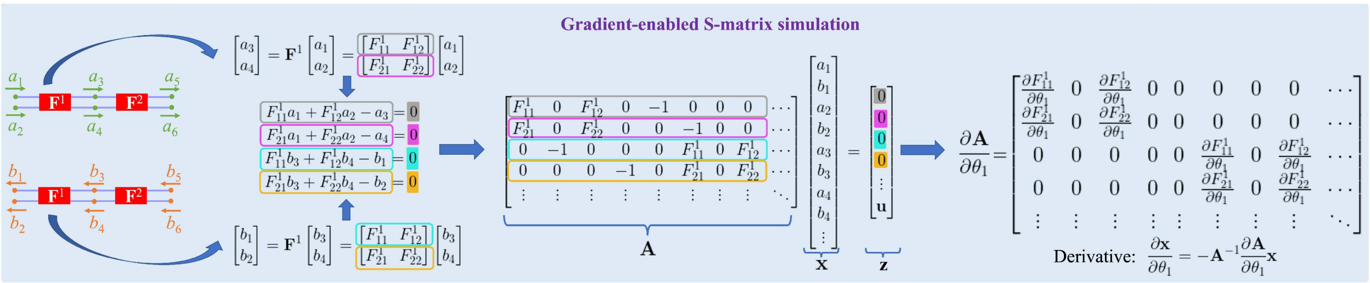

Gradient-enabled S-matrix simulation
a_1 a_3 a_5 a_2 a_4 a_6
$\mathbf{F}^1$ $\mathbf{F}^2$
b_1 b_3 b_5 b_2 b_4 b_6
$\begin{bmatrix}a_3\\a_4\end{bmatrix} = \mathbf{F}^1\begin{bmatrix}a_1\\a_2\end{bmatrix} = \begin{bmatrix}F^1_{11} & F^1_{12}\\F^1_{21} & F^1_{22}\end{bmatrix}\begin{bmatrix}a_1\\a_2\end{bmatrix}$
$F^1_{11}a_1 + F^1_{12}a_2 - a_3 = 0$
$F^1_{21}a_1 + F^1_{22}a_2 - a_4 = 0$
$F^1_{11}b_3 + F^1_{12}b_4 - b_1 = 0$
$F^1_{21}b_3 + F^1_{22}b_4 - b_2 = 0$
$\begin{bmatrix}b_1\\b_2\end{bmatrix} = \mathbf{F}^1\begin{bmatrix}b_3\\b_4\end{bmatrix} = \begin{bmatrix}F^1_{11} & F^1_{12}\\F^1_{21} & F^1_{22}\end{bmatrix}\begin{bmatrix}b_3\\b_4\end{bmatrix}$
A
x
z
u
$\frac{\partial \mathbf{A}}{\partial \theta_1}$
Derivative: $\frac{\partial \mathbf{x}}{\partial \theta_1} = -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial \theta_1}\mathbf{x}$

Fig. 2. Illustration of the gradient-enabled frequency-domain S-matrix simulation with a toy example.

demonstration purposes, we only use a toy circuit as an example in Fig. 2, but it obviously could be generalized to any PPIC of any topology (e.g., triangular, hexagonal, or even mixture). Moreover, it even works for a general integrated photonic circuit. But we will have different forms of scattering matrices for different types of devices in that case.

Now, we explain our proposed method to obtain the derivatives with a little computational overhead. To begin with, we denote some phase shifter in a PPIC by p and consider a small perturbation δp to its value. For instance, in Fig. 2, we show $p = \theta_1$, and δp represents a small perturbation to θ_1 in the toy circuit. Recall the setup procedure of the system equation $\mathbf{A}\mathbf{x} = \mathbf{z}$, we notice that δp will impact $\mathbf{A}$ and $\mathbf{x}$ and shift their values to $\mathbf{A}' = \mathbf{A} + \delta\mathbf{A}$ and $\mathbf{x}' = \mathbf{x} + \delta\mathbf{x}$. Then we have the following two equations for the unperturbed and perturbed circuit, respectively:

$$\mathbf{A}\mathbf{x} = \mathbf{z}, \quad (\mathbf{A} + \delta\mathbf{A})(\mathbf{x} + \delta\mathbf{x}) = \mathbf{z} \tag{3}$$

Subtracting the first equation from the second one, and omitting the second-order term, we obtain $\delta\mathbf{A}\cdot\mathbf{x} + \mathbf{A}\cdot\delta\mathbf{x} = \mathbf{0}$. After further rearranging, we obtain $\mathbf{x} = -\mathbf{A}^{-1}\delta\mathbf{A}\mathbf{x}$. Finally, we divide both sides by δp and let $\delta p \to 0$, yielding:

$$\frac{\partial \mathbf{x}}{\partial p} = -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial p}\mathbf{x} \tag{4}$$

Note that we derive the above equation using first-order perturbation; the same formula can also be obtained through the adjoint method as widely used in photonics [19], [25], [26]. We emphasize that during the original S-matrix simulation (i.e., solving $\mathbf{A}\mathbf{x} = \mathbf{z}$), $\mathbf{A}^{-1}$ could be stored, so that only $\partial\mathbf{A}/\partial p$ is unknown in (4). However, building $\partial\mathbf{A}/\partial p$ is fairly easy, and only eight entries of the matrix $\partial\mathbf{A}/\partial p$ are non-zero, as exemplified in the right part of Fig. 2. Thus, the evaluation of (4) has little computational cost.

In our case, we are actually concerned about the derivatives of $\mathbf{x}$ with respect to a vector $\mathbf{p}$ representing all phase shifts (i.e., the Jacobian matrix $\partial\mathbf{x}/\partial\mathbf{p}$) instead of merely a scalar p. For instance, in the toy example, we need $\mathbf{p} = [\theta_1, \phi_1, \theta_2, \phi_2]^T$. Nevertheless, this extra complexity doesn't impact the computation. A trivial method is to go through each entry of $\mathbf{p}$ one by one repeatedly using (4), while advanced techniques could utilize batched matrix multiplication (such as the matmul function in Numpy [27]). We postpone the reason for needing this Jacobian matrix $\partial\mathbf{x}/\partial\mathbf{p}$ and the overall functional synthesis algorithm to the next section.

III. Thermal Crosstalk Model and L1 Penalty

As detailed in the review [9], nowadays most PPICs use electrically driven heaters to induce a thermal-optic phase shift. Mathematically, it means that the phase shift of the active tuner (e.g., $\{\theta_1, \phi_1, \theta_2, \phi_2\}$ in Fig. 2) is governed by a characteristic function: $p = f(q)$, where $q \geq 0$ represents the delivered thermal power to a phase shifter, and p represents the phase shift as before. In our paper, for convenience purposes, we assume $f(0) = 0$. Namely, there is no phase shift when not imposing power onto a phase shifter. [1] As an example, from the characterization of our recent fabricated chip, $f(\cdot)$ is a linear function passing the origin with slope 0.06π/mW. In other words, imposing 1.0 mW power on the phases shifter, the resulting phase shift is 0.06π. Note that imposing $(100/3 + 1.0)$mW power yields phase shift 2.06π, but from a numerical perspective, this is indistinguishable from 0.06π. Thus, without loss of generality, we constrain the range of $f(\cdot)$ to $[0, 2\pi]$ (i.e., $p \in [0, 2\pi]$). Moreover, it is reasonable to assume that $f(\cdot)$ is monotonically increasing (hence invertible), as imposing more thermal power won't reduce the phase accumulation. Thus, the domain of $f(\cdot)$ will also be bounded, i.e., $q \in [0, q_{\max}]$ with $2\pi = f(q_{\max})$. In a general case, the function $f(\cdot)$ might be nonlinear.

