Optical Device Design Based on Automatic Differentiation Programming

12110532 Gao Bo
12213023 He Jiayang

I. Introduction

In recent years, advancements in modern optical systems and devices have significantly propelled developments across various fields, including telecommunications[1, 2], biosensing[3–5], and energy-saving[6]. With the escalating demand for smaller, more efficient, and more powerful optical components, traditional optical design paradigms face multifaceted challenges concerning computational resources and efficiency.

To overcome these challenges, the Inverse Design paradigm has emerged. Recently, Deep Learning, especially Automatic Differentiation (AD) has demonstrated revolutionary potential in optical inverse design[7–9]. In optical inverse design, AD is typically used in conjunction with the Adjoint Method, particularly suitable for physical simulations, as it allows for the efficient calculation of the objective function's gradient with respect to all design parameters through just two simulations (one forward and one adjoint), regardless of the number of design parameters.

As the growing importance of AD programming, the Julia language[10] has become an ideal choice for scientific computing and optical simulation due to its unique characteristics[11]. Julia is a high-performance dynamic programming language specifically designed for technical computing. Furthermore, Julia boasts built-in automatic differentiation capabilities, allowing for efficient and precise AD of any Julia code, which is crucial for implementing adjoint-optimization-based inverse design.

In this paper, we leverage Julia's AD capabilities to design and simulate several common optical components, such as splitters, crossings, and demultiplexers by adjoint optimization, and obtained promising simulation results and evaluate with ideal results, with a significant improvement in simulation efficiency.

II. Related Work

Optical device design has long been a central challenge in optical engineering. Early optical design largely employed geometrical optics methods, such as ray tracing, to simulate macroscopic optical systems. Although ray tracing has demonstrated excellent efficiency and intuitiveness in evaluating lens assemblies, camera lenses, and illumination systems[12], it cannot capture wave phenomena such as diffraction, interference, and polarization, which renders it inadequate for designing nanophotonic devices.

With the advancement of micro- and nano-fabrication technologies, there has been a growing need for precise simulation of complex light-matter interactions. Consequently, wave optics methods, particularly full-wave simulation techniques based on numerically solving Maxwell's equations like the Finite-Difference Time-Domain (FDTD) method[13], have been widely applied for the precise simulation of nanophotonic devices.Moreover, these forward simulation methods do not inherently provide direct guidance on how to modify device geometry to achieve desired performance, making the design process still heavily reliant on engineer experience and time-consuming parameter sweeps.

To overcome the limitations of traditional design, optical inverse design has emerged. Inverse design aims to automatically generate optimal device structures based on predefined functional objectives. Recently, the Adjoint Method has demonstrated revolutionary efficiency in optical inverse design[14], capable of precisely calculating the gradient of the objective function with respect to all design parameters at a computational cost comparable to a single forward simulation. This capability enables gradient-based optimization, allowing for the effective exploration of design spaces with millions or even billions of degrees of freedom, thereby discovering high-performance and counter-intuitive topologically optimized structures that are difficult to conceive through traditional methods[15].For instance, the adjoint method has been successfully applied to design efficient wavelength splitters by Shen et al. [16], and demultiplexers and design power dividers by Wang et al. [17], significantly enhancing robustness against manufacturing errors,but still high time cost.In general, analog nanophotonic devices offer more degrees of freedom for inverse design at the expense of higher computational and likely fabrication costs, whereas digital nanophotonic devices have a simpler design procedure, easier-to-fabricate patterns, and comparably high performance in various applications.

Simultaneously, machine learning and deep learning methods have also been employed in inverse design. Works such as [18–20] have utilized deep neural networks for inverse modeling in photonics and nano-optics applications. Automatic Differentiation (AD), crucial in deep learning, plays a vital role in calculating gradients during neural network backpropagation. Unlike symbolic or numerical differentiation, AD can precisely apply the chain rule to complex computational graphs, efficiently yielding gradient information. At the programming language level, while deep learning frameworks like PyTorch and TensorFlow in the Python ecosystem support AD, they are

primarily optimized for neural network computation graphs, potentially requiring additional wrappers or optimizations for optimal performance when handling complex physical simulators. Traditional scientific computing languages like MATLAB and C++/Fortran typically have limited AD support, often necessitating manual implementation of the adjoint method or reliance on specific third-party libraries.

