Inverse Design for Silicon Photonics

Subjects: Nanoscience & Nanotechnology
Submitted by: Simei Mao

Definition

Silicon photonics enables massive fabrication of integrated photonic devices at low cost, due to its compatibility with complementary metal oxide semiconductor (CMOS) process. It has emerged as one of the most important technologies for data communication and computations, as integrated electronic circuits are reaching their limits as well as incurring high energy costs. Recently, it has also shown the potential for other applications such as sensing and light detection and ranging (LiDAR). For miscellaneous silicon photonic devices, their versatile functions mainly come from design of device geometry. Intuitively, the design of single component with specific function is fulfilled by physics-based methods like analytical models, prior practical experiences and scientific intuitions. However, the physics-based methods are laborious and have high requirements on designers’ experience. Furthermore, the performance evaluation of light-material interaction with complicated geometries has to rely on numerical electromagnetic (EM) simulations, as the computation for irregular geometry is often non-intuitive. To make the design process more efficient, inverse design approaches have been introduced, which are assisted by various iterative optimization methods and deep neural networks (DNNs).

1. Iterative Optimization Algorithms

Iterative optimization algorithms can optimize device geometric parameters for targeted optical performance. They have been widely researched recently due to their time-efficiency and satisfactory optimization results. For devices with regular structure, key geometry parameters, which suggest the degrees of freedom (DOF) during design, are abstracted and tuned for targeted optical performance. For devices with one or two DOF, the optimal optical performance can be found by sweeping all the possible solutions. However, the sweeping process would be extremely laborious and require large time budget for higher DOF, as EM simulations are time-consuming. The relationships between geometry parameters and optical performance are quite similar to mathematical non-deterministic polynomial hard (NP-hard) problems \[1\] which are difficult to define by explicit function. Inspired by iterative gradient-free algorithms such as genetic algorithm (GA), particle swarm optimization (PSO) and direct binary search (DBS) for solving NP-hard problems, lots of regular silicon photonic devices are optimized in these ways \[2\]. Apart from gradient-free algorithms, gradient-based algorithms like topology optimization (TO) for devices with irregular structure have also been investigated for inverse design recently \[3\]. Instead of abstracting key parameters that describes device geometry, TO takes the whole design space as a bitmap, where all the pixels are iteratively updated towards the direction of gradient descent until the gradients are small enough. In this way, complex geometric pattern could be quickly generated for a targeted optical performance.

The iterative optimization algorithms discussed above could optimize only one device for one-round optimization and the whole optimization process has to be repeated again when the optical response target varies a little. This is extremely laborious when many similar devices are needed with slightly difference target response, such as power splitters with different splitting ratios or grating couplers (GCs) working at different wavelength bands \[4\]. Fortunately, DNN, an unprecedented tool capable of building a bridge between data features and data labels, has also found its way to assist the design of silicon photonic devices \[5\]. Based on the mapping direction between geometric parameters and optical response, DNNs are trained as forward models and inverse models. A forward model maps geometric parameter of a device to its optical response. It surrogates EM solvers to accelerate device performance evaluation in optimizations. On the contrary, an inverse model maps desired optical response to
geometric parameters. It takes almost no time to generate a device for a specific demand after the model is well-trained, eliminating the need for optimizations.

2. Inverse Design of Silicon Photonics with Iterative Optimization Algorithms

2.1. Inverse Design Schemes for Silicon Photonics

Different from electronic circuits where the versatile functions come from the combination of electronic components, a single silicon photonic device is able to achieve complex functions with delicate geometry design. For the given geometry of a device, its optical response can be evaluated by EM simulation before experimental validation. EM simulation tools have been developed by solving Maxwell’s equations numerically. These tools are important for device modelling and optimization, by allowing the designer to understand how light propagates in a given geometry and to obtain an accurate optical response. Widely adopted EM simulation methods include finite element method (FEM) [5], finite-difference time-domain method (FDTD) [2], eigenmode expansion method (EME) [8] and rigorous coupled wave analysis (RCWA) [9]. The FDTD method is among the most popular choices for silicon photonic device since it is a general EM solver well suited for high index contrast structures with complex geometry. It also allows wideband simulation as spectral response is generally needed for silicon photonic devices.