Note that there are multiple phase shifters $\mathbf{p}$ in a PPIC and ideally each of them obeys the characteristic function $p_i = f(q_i)$ with no correlation, where p_i is the i-th entry of $\mathbf{p}$ and q_i is the associated thermal power. However, in reality, the delivered thermal power q_i not only impacts p_i, but will also have a minor effect to p_j (where $j \neq i$). This phenomenon is known as the thermal crosstalk effect. To take it into consideration, we could use the vectorial version of the characteristic equation [21]:

$$\mathbf{p} = f(\mathbf{T}\mathbf{q}) \tag{5}$$

where $\mathbf{T}$ represents the thermal crosstalk coefficient matrix. It has ones on the diagonal, and all off-diagonals are real numbers close to zero. Ideally, when there is no thermal crosstalk effect, $\mathbf{T}$ reduces to an identity matrix.

[1] In a real-world scenario, compensation is needed so that we could assume so, which could be viewed as shifting the root of $f(\cdot)$. See Appendix A for detailed discussion.

Under this framework, our task is to realize desired light-processing functions (such as routing, splitting, and filtering) onto a PPIC with as little thermal power as possible. To begin with, assume we want to realize N light-processing functions represented by the complex transfer functions $\{U_n(\omega)|n = 1, 2, \ldots, N\}$ specifying the magnitude and phase responses in a range $[\omega_{\text{min}}, \omega_{\text{max}}]$ onto a given PPIC. We choose N_{grid} frequency points $\{\omega_1 = \omega_{\text{min}}, \omega_2 = \omega_{\text{min}} + \Delta\omega, \ldots, \omega_{N_{\text{grid}}} = \omega_{\text{max}}\}$ in this desired angular frequency range with incremental step equal to $\Delta\omega$. Then we define a cost function:

$$Cost(\mathbf{p}) = \sum_{k=1}^{N_{\text{grid}}} \sum_{n=1}^{N} |O_n(\omega_k, \mathbf{p}) - U_n(\omega_k)|^2 \tag{6}$$

where $O_n(\omega_k, \mathbf{p})$ represents the circuit response (a complex scalar) at a pre-selected output port O_n for the n-th light processing function at angular frequency ω_k. Note that we have made it explicit that $O_n(\omega_k, \mathbf{p})$ depends on the variable $\mathbf{p}$. In certain applications, such as power splitting, only the magnitude response is relevant. In such cases, we can define a real scalar $U_n(\omega)$ as the desired magnitude response target. This allows us to discard the phase part of $O_n(\omega_k, \mathbf{p})$ while maintaining the validity of (6). For ease of depiction, we will henceforth use M to denote the dimensions of $\mathbf{p}$ and $\mathbf{q}$. To perform power-efficient functional synthesis, we can solve the following multi-objective optimization:

$$\begin{aligned} &\min_{\mathbf{q}} \quad \left(Cost(\mathbf{p}),\, \mathbf{e}^T\mathbf{q}\right) \\ &s.t., \quad \mathbf{p} = f(\mathbf{Tq}) \quad \text{and} \quad \mathbf{q} \in [0, q_{\text{max}}]^M \end{aligned} \tag{7}$$

where $\mathbf{e}$ represents a vector with entries all equal to one so that $\mathbf{e}^T\mathbf{q}$ represents the consumed thermal power to realize the phase shifts $\mathbf{p}$. Certainly, as it is, the optimization shown in (7) could readily be solved in various ways.

In the following, we show a special simplification of (7) by exploiting the properties of $f(\cdot)$ and $\mathbf{T}$, and then discuss its benefits. If we could substitute the second objective $\mathbf{e}^T\mathbf{q}$ with $\mathbf{e}^T\mathbf{p}$, then both the objectives in (7) depend on $\mathbf{p}$, so (7) could be solved in the space of $\mathbf{p}$:

$$\min_{\mathbf{p}\in[0,2\pi]^M} \quad \left(Cost(\mathbf{p}),\, \mathbf{e}^T\mathbf{p}\right) \tag{8}$$

Namely, after obtaining an optimal solution $\mathbf{p}^\star$ of (8), we could obtain the associated optimal solution $\mathbf{q}^\star$ of (7) by $\mathbf{q}^\star = \mathbf{T}^{-1}f^{-1}(\mathbf{p}^\star)$. To understand why we can substitute $\mathbf{e}^T\mathbf{q}$ with $\mathbf{e}^T\mathbf{p}$ in 7), we first consider the simple case where $\mathbf{T}$ is an identity matrix, meaning there is no thermal crosstalk, and $f(\cdot)$ is a linear function of the form $f(q) = aq + b$. Here, $a > 0$ ensures that $f(\cdot)$ is increasing, and $b = 0$ ensures $f(0) = 0$. Under these conditions, the constraint $\mathbf{p} = f(\mathbf{Tq})$ becomes $\mathbf{p} = a\mathbf{q}$. Multiplying both sides by $\mathbf{e}^T$ reveals that larger values of $\mathbf{e}^T\mathbf{q}$ correspond to larger values of $\mathbf{e}^T\mathbf{p}$, and vice versa. Thus, we can replace $\mathbf{e}^T\mathbf{q}$ with $\mathbf{e}^T\mathbf{p}$. To extend to a general case, we present Theorem 1, whose proof is deferred to Appendix B for interested readers. Roughly, Theorem 1 states that as long as all off-diagonals of the matrix $\mathbf{T}$ are small enough (governed by (12)), then for the $\mathbf{p}^\star$ with the smallest $\mathbf{e}^T\mathbf{p}$ value (among all considered $\mathbf{p}$), its associated $\mathbf{q}^\star$ must also achieve the smallest $\mathbf{e}^T\mathbf{q}$ (among all considered $\mathbf{q}$).[2] Note that an immediate corollary of the theorem is that when there is no thermal crosstalk effect (i.e., $\epsilon = 0$), then $\mathbf{e}^T\mathbf{q}$ and $\mathbf{e}^T\mathbf{p}$ could always be interchangeable, as expected.

Theorem 1: Assume the maximum absolute value of the off-diagonals in $\mathbf{T}$ equals $\epsilon \geq 0$. Assume the derivative of $f(\cdot)$ is bounded in $[0, B]$. Assume that for any pair of constant (c_1, c_2), if there exists some $\mathbf{p}$ such that $(Cost(\mathbf{p}),\, \mathbf{e}^T\mathbf{p}) = (c_1, c_2)$, then this $\mathbf{p}$ must be unique.