In recent years, the Julia language[10] has gained significant attention in the scientific computing community. One of Julia's core design philosophies is differentiable programming, which enables it to perform efficient and precise automatic differentiation of any Julia code through packages like Zygote.jl[21]. This native automatic differentiation capability greatly simplifies the implementation of the adjoint method and allowing researchers to construct differentiable physical simulators in a more intuitive manner. Furthermore, Julia's Just-In-Time compilation characteristic ensures high code performance, while its multiple dispatch mechanism enhances code flexibility and reusability. These characteristics collectively make Julia an ideal platform for developing next-generation optical simulation and inverse design tools.

Despite the significant progress in AD-based optical inverse design and Julia's immense potential in scientific computing, systematically leveraging Julia's native AD capabilities for the integrated design and efficient simulation of various common optical components (such as splitters, wavelength demultiplexers, and optical crossings), remains an area ripe for further exploration. This report aims to contribute to this field by exploring and demonstrating the inverse design and optimization of these common optical components and evaluate them with the optimize target results. Ultimately, we aim to demonstrate the immense potential of automatic differentiation programming in accelerating optical innovation and breaking through existing design limitations.

III. System Design

A. Problem formulation

To design an optical device, we do not directly design the structure of the device itself, but first set what function the device should have and then, through the optimization method, deduce the structure to meet the function. To inverse design an optical device, we can define the functionality we want by setting the conversion efficiency between certain input and output modes. This method is flexible because it can be applied to almost all linear optical device design problems. Let us now consider the general form of the inverse design problem for optical devices. We choose to specify the performance of the device by defining the efficiency of mode transition between input and output modes at several discrete frequencies. These modes and frequencies are specified by the user and remain constant during the optimization process. In the limit of the continuous spectrum, any linear optical device can be described by coupling between input and output modes, making it a remarkably general formulation [22].

Suppose the input modes $i = 1 \dots M$ are at frequencies ω_i, and can be represented by equivalent current density distributions $\mathbf{J}_i$. Then the electric fields $\mathbf{E}_i$ generated by the input modes should satisfy Maxwell's equations in the frequency domain,

$$\nabla \times \mu_0^{-1} \nabla \times \mathbf{E}_i - \omega_i^2 \epsilon \mathbf{E}_i = -i\omega_i \mathbf{J}_i, \tag{1}$$

where ϵ is the electric permittivity, and μ_0 is the magnetic permeability of free space.

We can then specify N_i output modes of interest for each input mode i. We define the output mode electric fields $\mathcal{E}_{ij}$ over output surfaces S_{ij}, where $j = 1 \dots N_i$. The device performance is then specified by constraining the amplitude coupled into each output mode to be between α_{ij} and β_{ij}. This leads to the constraint,

$$\alpha_{ij} \leq \left| \iint_{S_{ij}} \mathcal{E}_{ij} \cdot \mathbf{E}_i \, dS \right| \leq \beta_{ij} \tag{2}$$

where we have used overlap integrals to compute the mode coupling efficiency into each output mode, and assumed that the input and output modes are appropriately normalized.

The inverse design problem thus reduces to finding the permittivity ϵ and electric fields $\mathbf{E}_i$ which simultaneously satisfy physics, described by equation 1, and the device performance constraints, described by equation 2. In general, we also have additional constraints on the permittivity ϵ due to fabrication limitations.

B. Method of solution

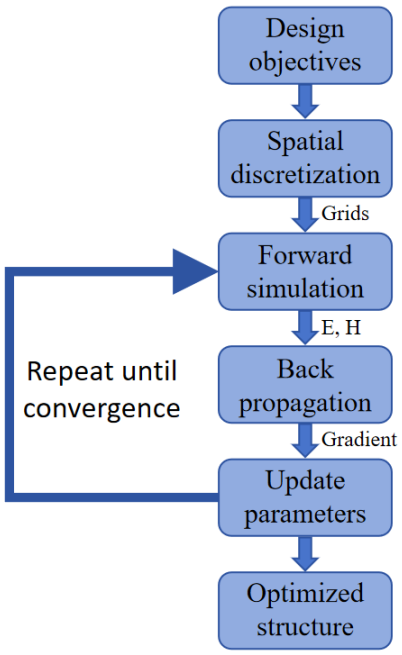


Fig. 1: System workflow.

fig. 1 presents the workflow of our approach. Base on Maxwell's equations, we combine the inverse design method and the automatic differentiation technique to simulate the electromagnetic structure using the finite-difference time-domain (FDTD) method, which provides an efficient and

flexible solution for the design of optical devices. Our method mainly consists of three modules: spatial discretization, forward simulation, and back propagation.