For a targeted optical response, inverse design tries to find a suitable device geometry. As shown in Figure 1, either a certain electric field distribution or an optical spectrum could serve as the figure of merit (FOM) which describes the phenomenon of interest to the designer. Other merits such as performance, compactness or fabrication feasibility may also be considered for different application scenarios. Based on the number of DOF, three types of structures are widely used for inverse design: empirical structure [10][11][12][13][14][15][16][17][18][19][20][21], QR-code like structure [22][23][24][25][26][27][28][29][30][31][32][33][34] and irregular structure [35][36][37][38][39][40][41][42][43][44][45][46][47]. An empirical structure making use of existing classical structures usually has only a few DOF. A QR-code like structure with hundreds of DOF selectively etches the design area with regular shapes such as rectangles or circles in a periodic manner. An irregular structure taking the whole design area as a bitmap has the most complex pattern with hundreds of thousands DOF. In the rest of this section, iterative optimization algorithms for optimizing these structures are illustrated and compared in detail.

![Figure 1. Inverse design schemes for integrated silicon photonics.](image)

2.2. Optimization of Empirical Structures

An empirical structure makes use of classical structures, where only a few parameters like radius, duty cycle, widths of waveguides, etc. are tuned for targeted optical response. These classical structures are proposed by researchers for different application scenarios. For example, micro-rings are preferred for wavelength selection due to their high wavelength-sensitivity [10][16]. Directional couplers (DCs) are used for polarization beam splitters (PBS), mode-division multiplexers (MDM) and wavelength-division multiplexers (WDM) since DCs couple light from one waveguide to another in a reciprocating way [10][13][15]. Y-splitter with three ports is widely used for power splitting (PS) [48]. Mach-Zehnder
interferometer (MZI) can separate light and finally combine with optical path difference, so that it could be utilized for optical switches or modulators [10][49][39]. Gratings which periodically arrange the silicon and air (or silica) have been applied to manipulate the effective refractive index of waveguides or act as fiber grating couplers (GC) [51][52]. With known structures and underlying theories to design a device for targeted FOM, it still costs lots of time to tailor the key parameters, as each parameter set requires at least one EM simulation.

The relationship between the abstracted key parameters from an empirical structure and its optical response is a kind of NP hard problem, which can be solved by heuristic algorithms. GA, a kind of heuristic optimized algorithm, is inspired by Darwinian evolution theory of survival of the fittest [33]. After initial chromosomes, which represent different parameter sets, are randomly generated, the fitness values describing the FOM of each chromosome are computed as shown in Figure 2a. Chromosome with better fitness values will be chosen as elite and passed to the next generation directly. For the rest of chromosomes, some chromosomes experience crossover with others to create new chromosomes and some produce offspring by mutation. Through the above elite selection, crossover and mutation, a new generation is born. This procedure is repeated until the best FOM is found.

![Figure 2. Schematic diagram of GA for the design of PBS based on directional coupler. (a) Operating principle of GA. (b) Initial structure of PBS and abstracted parameters. (c) Final structure of optimized PBS.](image)

We take reference [14] for example to illustrate utilizing GA to optimize an optical device with empirical structure. A broadband PBS based on directional coupler and subwavelength gratings (SWGs) structure is designed as shown in Figure 2b. Key parameters like pitch, duty cycle and coupling length are abstracted to represent this structure and coded as a chromosome. Initial chromosomes (20 parameters sets) are generated randomly within parameter boundaries. The FOM of each chromosome is calculated with respect to the output of transverse electric (TE) mode at through waveguide and transverse magnetic (TM) mode at cross waveguide. After hundreds of iterations, the best parameter set is found as Figure 2c suggests, where the transmission efficiencies for both TE and TM modes at desired output ports are larger than 85% over wavelength range from 1250 nm to 1680 nm.

Another popular heuristic algorithm is PSO [54]. Different form GA, where parts of parameter sets are updated at each iteration, PSO updates all the parameter sets towards the global optimized FOM. For PSO, the initial randomly generated parameter sets are called particles, each of which is assigned an initial velocity. PSO evaluates the FOM of each particle and finds the best global and local FOMs as well as the corresponding particles. According to the global best particle and the local best particle at their current iteration, the velocity of each particle is updated. With updated velocities, the new population are generated from original particles. This updating procedure is repeated until the global best FOM is converged to an acceptable value. PSO has also been widely used for the design of silicon photonic devices with empirical structures as listed in Table 1.