Now, for a specific task abstracted as $Cost(\cdot)$ and a given constant a, we denote:

$$\begin{aligned} \Gamma_a &= \{\mathbf{p} \in [0, 2\pi]^M \mid Cost(\mathbf{p}) = a\} \\ \Theta_a &= \{\mathbf{q} \in [0, q_{\text{max}}]^M \mid f(\mathbf{Tq}) = \mathbf{p} \in \Gamma_a\} \end{aligned} \tag{9}$$

where Γ_a represents the set of $\mathbf{p}$'s all achieving the same cost a, and Θ_a represents the corresponding set of power $\mathbf{q}$'s realizing Γ_a. Next, we denote:

$$h_a = \min_{\mathbf{p}\in\Gamma_a} \mathbf{e}^T\mathbf{p} \tag{10}$$

This minimum h_a must be achieved at a unique $\mathbf{p}^\star$ (i.e., $h_a = \mathbf{e}^T\mathbf{p}^\star$). For later simplicity, we denote:

$$h'_a = \min_{\mathbf{p}\in\Gamma_a/\{\mathbf{p}^\star\}} \mathbf{e}^T\mathbf{p} \tag{11}$$

Namely, h'_a represents the second smallest $\mathbf{e}^T\mathbf{p}$ value. If ϵ is sufficiently small, which is rigorously specified by:

$$\epsilon \leq \frac{h'_a - h_a}{2BM(M-1)q_{\text{max}}} \tag{12}$$

Then the $\mathbf{q}^\star$ corresponding to $\mathbf{p}^\star$ (i.e., $\mathbf{q}^\star = \mathbf{T}^{-1}f^{-1}(\mathbf{p}^\star)$) must achieve the smallest $\mathbf{e}^T\mathbf{q}$ value among all $\mathbf{q}$'s in Θ_a.

One benefit of utilizing (8) instead of (7) is that the former only requires a single call to the function $f^{-1}(\cdot)$ after the optimization process, whereas the latter necessitates numerous calls to the function $f(\cdot)$ during the optimization. Thus, if $f(\cdot)$ is time-consuming to execute, (8) is particularly advantageous. In addition, (8) inherently breaks down the overall problem into two sub-procedures: determining $\mathbf{p}^\star$ from (8) and evaluating $\mathbf{q}^\star$ using $\mathbf{p}^\star$. This modular approach makes it easier to migrate the calculation when $f(\cdot)$ changes. Namely, if $f(\cdot)$ changes for any reason, re-solving based on (7) requires repeating the entire optimization, whereas re-solving using (8) only re-executes the last evaluation step without the need to re-run the whole optimization.

Finally, in the context of multi-objective optimization, solving the problem presented in (8) typically yields multiple Pareto optimal solutions, which may be excessive or unnecessary for practical purposes. One simplification usually being performed in real implementation is to convert (8) to a single-objective optimization with a user-predefined weight η:

$$\min_{\mathbf{p}\in[0,2\pi]^M} \quad Cost(\mathbf{p}) + \eta \cdot \mathbf{e}^T\mathbf{p} = Cost(\mathbf{p}) + \eta||\mathbf{p}||_1 \tag{13}$$

[2] An even more mathematically rigorous analysis could be performed to show that the Pareto Front of (7) could be exactly traced by mapping Pareto Front of (8) when varying the value a in Theorem 1.

TABLE I
DETAILED SETUP OF OUR SIMULATIONS

	Topology	Function	Dim.	Source(s) → Input Node(s)	Output Node(s)	Cost Components	η
#1	3-by-2 Square	Ring Resonator	34	$1.0 \rightarrow N_{L2}$	N_{R1}	Log Magnitude + Phase	10
#2	Triangular	Routing	46	$1.0 \rightarrow N_{L1}$	N_{R6}	Magnitude	0.01
#3	Hexagonal	50%:50% Splitting	60	$1.0 \rightarrow N_{29}$	N_4, N_6	Magnitude	0.1
#4	Hexagonal	Light Distribution	60	$1.0 \rightarrow N_1$, $(0.707 + 0.707\text{j}) \rightarrow N_{11}$	N_2, N_4, N_6	Magnitude	0.01
#5	5-by-5 Square	Filtering	120	$1.0 \rightarrow N_{L4}$	N_{R4}	Log Magnitude	100
#6	5-by-5 Square	WDM	120	$1.0 \rightarrow N_{L4}$	N_{R4}, N_{R8}	Log Magnitude	100

where $\eta > 0$ balances the importance of the two objectives.[3] Notice that (13) is a classical objective form and could be resolved by a plain gradient descent method or an advanced stochastic gradient descent approach [28], [29], [30]. However, the key question is do we have the gradients for them to be applicable? Recall that in Section II, we already know how to obtain the Jacobian matrix $\partial\mathbf{x}/\partial\mathbf{p}$ at a particular frequency point, where $\mathbf{x}$ collectively represents the circuit responses at all nodes. Obviously, the output ports $\{O_n \,|\, n = 1, 2 \cdots, N\}$ are a subset of $\mathbf{x}$, and thus, we already know how to calculate $\partial O_n(\omega_k, \mathbf{p})/\partial\mathbf{p}$ for any k and n. Consequently, the derivative of $Cost(\mathbf{p})$ with respect to $\mathbf{p}$ is straightforward from (6), and (13) is readily solved. In our paper, we address the thermal crosstalk effect in a recirculating programmable photonic mesh. This is in contrast to previous work which has investigated this effect in feedforward meshes, such as in [13].

IV. Numerical Results

The numerical simulations in this paper were all conducted on a RedHat Linux server with 16 Intel Xeon CPU cores running at 2.4 GHz. We implemented the differentiable S-matrix simulation in Python; its accuracy has been thoroughly validated by comparing its simulation results to those obtained from Lumerical Interconnect on several circuits. For the optimization shown in (13), we employed the built-in gradient descent method in the Scipy package. To set the parameters of the simulation, we used the measurement results from a recently fabricated chip. Specifically, we set the effective refractive index n_{eff} and group index n_g to 2.35 and 4.0, respectively, at a wavelength of 1550 nm. We further set the length L of the TBU to 250 micrometers (um), the transmission loss α to 0.98, and the function $f(q)$ to $0.06q\pi/\text{mW}$. For the thermal coupling coefficients in the matrix $\mathbf{T}$, we set the diagonal entries to one and the off-diagonal entries to 0.04 for phase shifters from different TBUs and 0.09 for phase shifters from the same TBUs. This choice is based on the fact that phase shifters from the same TBU are spatially closer than phase shifters from different TBUs.