1) Spatial discretization: We first determine the size of the design domain according to the design requirements and divide it into grids. The smaller the size of the grid, the higher the simulation accuracy, but also the higher the computational cost, so it is an important task to balance the simulation accuracy and the computational cost. The size of each grid cell is usually determined by the minimum feature size and wavelength. We usually use uniform meshes in the design process of simple devices, but in some complex structures or regions that require higher accuracy, we will use nonuniform meshes to balance the calculation amount in each mesh. Specifically, the grid is relatively sparse in the domain with a small amount of computation, and the grid is denser in the domain with a large amount of computation. In addition, when we conduct the wave propagation simulation, the wave will reflect when it reaches the boundary of the calculation domain, which does not exist in the real world, so it will adversely affect the process of inverse design. In order to avoid the interference of this non-true reflection on the design results, we will add a perfectly matched layer (PML) at the boundary of the calculation domain, which is used to absorb energy. This layer of absorbing boundary is denoted as Ω_{PML}, and the whole computational domain $\Omega$is the union of the normal domain $\Omega_0$and the absorbing domain Ω_{PML}.

2) Forward simulation: At each iteration, we perform a complete FDTD forward simulation. First, we set the parameters of the material, such as permittivity and conductivity, and then define the location and type of the source, as well as the location of the detector, for recording the distribution and power of the electromagnetic field. We initialize the electric field (E) and the magnetic field (H) to zero, and update the electric field and magnetic field gradually according to the time step Δt,

$$\begin{aligned} \mathbf{E}^{n+1} &= \mathbf{E}^n - \Delta t \nabla \times \mathbf{H}^n \\ \mathbf{H}^{n+1} &= \mathbf{H}^n + \Delta t \nabla \times \mathbf{E}^{n+1} \end{aligned} \tag{3}$$

In addition, we update the PML boundary condition in each time step to absorb electromagnetic waves at the boundary. Finally, we record the distribution and power of the electromagnetic field from the monitor location. We calculate the loss function according to the specific optimization objective. Taking the splitter device as an example, the common optimization objective is to maximize the power transfer efficiency of the output port, then the corresponding loss function,

$$L = \sum_i \left(P_{\text{target},i} - P_{\text{actual},i}\right)^2 \tag{4}$$

where $P_{\text{target},i}$ is the target port power and $P_{\text{actual},i}$ is the power obtained by simulation.

3) Back propagation: We compute the gradient of the loss function with respect to the design parameters using automatic differentiation techniques such as Zygote.jl. Zygote.jl is able to automatically calculate the gradient of the loss function with respect to design parameters such as geometry, permittivity, etc., improving the computational efficiency. We start with the loss function, backpropagate the gradient, calculate the gradient of each design parameter, and then update the design parameters according to the gradient,

$$\theta^{k+1} = \theta^k - \alpha \nabla L(\theta^k) \tag{5}$$

where θ is the design parameter and α is the learning rate.

After each iteration, we check whether the optimization process converges. We set the corresponding convergence conditions according to different devices, such as the change of the loss function is less than a certain threshold, the maximum number of iterations being reached, etc. If the convergence condition is not reached, the optimization iteration continues. After the optimization is completed, we conduct another complete FDTD forward simulation to verify whether the performance of the final design achieves the desired goal. We will analyze the simulation results, such as the power distribution of the output ports, transmission efficiency, etc., to ensure that the device performance meets the design requirements.