### 2.3. Optimization of QR-Code like Structures

A QR-code like structure, where regular shapes like circles or rectangles are selectively etched on a
periodic 2-D surface, is also an important approach for manipulating light propagation in waveguides. Different from empirical structure-based design, QR-code like structure with hundreds of DOF is a more general approach for different problems. In a QR-code like structure, designers only have to define the pitch and size of etched holes as well as the footprint of design area. For each hole, to be etched or not is determined automatically to achieve better FOM. Using a compact QR-code like structure, many different functional components like power splitter $^{[30]}$ $^{[34]}$, polarization rotator $^{[22]}$ $^{[31]}$ $^{[33]}$, PBS $^{[29]}$, WDM $^{[31]}$, MDM $^{[23]}$ $^{[28]}$, crossing $^{[55]}$, GC $^{[56]}$ and photonic crystal $^{[57]}$ have been designed.

DBS, a brute-force searching algorithm, initially proposed by Seldowitz et al. for the synthesis of digital holograms $^{[58]}$, has been proven to be efficient for the design of devices with QR-code structure. Shen et al. introduced DBS to design a very compact integrated PBS on SOI platform $^{[29]}$, as shown in Figure 3. They defined the size of design area to be $2.4 \times 2.4 \, \mu m^2$ with each hole size to be $0.12 \times 0.12 \, \mu m^2$. There are 400 rectangular pixels to determine whether to etch or not. During each iteration, one pixel in the design area is reversed and the FOM (average transmission efficiency of TE and TM modes at the targeted output ports) is calculated and compared with the FOM for last iteration. If the new FOM is larger than the old one, the new geometry with reversed pixel is past to the next generation, otherwise, the old geometry is kept. The above procedure is repeated until the FOM is large enough to be acceptable. After about 140 h of optimization, the average transmission efficiency of optimized PBS is higher than 70% and its 1 dB bandwidth is 83 nm.

In addition to DBS, heuristic algorithms like GA and PSO have also been applied for the design of integrated silicon photonic devices based on QR-code like structure. Different from DBS, which updates one pixel at each iteration, heuristic algorithms code all the pixels as an individual and update them together during each iteration, hence the total number of iterations could be reduced.

2.4. Optimization of Irregular Structures

An irregular structure owns the highest DOF since the whole design area is segmented into very small pixels $^{[1]}$. Different from QR-code like structure which also pixelates the design area with 100 or 200 nm resolution, an irregular structure is more intricate with ultra-high resolution (10 or 20 nm scale). It would be extremely time-consuming for gradient-free algorithms discussed above to optimize an irregular structure with thousands of DOF, as the required simulations is proportional to the number of DOF. Fortunately, gradient-based algorithms cast light on this time-consuming problem, where all the parameters are updated with the fewest simulations.

To describe the optimization process, a mathematical model which tries to optimize the FOM at the constraints of Maxwell equations is built as in Equation (1):
where FOM $f(\cdot)$ is the function of electric field $E$ and magnetic field $H$. Permittivity and permeability in the design area are variables to be tuned for better FOM. Objective-first (OF) method [15] and topology optimization (TO) with adjoint method [39] have been proposed to solve Equation (1).

We take irregular PBS in reference [60] as an example to illustrate the TO process. As shown in Figure 4a, the design domain $D_s$ is optimized by adjoint method for two FOMs. The first FOM is the transmission of TE mode at the upper arm while the second FOM is the transmission of TM mode at the lower arm.

![Figure 4](image)

**Figure 4.** Topology optimization of a PBS via adjoint method. (a) Schematic diagram of adjoint method. (b) Optimization process for the design area of PBS. (c) Final structure of optimized PBS.

As the optimization processes of two arms are similar, we only give detailed derivation of TE mode here. In the objective domain $D_0$, the actual EM fields are optimized towards fundamental TE mode. The targets at each position $x_0$ in the objective domain is defined in the form of Poynting vectors [59] as in Equation (2):

$$f(E(x_0), H(x_0)) = E(x_0) \times \overline{H_T(x_0)} + \overline{E_T(x_0)} \times H(x_0),$$

where $E, H$ are the actual EM fields from the simulation, and

$$\overline{E_T}$$

$$\overline{X_T}$$

are the constant conjugated EM fields of the fundamental TE mode. The best target $f(E(x_0), H(x_0))$ requires the value of actual EM fields to be the largest and the direction of actual EM fields to be the same with fundamental TE mode at position $x_0$.