To be as comprehensive as possible, we evaluate all commonly used topologies, such as triangular, square, and hexagonal mesh shown in Figs. 1 and 3, as well as the common light processing functions needed to be implemented on a PPIC, such as routing, splitting, filtering, and wavelength demultiplexing (WDM). Table I comprehensively lists all the considered test cases in our paper. Without explicitly mentioning, the considered triangular mesh and hexagonal mesh refer to those shown in Fig. 3. Additionally, we highlight that (6) may require slight modifications depending on the particular test case. For instance, when only a magnitude requirement is present, we can omit the phase part in (6) and use $||O_n(\omega_k, \mathbf{p})| - |U_n(\omega_k)||^2$ within the summation. Furthermore, we discovered that the use of log magnitude $|\log|O_n(\omega_k, \mathbf{p})| - \log|U_n(\omega_k)||^2$ in the summation of (6) is advantageous for applications that have a stopband. The unique treatment applied to the cost for each test case is outlined in the last column of Table I. To demonstrate the superiority of our proposed approach, we compare it against several baselines and implement the following methods:

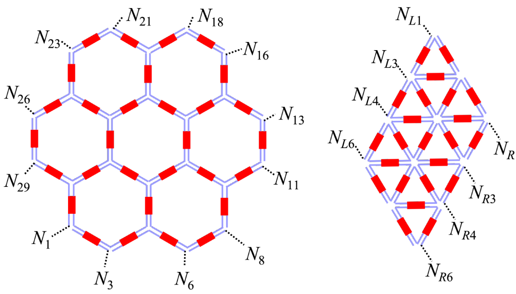


Fig. 3. Node naming convention for the hexagonal (left) and triangular mesh used in our article. As a hexagonal mesh does not have a left or right side, we have defined node naming based on a counter-clockwise order.

1) GD + L1 (Ours): solving (13) with gradient descent.
2) GD: solving (13) with $\eta = 0$.
3) BO: solving (13) with Bayesian optimization.
4) DE: solving (13) with differential evolution.
5) Prune-ρ-κ: solving (13) using gradient descent with $\eta = 0$, during which freezing $\rho\%$ of the elements in $\mathbf{p}$ to zero. The elements to be frozen are selected based on larger derivative values, and this process is repeated for κ predefined iterations with certain intervals.

For instance, by comparing BO/DE to our proposed method, we can demonstrate the effectiveness of the gradient-based approach for functional synthesis. On the other hand, comparing GD to our method can justify the power efficiency of L1 regularization. Note that for a fair comparison, we use the same $\eta \neq 0$ in our, BO, and DE approaches. All algorithms start with random phase initialization.

To ensure reliable comparisons between the different methods, we conducted 10 independent runs of each method and averaged the results to account for random fluctuations. We evaluated

[3] In our implementation, we only solve (13) once with one fixed η. However, if it is desired, (13) could be repeatedly solved with different η's. In doing so, the set of Pareto optimal $\mathbf{p}$'s and the Pareto Front are obtained.

TABLE II
RESULTS OF DIFFERENT APPROACHES ON DIFFERENT TEST CASES

	Method	Primary Evaluation Metrics			Other Reference Metrics	
		Run-time (mins)	Visual Check	Thermal Power	Cost w/ $\eta = 0$	Sparsity
Case 1	GD + L1 (ours)	3.52 ± 1.41 (No.3)	6/10 (No.2)	136.48 ± 47.76 (No.1)	133.20 ± 77.93 (No.3)	9.20 ± 7.76 (No.1)
	GD	3.39 ± 1.13	7/10	237.87 ± 23.70	96.94 ± 71.75	0.20 ± 0.40
	BO	59.62 ± 5.37	1/10	203.85 ± 21.09	314.30 ± 66.02	3.40 ± 2.29
	DE	19.21 ± 0.33	2/10	183.70 ± 14.34	260.20 ± 103.60	0.00 ± 0.00
	Prune-10%-2	2.84 ± 1.30	6/10	228.20 ± 21.88	148.80 ± 109.20	6.00 ± 0.00
Case 2	GD + L1 (ours)	2.16 ± 0.61 (No.3)	9/10 (No.1)	41.94 ± 26.56 (No.1)	5.00 ± 15.00 (No.1)	26.90 ± 11.51 (No.1)
	GD	1.01 ± 0.45	9/10	272.04 ± 23.09	5.00 ± 15.00	1.10 ± 0.94
	BO	18.84 ± 0.41	0/10	262.08 ± 19.58	42.96 ± 3.38	0.00 ± 0.00
	DE	15.08 ± 0.41	0/10	257.73 ± 17.10	31.96 ± 3.19	0.00 ± 0.00
	Prune-20%-2	1.31 ± 0.45	9/10	221.62 ± 16.67	5.00 ± 15.00	18.10 ± 0.30
Case 3	GD + L1 (ours)	8.52 ± 2.75 (No.2)	5/10 (No.2)	175.87 ± 98.38 (No.1)	0.19 ± 0.17 (No.3)	13.40 ± 15.40 (No.1)
	GD	6.30 ± 1.77	5/10	288.78 ± 19.18	0.16 ± 0.24	0.80 ± 0.87
	BO	68.59 ± 11.10	0/10	249.53 ± 13.97	0.18 ± 0.12	4.10 ± 2.21
	DE	35.84 ± 0.32	0/10	211.00 ± 8.15	0.28 ± 0.23	0.00 ± 0.00
	Prune-10%-2	11.19 ± 2.24	6/10	274.40 ± 15.54	0.06 ± 0.14	12.00 ± 0.00
Case 4	GD + L1 (ours)	109.4 ± 114.6 (No.2)	3/10 (No.1)	265.62 ± 49.34 (No.2)	4.42 ± 11.68 (No.4)	3.30 ± 3.44 (No.2)
	GD	75.25 ± 40.53	3/10	294.97 ± 13.72	4.47 ± 11.66	1.30 ± 2.28
	BO	610.34 ± 89.00	0/10	291.48 ± 20.82	0.21 ± 0.35	1.20 ± 1.40
	DE	419.2 ± 59.7	0/10	265.26 ± 16.00	0.12 ± 0.05	0.00 ± 0.00
	Prune-10%-2	373.4 ± 249.8	3/10	279.63 ± 14.48	0.74 ± 0.83	12.20 ± 0.00
Case 5	GD + L1 (Ours)	305.40 ± 22.81 (No.1)	7/10 (No.1)	242.35 ± 66.96 (No.1)	2.22e3 ± 1.18e3 (No.1)	8.70 ± 9.86 (No.2)
	GD	318.80 ± 25.94	7/10	345.50 ± 19.23	2.53e3 ± 1.11e3	0.40 ± 0.49
	BO	528.2 ± 13.09	1/10	316.85 ± 4.99	1.36e4 ± 1.38e3	0.00 ± 0.00
	DE	516.2 ± 2.07	0/10	288.82 ± 12.46	1.08e4 ± 1.44e3	0.00 ± 0.00
	Prune-10%-2	319.10 ± 23.24	6/10	323.66 ± 21.39	3.18e3 ± 1.49e3	24.00 ± 0.00
Case 6	GD + L1 (ours)	831.80 ± 778.50 (No.1)	8/10 (No.1)	320.09 ± 22.45 (No.1)	1.82e4 ± 2.80e3 (No.1)	0.50 ± 0.92 (No.2)
	GD	921.60 ± 713.80	8/10	352.50 ± 17.22	1.82e4 ± 1.57e3	0.30 ± 0.46
	BO	912.80 ± 251.50	0/10	327.4 ± 19.82	2.81e4 ± 2.11e3	0.00 ± 0.00
	DE	5282 ± 1093	0/0	387.00 ± 7.20	2.48e4 ± 1.80e3	0.00 ± 0.00
	Prune-10%-2	2950 ± 2070	8/10	328.44 ± 16.01	1.98e4 ± 3.66e3	24.00 ± 0.00

the performance of the methods based on three primary metrics: algorithm run-time, thermal power required to achieve the synthesized results, which is calculated as $\mathbf{e}^T\mathbf{q}^\star = \mathbf{e}^T\mathbf{T}^{-1}f^{-1}(\mathbf{p}^\star)$, and the number of synthesized responses that satisfied certain human visual standards (e.g., located within ±3 dB range). We also include two extra metrics for the readers' reference. The first is sparsity, which counts the number of phase shifts with an absolute value less than 0.0001. The second metric is obtained by recalculating the $Cost(\mathbf{p})$ value with $\eta = 0$ of the synthesized result during the evaluation stage, as it actually measures the mean squared error between the synthesized result and the target.