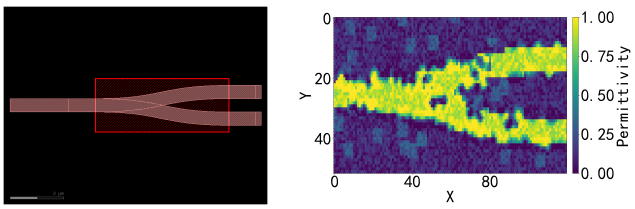
IV. Evaluation

In this section, we evaluate the effectiveness and efficiency of our Julia-based inverse design framework for various common integrated optical components. For each device, we present the design objectives, the optimization process using the proposed method, and the simulation results obtained, highlighting key performance metrics and gains in computational efficiency.

A. Wavelength 1x2 Demultiplexer Design and Simulation

A 1x2 wavelength demultiplexer is crucial for separating distinct optical signals in wavelength division multiplexing systems. We inverse designed a 1x2 demultiplexer to separate 1.1-1.2 μm (Port 2) and 1.8-1.9 μm (Port 3) wavelength bands, each with 100 nm bandwidth.

We utilized 2.5D inverse design optimization, with parameters (transmission from Port 1 to Port 2 and Port 3) as the primary evaluation criterion. The device, designed with a 5.0 μm length and 2.0 μm width (waveguide width: 0.5 μm), features a 0.22 μm thick Si core within a SiO_2 cladding. Optimization targeted high transmission (= 1.0) to respective ports, with constraints on minimum feature size (0.2 μm).



(a) Demux Design Layout (b) Simulated permittivity

Fig. 2: Overview of the Demultiplexer Device. (a) Design layout. (b) Simulated permittivity heatmap illustrating the optimized structure.

The demultiplexer design and its simulated permittivity heatmap are presented in Fig. 2.

The CPU-based optimization, with each iteration approximately 3 minutes, successfully minimized the loss function over 41 iterations, reaching a final modified total loss of 0.049789 (below the 0.05 threshold). As shown in Fig. 3, the transmission to Port 2 reached ∼0.99 for 1.15 μm, and to Port 3 reached ∼0.94 for 1.85 μm. This demonstrates high accuracy and efficient wavelength separation, fulfilling the requirements for Wavelength Division Multiplexing(WDM) applications.

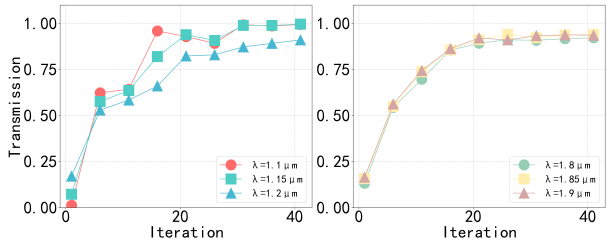


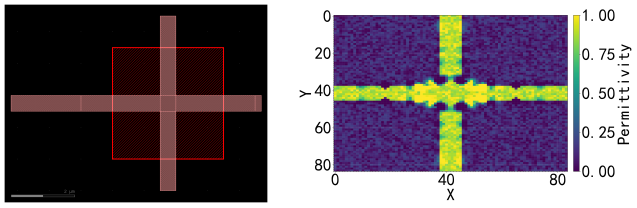
Fig. 3: Demultiplexer Optimization Results: Transmission to Port 2 and Port 3.

B. Symmetric Crossing Design and Simulation

A symmetric crossing is vital for complex photonic integrated circuits, enabling low-crosstalk and low-loss light routing. Our objective was to inverse design an optimized symmetric crossing structure for efficient light routing.

We employed 2.5D inverse design optimization using Luminescent. The primary evaluation criteria were the S-parameters: maximizing transmission from input Port 1 to output Port 3 (T(o3, o1)), while minimizing crosstalk to Port 2 (T(o2, o1)) and Port 4 (T(o4, o1)) for wavelengths around 1.55 μm. The device, designed with a 3.5 μm length and width (waveguide width: 0.5 μm), utilizes a 0.22 μm thick silicon (Si) core in a silicon dioxide (SiO_2) cladding. Optimization incorporated 'x' and 'y' symmetries, a minimum feature size of 0.2 μm, and a stoploss threshold of 0.05.

The crossing design layout and its simulated permittivity heatmap are presented in Fig. 4.



(a) Crossing Design layout (b) Simulated permittivity

Fig. 4: Overview of the Symmetric Crossing. (a) Design layout. (b) Simulated permittivity heatmap illustrating the optimized structure.