However, it is still hard to determine whether the FOM is good or not when some positions get better targets while others get worse. Therefore, the total FOM is defined as the integral of all the targets in the objective domain as in Equation (3):
The derivative of total FOM to permittivity and permeability in the design domain is calculated using chain rule. Firstly, the derivative of total FOM to actual EM fields in the objective domain

\[ D_0 \]

is calculated as in Equation (4):

\[
\delta F(E, H) = 2 \Re \int_{D_0} \left[ \frac{\partial f}{\partial E(x_0)} \delta E(x_0) + \frac{\partial f}{\partial H(x_0)} \delta H(x_0) \right] d^2 x_0, \tag{4}
\]

where factors

\[
\frac{\partial f}{\partial E(x_0)}, \frac{\partial f}{\partial H(x_0)}
\]

are the constant values and are easy to get from Equation (2) as in Equation (5):

\[
\frac{\partial f}{\partial E(x_0)} = \hat{n}_E \times \frac{H_T(x_0)}{E_T(x_0) \times \hat{n}_H}. \tag{5}
\]

However, the terms \( \delta E, \delta H \) cannot be calculated analytically, as the relationship between \( E, H \) at objective domain and \( \epsilon, \mu \) at design domain cannot be expressed by an explicit function. Since the variation of permeability \( \delta \mu \) causes little change to EM fields, only the variation of permittivity \( \delta \epsilon \) is considered. In the design domain, the original geometry has electric field \( E(x_s) \) at \( x_s \). When a very small volume at this position has a small permittivity change \( \delta \epsilon \), this will cause an induced dipole moment

\[
\vec{p}^{\text{ind}}(x_s) = V_{\epsilon \text{ref}} E(x_s)
\]

Numerically, EM field variation

\[
\delta E(x_0), \delta H(x_0)
\]

at \( x_0 \) in the objective domain come from the summation effects of this induced dipole moments in the design domain, which can be expressed with Green’s function \[ \text{[61]} \] as in Equation (6):

\[
\begin{align*}
\delta E(x_0) &= \int_{D_s} G^{EP}(x_0, x_s) \vec{p}^{\text{ind}}(x_s) d^3 x_s, \\
\delta H(x_0) &= \int_{D_s} G^{HP}(x_0, x_s) \frac{1}{\mu} \vec{p}^{\text{ind}}(x_s) d^3 x_s \tag{6}
\end{align*}
\]

where

\[
G^{EP}(x_0, x_s)\]

and

\[
G^{HP}(x_0, x_s)
\]
represent the EM fields at $x_0$
in the objective domain from a unit dipole at $x_s$
in the design domain. The symmetry theory suggests the following relation as in Equation (7):

\[
G^{EP}(x_0, x_s) = G^{EP}(x_s, x_0) \\
G^{HP}(x_0, x_s) = -G^{HP}(x_s, x_0).
\]

By substituting Equations (6) and (7) into Equation (4) and making an equivalent transformation, we get the total FOM as in Equation (8):

\[
F(E, H) = \varepsilon_0 \varepsilon V \int_{B_0} d^3 x_0 E(x_s) \cdot 2 \Re \int_{B_0} d^3 x_0 \left[ G^{EP}(x_s, x_0) \frac{\partial f}{\partial E(x_0)} - G^{HP}(x_s, x_0) \frac{\partial f}{\mu H(x_0)} \right].
\]

The inner integration is defined as the “adjoint” electric field $E^A(x_s)$ at position $x_s$ in design domain as in Equation (9):

\[
E^A(x_s) = \int_{B_0} d^3 x_0 \left[ G^{EP}(x_s, x_0) \cdot \frac{\partial f}{\partial E(x_0)} - G^{HP}(x_s, x_0) \right].
\]

This adjoint electric field $E^A(x_s)$ can be obtained through the integration of all induced electric fields which come from EM dipoles with amplitudes

\[
\left[ \frac{\partial f}{\partial E(x_0)}, -\frac{\partial f}{\mu H(x_0)} \right]
\]
at different positions $x_0$
at the objective domain. These amplitudes are constant values which are calculated by using Equation (5). Through an “inverse” simulation, where light source with amplitude

\[
\left[ \frac{\partial f}{\partial E}, -\frac{\partial f}{\mu H} \right]
\]
is placed in objective domain, all the adjoint fields in design domain can be numerically calculated. Substituting Equation (9) into Equation (8), the derivative of total FOM in the objective domain to permittivity $\varepsilon_s$ in the design domain is shown in Equation (10):

\[
\frac{\delta F(E, H)}{\delta \varepsilon_s} = \varepsilon_0 V \Re [E(x_s) E^A(x_s)].
\]