Table II presents a comprehensive performance comparison of various approaches on different test cases. The visual inspection metric is represented as x/y, where x is the number of synthesized results that pass the human eye check, and $y = 10$ is the total number of synthesized results. Other metrics are expressed as $x \pm y$, where x and y respectively represent the mean and standard deviation obtained from 10 repeated simulations. To better evaluate the effectiveness of the proposed method, we also report the ranking of the metric values of our approach when compared to other baselines in parentheses. In these rankings, red and green respectively denote that a smaller and a larger metric value are desired.

After analyzing the primary metrics of various methods across multiple test cases in Table II, we can conclude that our approach yields synthesized results that are comparable to those obtained through GD and pruning, while outperforming those achieved through BO and DE. In addition, our approach demonstrates a significant reduction in thermal power, despite occasionally experiencing a minor increase in algorithm run-time. It is worth noting that we deliberately allocated more run-time to BO and DE, yet they still underperformed in comparison to our method, justifying the importance of gradients in our functional synthesis task.

To obtain a more comprehensive assessment of our approach's performance, we here conduct a qualitative evaluation of the first test case. We actually construct this test case manually with a potential winning (golden) configuration in mind beforehand, which is shown in Fig. 4, and then use its frequency response as the target to run our algorithm. By setting θ and ϕ to the same value (e.g., 0π) in a TBU, the TBU is configured to a cross state where the signal changes arms upon propagating through the TBU. When θ and ϕ are set to have a π difference, the TBU is put into a bar state, resulting in the optical signal being confined to the same arm when propagating through the TBU. Other configurations lead to partial coupling states, where part of the optical signal is directed to the other arm. In Fig. 4, we utilize the aforementioned observations and construct a zigzag optical path in the first row of the square mesh, loaded with a ring resonator. We evaluate this constructed configuration, find that it consumes 63.29 mW, and has magnitude and phase response shown in Fig. 6(c). Correspondingly, we show two

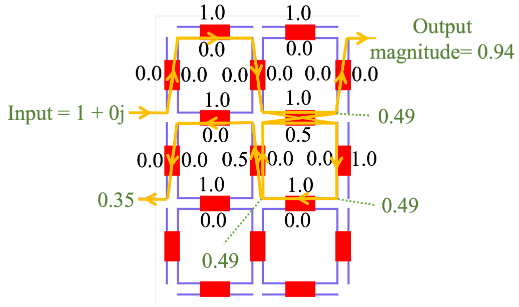
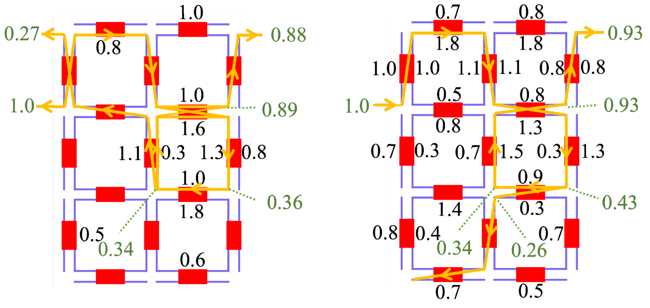


Fig. 4. We manually construct a phase shift configuration (marked by black) as case 1. For clearness, π is omitted (i.e., 0.5 in the figure means 0.5π phase shift) and the phase shifts in the bottom row are also omitted because they don't impact the light propagation. The input source (i.e., unit one) is injected at the node N_{L2}, and is highlighted by green. Other green markers highlight the magnitude response at selected nodes at 193.58 THz as an example. The yellow lines with arrows mark the major light paths in this circuit.



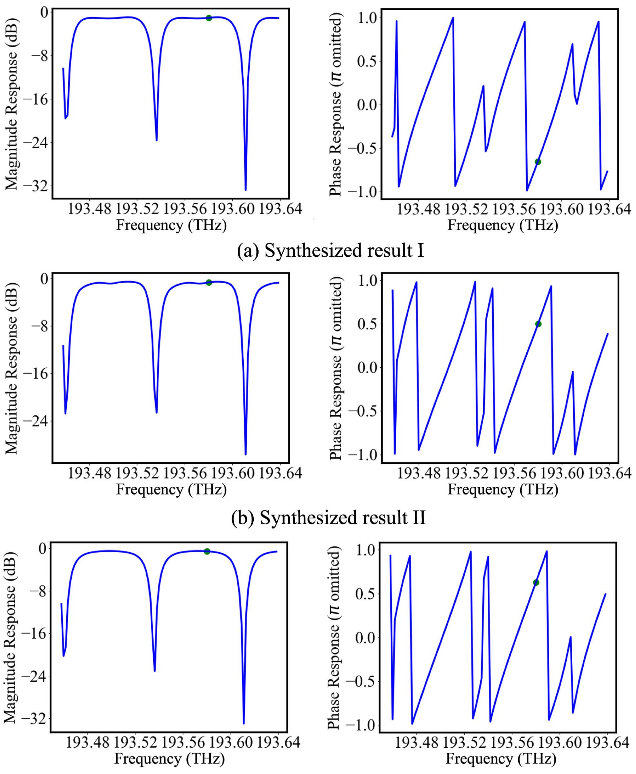
(a) Synthesized configuration I

(b) Synthesized configuration II

Fig. 5. Synthesized phase shift configurations (marked by black) obtained from our method in two independent runs. The suffix π is omitted, phase shifts smaller than 0.2π are omitted, and we only keep one decimal point of phases. Green markers highlight the magnitude response at selected nodes at 193.58 THz. The yellow lines with arrows mark the major light paths.

synthesized results in Figs. 5 and 6(a)–(b), and note that synthesized result I and II require 82.82 mW and 174.87 mW, respectively. Compared to the golden configuration, we first conclude that the synthesized results might not fully recover our golden configuration in mind. This observation is reasonable if we notice the following aspects: (i) this is an inverse problem in nature — there might be several configurations other than our constructed to realize the frequency response, (ii) the gradient descent approach might converge to local optima, and (iii) the cost function only set constraints on the output node N_{R1}. Nevertheless, it is essential to emphasize that the synthesized results are of satisfactory quality. For example, result I requires only approximately ~ 20 mW of additional power compared to the golden, and result II exhibits an almost identical response to the golden configuration. Furthermore, as detailed in Table II, our proposed approach meets specific specifications in six out of ten independent runs, further underscoring its robustness.