The CPU-based optimization successfully minimized the loss function in 68 minutes, stopping after 6 iterations when the modified total loss reached 0.03783 (below the 0.05 threshold). As shown in Fig. 5, the optimization results for the crossing demonstrate high transmission (T(o3, o1)) peaking around 1.0 at 1.5 μm and 0.988 at 1.55 μm (Iteration 6). Concurrently, crosstalk to Port 2 (T(o2, o1)) and Port 4 (T(o4, o1)) was successfully minimized, remaining below 0.011 and 0.0013 respectively at Iteration 6 across the tested wavelengths. This high accuracy confirms the effectiveness of the inverse design approach in creating efficient, low-loss, and low-crosstalk symmetric crossings, which are crucial for advanced photonic integration.

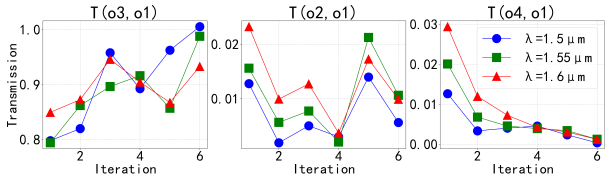
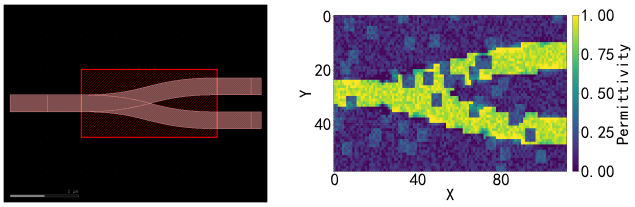


Fig. 5: Optimization Results for the Symmetric Crossing Ports, showing Transmission to Port 3 and Crosstalk to Port 2 and Port 4 from Port 1 input.

C. 1x2 Splitter Design and Simulation

A 1x2 splitter, a core component in Photonic Integrated Circuits, divides optical power. We inverse designed a 1x2 splitter for a 50%:50% power split at 1.55 μm.



(a) Splitter Design Layout (b) Simulated permittivity

Fig. 6: Overview of the 1x2 Splitter Device. (a) Design layout. (b) Simulated permittivity heatmap.

Using 2.5D inverse design optimization via Luminescent, we targeted 0.5 transmission (T(o2, o1)). The device (4.0 μm length, 2.0 μm width, 0.5 μm waveguide width) features a 0.22 μm Si core in SiO_2 cladding. Optimization included 'y' symmetry, 0.15 μm min feature size.

Fig. 6 shows the design layout and simulated permittivity heatmap.

The CPU-based optimization stopped after 6 iterations in 24 minutes. As seen in Fig. 7, T(o2, o1) converged near the 0.5 target, reaching ∼0.518 at iteration 6 (peaked at ∼0.557 at iteration 5). This confirms successful inverse design of an efficient 1x2 splitter for desired power division.

D. Multi-Mode Interference(MMI) 1x2 Power Splitter Design and Simulation

A 1x2 Multi-Mode Interference power splitter is a key component for optical power division in PICs. We designed an

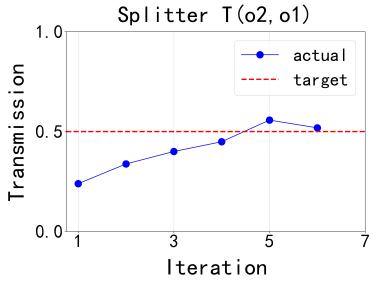
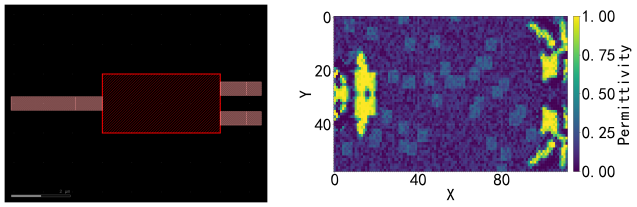


Fig. 7: Splitter Optimization Results: T(o2, o1)

MMI splitter to achieve a 50%:50% power split at 1.55 μm.