With one forward simulation which calculates all the electric fields $E(x_s)$ and one inverse simulation which computes all the adjoint electric fields $E^A(x_s)$ in the design domain, the derivatives of total FOM to all the permittivity $\varepsilon_s$ can be obtained as Equation (10) suggested. As discussed before, there are two FOMs for PBS. For another FOM' which tries to guide TM mode to the second waveguide arm, the derivative of FOM' to permittivity in the design domain is also calculated in similar way. The permittivity in the design domain is updated towards the direction of total gradient decent as in Equation (11):
where $\alpha$ is the updating rate. The updating process is repeated until the gradient is small enough.

During the updating process, permittivity $\epsilon_s$ is taken as a continuous variable for convenience of derivative calculation, which has to be discretized for real application scenarios. Both level-set \cite{62} and density optimization \cite{63} have been applied for discretization. In silicon platform, the design domain usually composes of silicon and silica (or air) with permittivity $\epsilon_1$ and $\epsilon_2$, respectively. The level-set method defines a continuous variable with values ranging from negative to positive. $\phi_s = 0$ suggests the boundary between two materials and the discrete permittivity in the design domain is defined as in Equation (12):

$$
\epsilon_s = \begin{cases} 
\epsilon_1, & \phi_s \geq 0 \\
\epsilon_2, & \phi_s < 0 
\end{cases}.
$$

On the other hand, for density method, the values of continuous variable $\phi_s$ range from 0 to 1, and the permittivity in the design domain is expressed as in Equation (13):

$$
\epsilon_s = \epsilon_2 + \phi_s (\epsilon_1 - \epsilon_2).
$$

Different from level-set method which is discretized after applying Equation (12), the density method would generate “gray structures” with continuous permittivity between $\epsilon_1$ and $\epsilon_2$. To further discretize permittivity, Su et al. proposed a discretization method by introducing self-biasing or neighbor-biasing \cite{35}. In the example of PBS design, density method is adopted and the optimization process is shown in Figure 4b, where an initial random structure is finally optimized and discretized after 278 iterations.

Apart from discretization of permittivity, minimal feature size control is also an important issue for irregular structures as the large DOF bring features with critical size too small to be fabricated. Different approaches are proposed to solve this issue, such as density filters \cite{64}, penalty functions \cite{65}, artificial damping \cite{39} and morphological filters \cite{66}. Recently, Khoram et al. introduced b-splines to control minimal feature size of irregular structures \cite{67}. Instead of filtering out small features directly, this method transfers the design domain to lower space dimension composed of b-spline functions, which damps small features internally. In the example of PBS design, the optimized device geometry is shown in Figure 4c, where density adjoint method with b-spline function is adopted. Within the footprint of $2.4 \times 2.8 \, \mu m^2$, this optimized PBS splits TE and TM modes with over 90% transmission efficiency covering 420-nm wavelength range.

### 2.5. Comparison of Iterative Optimization Algorithms for Silicon Photonics Design

We have discussed three examples of inverse designed PBSs via different initial structures with corresponding iterative optimization algorithms in the above three subsections. The PBS based on empirical structure has high transmission efficiency (over 85% within 420 nm wavelength range) for both TE and TM modes and the critical size is under the control of designers. However, it also has the largest footprint (approximately $7 \times 4 \, \mu m^2$) and the lowest DOF (only 3). It takes about one week to optimize this device on two 12-core central processing units (CPUs). Profound prior knowledge about the coupling theory is also required and a skillful initial structure has to be chosen, otherwise, the device may not work at all. The PBS based on QR-code like structure has ultra-compact size ($2.4 \times 2.4 \, \mu m^2$) as well as high DOF (i.e., 400). Its design process does not need much prior knowledge but the optimization time is long (~140 design hours) as each QR-code has to be verified during DBS. This time-consuming optimization process limits both the size of footprint and the number of DOF, so the transmission efficiency of DBS optimized PBS is not very high. The peak efficiency of optimized PBS is ~80% and its 1 dB bandwidth is
83 nm. The irregular PBS optimized via adjoint method is also ultra-compact (2.8 × 2.4 μm²), and it has 16,800 DOF due to its ultra-high resolutions (i.e., 20 nm). After 64 h of optimization on two 12-core CPUs, the final irregular PBS has transmission efficiency over 90% covering 420-nm wavelength range. The irregular PBS has a time-efficient optimization process, where all the parameters can be updated in one round via only two simulations (one forward simulation and one adjoint simulation). However, profound knowledge about light-material interaction process, as well as advanced mathematics, is required for gradient calculations. Furthermore, gradient-based algorithms are easy to fall into local optimal. For example, the resolution of an irregular structure is usually 20 nm, which may result in minimal features too small to be fabricated.