We present several visualization results for additional test cases. In the left side of Fig. 7, we observe that all undrawn phase shifts are close to zero, and both output nodes have a magnitude close to 0.69, meeting the requirement of a 50%:50% splitting, and the output optical signal power sum is approximately 1.0. On the right side of Fig. 7, our objective is to achieve an 80%:70%:50% power splitting ratio at the three output nodes N_2, N_4, and N_6. Our synthesized results satisfy this goal, as evidenced by the ratios $0.89^2 : 0.84^2 : 0.70^2 = 0.792 : 0.706 : 0.489$. It is important to note that this mesh visualization suggests that case 4 is difficult to optimize potentially because of many local optima, as large phase values are observed in the top hexagonal cells, despite the main optical path being located at the bottom (highlighted in yellow). This observation is further confirmed in Table II, where only three out of ten



(a) Synthesized result I

(b) Synthesized result II

(c) Target frequency response

Fig. 6. Target frequency response and the synthesized results by our method in case 1. The green dot in this figure corresponds to the frequency response highlighted by green in Figs. 4 and 5. Compared (a) and (b) to (c), we see our synthesized configurations achieve satisfying results.

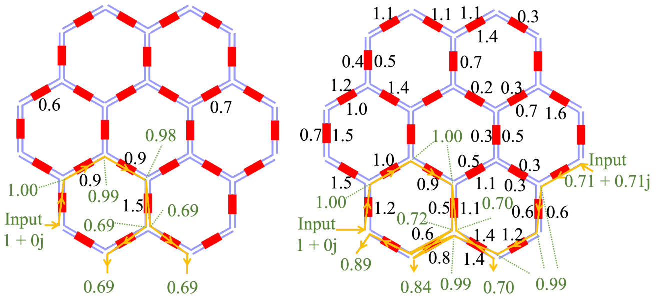


Fig. 7. Examples of synthesized phase configurations for case 3 (left) and case 4 (right). We follow the same procedure depicted in the caption of Fig. 5 to draw these figures. The configuration at the left and right need 22.92 mW and 157.09 mW thermal power to realize, respectively.

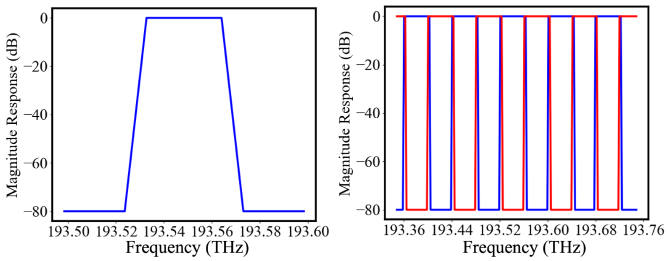


Fig. 8. Target magnitude response in case 5 (left) and case 6 (right).

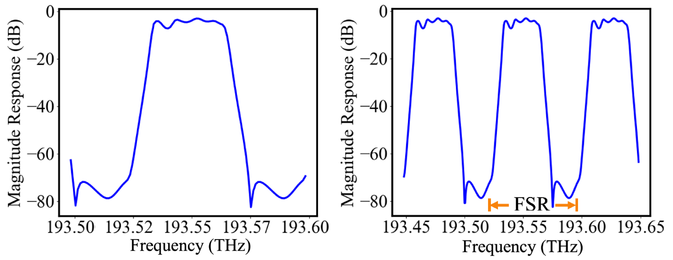


Fig. 9. Magnitude response of one synthesized result in case 5. Left: frequency range matches with that specified in the target response. Right: We expand the frequency range to outside that specified by the target to show FSR.

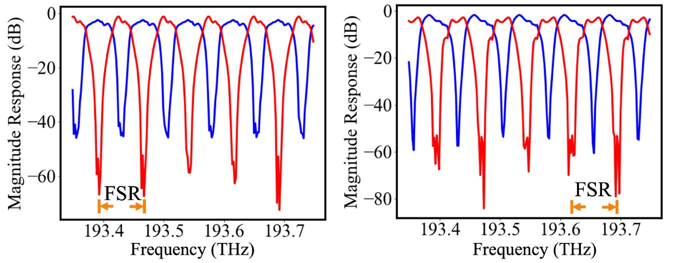


Fig. 10. Magnitude response of synthesized results in case 6, with left and right being two examples.

synthesized results satisfy the objective. Our future research will focus on enhancing the proposed approach's performance for challenging cases such as this one by incorporating human prior knowledge.

We further provide visualizations on cases 5 and 6. We first show the target magnitude responses in Fig. 8. Next, in Fig. 9, we plot the magnitude response of our synthesized result in case 5. It is noteworthy that the synthesized result exhibits a periodicity property known as free spectral range (FSR) [31]. Note that the FSR value achievable in a PPIC is determined by the phase configuration, the length L, and the group index n_g. As shown in Fig. 9, our approach automatically determines a configuration and FSR based on the given target response, without requiring explicit FSR specification. If a certain FSR is pre-specified by the user, we could also incorporate it by analogy to the right side of Fig. 8. Namely, we specify the target response in several periods to enforce an FSR. The corresponding synthesized results are shown in Fig. 10.

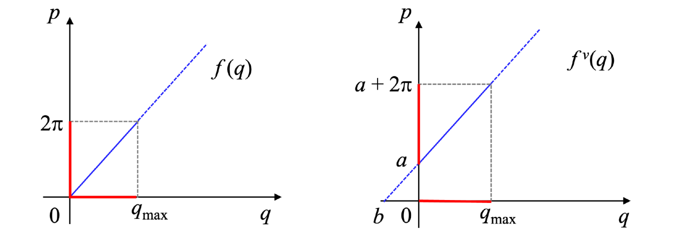


Fig. 11. An example of an affine characteristic function from power q to phase shift p when not considering initial phase (Left) and considering initial phase (Right). The red highlights indicate the domain of q and the range of p for both cases.

V. Conclusion

In this paper, we propose a unified approach for automating the synthesis of light processing functions on programmable photonic integrated circuits, which is applicable to any topology. We begin by modifying the scattering matrix simulation to make it differentiable and abstract a cost function in the frequency domain. We then show that the power-efficient functional synthesis problem can be formulated as a single-objective L1-regularized optimization problem in the space of phase shift. Our proposed approach outperforms several common baselines in terms of power efficiency and synthesized result quality, with only a slight and manageable increase in algorithm runtime.

Appendix

A. Discussion on Initial Phase Shifts

In the main text, we use the characteristic function $f(q) = p$ to represent the phase shift $p \in [0, 2\pi]$ induced by the delivered power $q \geq 0$ to a phase shifter. We assume that the phase shift is zero ($\theta = 0$ or $\phi = 0$) when not delivering power to a phase shifter, and mathematically this corresponds to $f(0) = 0$. When a non-zero initial phase due to manufacturing or environmental variation is considered, we use f^v to denote the corresponding characteristic function. Now f^v has a non-zero root, i.e., $f^v(b) = 0$ or $f^v(0) = a$, where $b < 0$ and $a > 0$, as demonstrated by Fig. 11. Intuitively, the consideration of initial phase is equivalent to shifting the root of f leftward, i.e., $f^v(q) = f(q + b)$.