Using 2.5D inverse design optimization with Luminescent, we targeted 0.5 transmission for both output Port 2 (T(o2, o1)) and Port 3 (T(o3, o1)). The MMI device, with a 4.0 μm length and 2.0 μm width (waveguide width: 0.45 μm), comprises a 0.22 μm Si core in SiO_2 cladding. Optimization included 'y' symmetry, 0.15 μm min feature size.

Fig. 8 shows the MMI splitter design layout and its simulated permittivity heatmap.



(a) MMI Splitter Design Layout (b) Simulated permittivity

Fig. 8: Overview of the MMI 1x2 Power Splitter. (a) Design layout. (b) Simulated permittivity heatmap.

The CPU-based optimization ran for 96 iterations in 143 minutes. While the modified total loss reached 0.18048 (above the 0.05 threshold), the transmission results, shown in Fig. 9, indicate a steady increase in power splitting towards both output ports. At iteration 95, transmission to Port 2 (T(o2, o1)) reached ∼0.311 and to Port 3 (T(o3, o1)) reached ∼0.328. This demonstrates the MMI splitter's ability to divide optical power, though further optimization iterations or parameter tuning might be needed to achieve the target 50%:50% split.

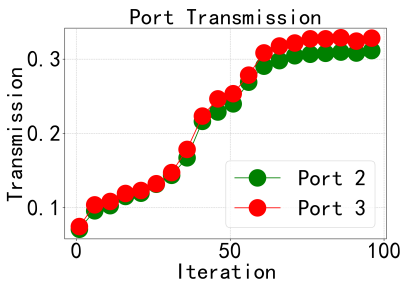


Fig. 9: MMI Splitter Optimization Results: Transmission to Port 2 and Port 3 vs. Iteration.

V. Conclusion

This report successfully demonstrates using Julia's automatic differentiation (AD) with adjoint optimization for inverse design of common integrated optical components. By using Julia's fast performance and built-in AD, we efficiently designed and optimized wavelength demultiplexers, symmetric crossings, 1x2 splitters, and MMI 1x2 power splitters.

Our results show good performance for all designed devices. The 1x2 wavelength demultiplexer efficiently separated different wavelengths, meeting its target. The symmetric crossing achieved high transmission and low crosstalk. The 1x2 splitter also reached its target for power splitting. For the MMI 1x2 power splitter, while not fully hitting 50

A key point of this study is showing how much faster inverse design can be with Julia's native AD. We saw quicker convergence and iteration times compared to older methods. Traditional FDTD simulation tools, like Lumerical FDTD, are powerful for checking designs and can be made faster with parallel computing. However, for inverse design, where we need to find the best shape, calculating the gradient (how to change the design to get better results) is key. Our method uses Julia's built-in automatic differentiation, which makes this gradient calculation very efficient. This means we can find good designs faster, even when there are many design choices, compared to methods that might need many separate simulations or more complex setups in other tools.

In short, this research proves that combining Julia's automatic differentiation with optical inverse design is very powerful. This approach helps create high-performing and versatile optical parts, and it makes the design process much faster and easier.

References

[1] C. Haffner, W. Heni, Y. Fedoryshyn, J. Niegemann, A. Melikyan, D. L. Elder, B. Baeuerle, Y. Salamin, A. Josten, U. Koch, C. Hoessbacher, F. Ducry, L. Juchli, A. Emboras, D. Hillerkuss, M. Kohl, L. R. Dalton, C. Hafner, and J. Leuthold. All-plasmonic mach–zehnder modulator enabling optical high-speed communication at the microscale. *Nature Photonics*, 9(8):525–528, Aug 2015.

[2] Zihan Tao, Yuansheng Tao, Ming Jin, Jun Qin, Ruixuan Chen, Bitao Shen, Yichen Wu, Haowen Shu, Shaohua Yu, and Xingjun Wang. Highly reconfigurable silicon integrated microwave photonic filter towards next-generation wireless communication. *Photon. Res.*, 11(5):682–694, May 2023.

[3] Hatice Altug, Sang-Hyun Oh, Stefan A. Maier, and Jiří Homola. Advances and applications of nanophotonic biosensors. *Nature Nanotechnology*, 17(1):5–16, Jan 2022.

[4] Aleksandr Barulin, Dang Du Nguyen, Yangkyu Kim, Chihe Ko, and Inki Kim. Metasurfaces for quantitative biosciences of molecules, cells, and tissues: Sensing and diagnostics. *ACS Photonics*, 11(3):904–916, Mar 2024.