Apart from the examples of PBS discussed above, inverse design based on these methods has been widely used for other silicon photonic devices as shown in Table 1. Empirical structure-based devices are large but they have high performance and are easy to process since their critical sizes are easy to control. QR-code like structures have compact footprints and controllable minimal feature sizes at the cost of some decrease in optical performance. Irregular devices have the most appealing performance, but minimal feature size control is in great need for the convenience of device fabrication.

Table 1. Examples of silicon photonic devices designed using iterative optimization algorithms.

<table>
<thead>
<tr>
<th>Device</th>
<th>Structure</th>
<th>Algorithms</th>
<th>Footprint (μm²)</th>
<th>DOF</th>
<th>Response (dB)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR [11]</td>
<td>Taper</td>
<td>PSO</td>
<td>15.3 × 1.5</td>
<td>29</td>
<td>S: 0.2</td>
<td>Easy to fabricate</td>
</tr>
<tr>
<td>PBS [13]</td>
<td>DC</td>
<td>PSO</td>
<td>5 × 1.5</td>
<td>10</td>
<td>S: 0.1</td>
<td>Low DOF</td>
</tr>
<tr>
<td>Convertor [18]</td>
<td>Taper</td>
<td>PSO</td>
<td>18.6 × 2.8</td>
<td>12</td>
<td>S: 0.06</td>
<td>Ultra-compact</td>
</tr>
<tr>
<td>Crossing [21]</td>
<td>SWG</td>
<td>GA</td>
<td>12.5 × 12.5</td>
<td>3</td>
<td>S: 0.64</td>
<td>Not compact</td>
</tr>
<tr>
<td>PR [23]</td>
<td>QR-code</td>
<td>GA</td>
<td>4.2 × 0.96</td>
<td>280</td>
<td>S: 0.7</td>
<td>High DOF</td>
</tr>
<tr>
<td>PBS [29]</td>
<td>QR-code</td>
<td>DBS</td>
<td>2.4 × 2.4</td>
<td>400</td>
<td>S: N/A</td>
<td>Compact</td>
</tr>
<tr>
<td>Bend [26]</td>
<td>QR-code</td>
<td>DBS</td>
<td>3 × 3</td>
<td>900</td>
<td>S: 0.9</td>
<td>Not-hard to fabricate</td>
</tr>
<tr>
<td>Diodes [32]</td>
<td>QR-code</td>
<td>DBS</td>
<td>3 × 3</td>
<td>900</td>
<td>S: 1.5</td>
<td>Not very high performance</td>
</tr>
<tr>
<td>MDM [23]</td>
<td>QR-code</td>
<td>DBS</td>
<td>2.4 × 3</td>
<td>500</td>
<td>S: 0.91</td>
<td></td>
</tr>
<tr>
<td>PS [27]</td>
<td>QR-code</td>
<td>DBS</td>
<td>2.72 × 2.72</td>
<td>400</td>
<td>S: 0.4</td>
<td></td>
</tr>
<tr>
<td>Convertor [18]</td>
<td>QR-code</td>
<td>DBS</td>
<td>4 × 1.6</td>
<td>320</td>
<td>S: 1.4</td>
<td></td>
</tr>
<tr>
<td>PR [23]</td>
<td>QR-code</td>
<td>DBS</td>
<td>5 × 1.2</td>
<td>600</td>
<td>S: 3</td>
<td>Very high DOF</td>
</tr>
<tr>
<td>PBS [33]</td>
<td>Irregular</td>
<td>TO</td>
<td>1.4 × 1.4</td>
<td>1225</td>
<td>S: 0.6</td>
<td></td>
</tr>
<tr>
<td>MDM [37]</td>
<td>Irregular</td>
<td>TO</td>
<td>2.6 × 4.22</td>
<td>11,429</td>
<td>S: 1</td>
<td></td>
</tr>
<tr>
<td>Matrix [69]</td>
<td>Irregular</td>
<td>TO</td>
<td>4 × 4</td>
<td>6400</td>
<td>MSE: 0.0001</td>
<td></td>
</tr>
<tr>
<td>Convertor [49]</td>
<td>Irregular</td>
<td>TO</td>
<td>4 × 1.5</td>
<td>4382</td>
<td>S: 0.08</td>
<td></td>
</tr>
<tr>
<td>WDM [43]</td>
<td>Irregular</td>
<td>OF</td>
<td>2.8 × 2.8</td>
<td>N/A</td>
<td>S: 2</td>
<td></td>
</tr>
</tbody>
</table>