To generalize, we need to use the vectorial format: $\mathbf{p} = f(\mathbf{T}(\mathbf{q} + \mathbf{b}))$ when considering all phase shifters in the circuit since thermal crosstalk among them exists. The components of $\mathbf{b}$ are essentially random variables, taking on different values for distinct circuits. However, once a programmable photonic circuit is fabricated, $\mathbf{b}$ becomes fixed and can be measured. Inspecting the derivation of our method, we observe that the initial phase shift $\mathbf{b}$ will modify the constraint in (7), from $\mathbf{p} = f(\mathbf{T}\mathbf{q})$ to $\mathbf{p} = f(\mathbf{T}(\mathbf{q} + \mathbf{b}))$. Nevertheless, this does not impact the validity of Theorem 1, because the additional $\mathbf{b}$ will be canceled out in (14). Thus, instead of solving the optimization in the $\mathbf{q}$ space shown in (7), we can solve in the $\mathbf{p}$ space shown in (8), while now the constraint is $\mathbf{a} \leq \mathbf{p} \leq 2\pi$ in (8). Moreover, we emphasize that calculating the optimal $\mathbf{p}^\star$ by solving (8) is not impacted by considering $\mathbf{b}$.

We note that only when the initial phase shifts $\mathbf{b}$ are measured, we can calculate the optimal power by $\mathbf{q}^\star = \mathbf{T}^{-1} f^{-1}(\mathbf{p}^\star) + \mathbf{b}$.

This finding underscores the advantage of our approach in solving within the $\mathbf{p}$ space, as it allows for the reuse of $\mathbf{p}^\star$ when confronted with two distinct programmable photonic circuits, each with its own phase shifts $\mathbf{b_1}$ and $\mathbf{b_2}$, respectively. Furthermore, it is important to note that measuring (as well as compensating) the random initial phase $\mathbf{b}$ poses not a unique challenge to our method but rather a universal one inherent to all software and hardware pertaining to programmable photonic circuits. Even if the fabrication-induced fixed random phase of each TBU were pre-characterized with high precision and compensated accordingly at the start, different environmental or noise sources may give rise to time-dependent phase variations during the operation of the circuit. Theoretically, power taps with integrated photo-detectors could be used inline with each TBU at the exterior of the photonic mesh, enabling the inverse calculation of phase shifts in parallel with the operation of the circuit. Infrared-camera-based imaging of the waveguides could allow direct monitoring of the power output of TBUs in the interior of the mesh. Active feedback control methods (e.g., [32]) could be used to dynamically correct for TBU phase shifts resulting from both initial fabrication variations as well as time-varying perturbations, such as thermal cross-talk and environmental fluctuations. The efficient implementation of electronic circuitry required for accurate phase measurement and correction in large-scale photonic integrated circuits will necessitate advanced electronic-photonic heterogeneous integration and co-packaging solutions.

B. *Proof of Theorem 1*

To prove Theorem 1, we need to show four things: (i) (10) is achieved at a unique $\mathbf{p}^\star$, (ii) the defined $\mathbf{q}^\star$ is in Θ_a, (iii) $h'_a - h_a$ is strictly larger than zero so that (12) is meaningful, and (iv) $\mathbf{e}^T\mathbf{q}^\star$ is smaller than $\mathbf{e}^T\mathbf{q}$ for any $\mathbf{q} \in \Theta_a/\{\mathbf{q}^\star\}$. For the first point, assume that there is another $\mathbf{p}^\dagger \neq \mathbf{p}^\star$ which could also achieve $\mathbf{e}^T\mathbf{p}^\dagger = h_a$. Then we will have $Cost(\mathbf{p}^\star) = Cost(\mathbf{p}^\dagger) = a$ and $\mathbf{e}^T\mathbf{p}^\star = \mathbf{e}^T\mathbf{p}^\dagger = h_a$, which violates the third assumption depicted at the beginning of Theorem 1. The second point is straightforward if we notice $\mathbf{q}^\star = \mathbf{T}^{-1}\mathbf{f}^{-1}(\mathbf{p}^\star)$ by definition. For the third point, it is straightforward based on the definition of h_a and h'_a after we have proved the first point (i.e., the uniqueness of $\mathbf{p}^\star$).

To prove the fourth point, we start from $(\mathbf{p}^\star, \mathbf{q}^\star)$ and another $(\mathbf{p}, \mathbf{q})$ with $\mathbf{p} \in \Gamma_a/\{\mathbf{p}^\star\}$ and $\mathbf{q} \in \Theta_a/\{\mathbf{q}^\star\}$:

$$\begin{aligned} &f(\mathbf{T}\mathbf{q}^\star) = \mathbf{p}^\star, \quad f(\mathbf{T}\mathbf{q}) = \mathbf{p} \\ \Rightarrow\ &\dot{f}\cdot\mathbf{T}\cdot(\mathbf{q}-\mathbf{q}^\star) = \mathbf{p}-\mathbf{p}^\star \end{aligned} \tag{14}$$

where in the second line, we apply Taylor expansion, and $\dot{f} \in (0, B]$ is used to represent the derivative of $f(\cdot)$. Now, notice that the above equation actually contains M equations in total, we consider the i-th equation:

$$\dot{f}\sum_{j=1}^{M} T_{ij}(q_j - q_j^\star) = p_i - p_i^\star \tag{15}$$

where T_{ij} represents the entry at the i-th row and j-th column of $\mathbf{T}$, and p_j represents the j-th entry of $\mathbf{p}$. Similar notations are adopted for $\{p_j^\star, q_i, q_i^\star\}$. Notice $T_{ii} = 1$, we have:

$$\dot{f}\left[q_i - q_i^\star + \sum_{j\neq i} T_{ij}(q_j - q_j^\star)\right] = p_i - p_i^\star \tag{16}$$

Furthermore, we know $|T_{ij}| \leq \epsilon$ and $|q_j - q_j^\star| \leq 2q_{\max}$, so the absolute value $|T_{ij}(p_j - p_j^\star)|$ is bounded by $2\epsilon q_{\max}$, and there are $(M-1)$ such terms inside the summation. Consequently, we could derive two inequalities from the above equation:

$$\Delta q_i - 2(M-1)\epsilon q_{\max} \leq \frac{\Delta p_i}{\dot{f}} \leq \Delta q_i + 2(M-1)\epsilon q_{\max} \tag{17}$$

where for simplicity we denote: $\Delta p_i = p_i - p_i^\star$ and $\Delta q_i = q_i - q_i^\star$. Now performing summation of the above equation with respect to i from 1 to M, we obtain:

$$|\mathbf{e}^T\mathbf{\Delta q} - \frac{\mathbf{e}^T\mathbf{\Delta p}}{\dot{f}}| \leq 2M(M-1)\epsilon q_{\max} \leq \frac{h'_a - h_a}{B} \tag{18}$$

where $\Delta\mathbf{p} = \mathbf{p} - \mathbf{p}^\star$ and its i-th entry is Δp_i. A similar definition is adopted for $\Delta\mathbf{q}$. Note that we have used the fact $\sum_{i=1}^{M}\Delta p_i = \mathbf{e}^T\mathbf{p}$ for an all-one vector $\mathbf{e}$ in the first step, and the condition (12) in the last step. Alternatively, we know:

$$\frac{\mathbf{e}^T\mathbf{\Delta p}}{\dot{f}} \geq \frac{\mathbf{e}^T\mathbf{\Delta p}}{B} = \frac{\mathbf{e}^T\mathbf{p} - \mathbf{e}^T\mathbf{p}^\star}{B} \geq \frac{h'_a - h_a}{B} \tag{19}$$

because $\mathbf{p} \in \Gamma_a/\{\mathbf{p}^\star\}$. Notice that the right-hand sides of (18) and (19) are identical. Then it is straightforward to prove $\mathbf{e}^T\Delta\mathbf{q} \geq 0$.