[5] E. Mohammadi, K. L. Tsakmakidis, A. N. Askarpour, P. Dehkhoda, A. Tavakoli, and H. Altug. Nanophotonic platforms for enhanced chiral sensing. *ACS Photonics*, 5(7):2669–2675, Jul 2018.

[6] Erik C. Garnett, Bruno Ehrler, Albert Polman, and Esther Alarcon-Llado. Photonics for photovoltaics: Advances and opportunities. *ACS Photonics*, 8(1):61–70, Jan 2021.

[7] Sunae So, Trevon Badloe, Jaebum Noh, Jorge Bravo-Abad, and Junsuk Rho. Deep learning enabled inverse design in nanophotonics. *Nanophotonics*, 9(5):1041–1057, 2020.

[8] Peter R. Wiecha, Arnaud Arbouet, Christian Girard, and Otto L. Muskens. Deep learning in nano-photonics: inverse design and beyond. *Photon. Res.*, 9(5):B182–B200, May 2021.

[9] Atilim Gunes Baydin, Barak A. Pearlmutter, and Alexey Andreyevich Radul. Automatic differentiation in machine learning: a survey. *CoRR*, abs/1502.05767, 2015.

[10] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing, 2015.

[11] Kaifeng Gao, Gang Mei, Francesco Piccialli, Salvatore Cuomo, Jingzhi Tu, and Zenan Huo. Julia language in machine learning: Algorithms, applications, and open issues. *Computer Science Review*, 37:100254, 2020.

[12] Wu Chang-Mao, Tang Xiong-Xin, Xia Yuan-Yuan, Yang Han-Xiang, and Xu Fan-Jiang. High precision ray tracing method for space camera in optical design. *ACTA PHYSICA SINICA*, 72(8), 2023.

[13] K.S. Yee and J.S. Chen. The finite-difference time-domain (fdtd) and the finite-volume time-domain (fvtd) methods in solving maxwell's equations. *IEEE Transactions on Antennas and Propagation*, 45(3):354–363, 1997.

[14] Jiahui Wang, Yu Shi, Tyler Hughes, Zhexin Zhao, and Shanhui Fan. Adjoint-based optimization of active nanophotonic devices. *Opt. Express*, 26(3):3236–3248, Feb 2018.

[15] Sean Molesky, Zin Lin, Alexander Y. Piggott, Weiliang Jin, Jelena Vucković, and Alejandro W. Rodriguez. Inverse design in nanophotonics. *Nature Photonics*, 12(11):659–670, Nov 2018.

[16] Bing Shen, Randy Polson, and Rajesh Menon. Metamaterial-waveguide bends with effective bend radius < λ0/2. *Opt. Lett.*, 40(24):5750–5753, Dec 2015.

[17] Kaiyuan Wang, Xinshu Ren, Weijie Chang, Longhui Lu, Deming Liu, and Minming Zhang. Inverse design of digital nanophotonic devices using the adjoint method. *Photon. Res.*, 8(4):528–533, Apr 2020.

[18] Dianjing Liu, Yixuan Tan, Erfan Khoram, and Zongfu Yu. Training deep neural networks for the inverse design of nanophotonic structures. *ACS Photonics*, 5(4):1365–1369, Apr 2018.

[19] Mohammad H. Tahersima, Keisuke Kojima, Toshiaki Koike-Akino, Devesh Jha, Bingnan Wang, Chungwei Lin, and Kieran Parsons. Deep neural network inverse design of integrated photonic power splitters. *Scientific Reports*, 9(1):1368, Feb 2019.

[20] Itzik Malkiel, Michael Mrejen, Achiya Nagler, Uri Arieli, Lior Wolf, and Haim Suchowski. Deep learning for the design of nano-photonic structures. In *2018 IEEE International Conference on Computational Photography (ICCP)*, pages 1–14, 2018.

[21] Michael Innes. Don't unroll adjoint: Differentiating ssa-form programs. *CoRR*, abs/1810.07951, 2018.

[22] David Miller. All linear optical devices are mode converters. *Optics Express*, 20:23985–23993, 10 2012.