References

[1] W. Bogaerts and L. Chrostowski, “Silicon photonics circuit design: Methods, tools and challenges,” *Laser Photon. Rev.*, vol. 12, no. 4, 2018, Art. no. 1700237.

[2] W. Bogaerts, M. Fiers, and P. Dumon, “Design challenges in silicon photonics,” *IEEE J. Sel. Topics Quantum Electron.*, vol. 20, no. 4, Jul./Aug. 2013, Art. no. 8202008.

[3] L. Zhuang, C. G. H. Roeloffzen, M. Hoekman, K.-J. Boller, and A. J. Lowery, “Programmable photonic signal processor chip for radiofrequency applications,” *Optica*, vol. 2, no. 10, pp. 854–859, Oct. 2015.

[4] D. Pérez et al., “Multipurpose silicon photonics signal processor core,” *Nature Commun.*, vol. 8, no. 1, pp. 1–9, 2017.

[5] D. Pérez, I. Gasulla, J. Capmany, and R. A. Soref, “Reconfigurable lattice mesh designs for programmable photonic processors,” *Opt. Exp.*, vol. 24, no. 11, pp. 12093–12106, May 2016.

[6] D. Pérez-López, A. López, P. DasMahapatra, and J. Capmany, “Multipurpose self-configuration of programmable photonic circuits,” *Nature Commun.*, vol. 11, no. 1, pp. 1–11, 2020.

[7] A. López, D. Pérez, P. DasMahapatra, and J. Capmany, “Auto-routing algorithm for field-programmable photonic gate arrays,” *Opt. Exp.*, vol. 28, no. 1, pp. 737–752, 2020.

[8] J. Capmany, I. Gasulla, and D. Pérez, “The programmable processor,” *Nature Photon.*, vol. 10, no. 1, pp. 6–8, 2016.

[9] W. Bogaerts et al., “Programmable photonic circuits,” *Nature*, vol. 586, no. 7828, pp. 207–216, 2020.

[10] Z. Gao, X. Chen, Z. Zhang, U. Chakraborty, W. Bogaerts, and D. S. Boning, “Automatic realization of light processing functions for programmable photonics,” in *Proc. IEEE Photon. Conf.*, 2022, pp. 1–2.

[11] X. Chen, P. Stroobant, M. Pickavet, and W. Bogaerts, “Graph representations for programmable photonic circuits,” *J. Lightw. Technol.*, vol. 38, no. 15, pp. 4009–4018, Aug. 2020.

[12] Z. Gao, X. Chen, Z. Zhang, U. Chakraborty, W. Bogaerts, and D. S. Boning, “Automatic synthesis of light-processing functions for programmable photonics: Theory and realization,” *Photon. Res.*, vol. 11, no. 4, pp. 643–658, Apr. 2023.

[13] Y. Shen et al., “Deep learning with coherent nanophotonic circuits,” *Nature Photon.*, vol. 11, no. 7, pp. 441–446, 2017.

[14] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley, “Optimal design for universal multiport interferometers,” *Optica*, vol. 3, no. 12, pp. 1460–1465, 2016.

[15] D. A. B. Miller, “Self-aligning universal beam coupler,” *Opt. Exp.*, vol. 21, no. 5, pp. 6360–6370, Mar. 2013.

[16] D. A. Miller, “Self-configuring universal linear optical component,” *Photon. Res.*, vol. 1, no. 1, pp. 1–15, 2013.

[17] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, “Experimental realization of any discrete unitary operator,” *Phys. Rev. Lett.*, vol. 73, no. 1, 1994, Art. no. 58.

[18] S. Bandyopadhyay, R. Hamerly, and D. Englund, “Hardware error correction for programmable photonics,” *Optica*, vol. 8, no. 10, pp. 1247–1255, Oct. 2021.

[19] T. W. Hughes, M. Minkov, Y. Shi, and S. Fan, “Training of photonic neural networks through in situ backpropagation and gradient measurement,” *Optica*, vol. 5, no. 7, pp. 864–871, Jul. 2018.

[20] D. Pérez and J. Capmany, “Scalable analysis for arbitrary photonic integrated waveguide meshes,” *Optica*, vol. 6, no. 1, pp. 19–27, 2019.

[21] M. Milanizadeh, D. Aguiar, A. Melloni, and F. Morichetti, “Canceling thermal cross-talk effects in photonic integrated circuits,” *J. Lightw. Technol.*, vol. 37, no. 4, pp. 1325–1332, 2019.

[22] L. Fan et al., “Multidimensional convolution operation with synthetic frequency dimensions in photonics,” *Phys. Rev. Appl.*, vol. 18, no. 3, 2022, Art. no. 034088.

[23] L. Fan, K. Wang, H. Wang, A. Dutt, and S. Fan, “Experimental realization of convolution processing in photonic synthetic frequency dimensions,” *Sci. Adv.*, vol. 9, no. 32, 2023, Art. no. eadi4956.

[24] S. Buddhiraju, A. Dutt, M. Minkov, I. A. Williamson, and S. Fan, “Arbitrary linear transformations for photons in the frequency synthetic dimension,” *Nature Commun.*, vol. 12, no. 1, 2021, Art. no. 2401.

[25] T. W. Hughes, M. Minkov, I. A. Williamson, and S. Fan, “Adjoint method and inverse design for nonlinear nanophotonic devices,” *ACS Photon.*, vol. 5, no. 12, pp. 4781–4787, 2018.

[26] C. M. Lalau-Keraly, S. Bhargava, O. D. Miller, and E. Yablonovitch, “Adjoint shape optimization applied to electromagnetic design,” *Opt. Exp.*, vol. 21, no. 18, pp. 21693–21701, 2013.

[27] C. R. Harris et al., “Array programming with numpy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

[28] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations*, 2015.

[29] S. Ruder, “An overview of gradient descent optimization algorithms,” 2016, *arXiv:1609.04747*.

[30] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 315–323.

[31] L. Chrostowski and M. Hochberg, *Silicon Photonics Design: From Devices to Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[32] J. Carolan et al., “Scalable feedback control of single photon sources for photonic quantum technologies,” *Optica*, vol. 6, no. 3, pp. 335–340, 2019.