Some researchers also combined different structures and algorithms to solve the limitations of certain methods for the design of a single device. For example, Xu et al. combined QR-code like structure into an empirical waveguide crossing structure to work as a wavelength filter [24]. Xie et al. proposed a global optimization method combining GA and annealing algorithm for optimizing on-chip twisted light emitter.
3. Deep Neural Networks Assisted Nanophotonics Design for Silicon Platform

Traditionally, the optical response of a given silicon geometry is fulfilled via EM simulations, which are accurate but time-consuming. If more than one device with similar structures and optical responses is needed for a specific scenario like arbitrary power splitter, different mode converter, wavelength filter or vertical grating for different wavelength bands, methods for iterative optimization in Section 2 will need to be performed multiple times for different target responses. It is important to find the relationship between optical response and geometric parameters instead of optimizing one by one. However, this implicit relationship is hard to find. Fortunately, DNNs open up a new way for silicon photonics design. DNNs have gained great enthusiasm from research community for their ability to find the complex and nonlinear relationships between input and output data [71].

Recently, DNNs have been introduced to accelerate the design process of silicon photonics. As shown in Figure 5, designer abstracts geometric parameters from a certain device structure like empirical structure, QR code-like structure or irregular structure. Optical response is sampled as like EM field distribution or optical spectrum which describes the designer's interested phenomena. Usually several groups of device geometric parameters are corresponding to a certain optical response [72]. The mapping from device geometric parameters to optical response is called a forward model [73][74][75][76][77][78][79][80][81][82]. While the inverse model describes the mapping from optical response to device geometric parameters [80][81][82]. Both of above-mentioned mapping types have been widely performed through DNNs.

Figure 5. Schematic of DNN-assisted silicon photonic device design approach.

Many advanced DNNs have already been considered for the design of metasurface [83][84][85][86][87][88][89][90][91][92] since many similar components with different diffraction angles are needed in this case. Advanced DNN architectures may also enable the design of integrated silicon photonic devices. In this section, we will discuss how DNNs can be trained as forward models and inverse models to assist the design ofnanophotonics on silicon platform. For each kind of DNNs, we will demonstrate a few examples as well as give basic mathematical background of some fundamental DNN types.

3.1. Training Discriminative Neural Networks as Forward Models

A forward model, which takes the device geometric parameters as features x and takes optical response as label y, builds a mapping relation as in Equation (14):

\[ y = F(x), \]

where F(·) is the complicated nonlinear function constructed by the DNN. The mapping from device geometric parameters x to optical response y is a kind of one-to-one problem, as a specific device geometry corresponds to only one optical response. Discriminative neural networks excel at modeling one-to-one mapping, such as regression or classification. In this section, we will focus on multi-layer perceptron (MLP) and convolutional neural network (CNN) for this kind of discriminable problem.
3.1.1. Multi-Layer Perceptron

The origin of MLP dates back to 1989 when Cybenko et al. proposed a feed-forward network with a single hidden layer containing a finite number of neurons. Under mild assumptions on the activation function, this neural network could approximate almost any continuous function with enough training data samples.

We take [24] for example to illustrate the process of constructing and training an MLP to surrogate EM simulation of grating couplers. As shown in Figure 6, the geometric parameters (grating pitch, duty cycle, fill factor, fiber angle and polarization) are encoded as data feature \( \mathbf{x} = [x_1, x_2, x_3, x_4, x_5] \). Peak efficiencies and corresponding wavelengths for TE and TM modes are marked as data label. The MLP adopted by Gostimirovic et al. consists of one input layer (5 neurons), three hidden layers (100, 50, 50 neurons) and one output layer (4 neurons), where the output value of each non-input neuron is determined by all the neurons in the last layer. Taking the first neuron in the third hidden layer for example, the weighted summation of neurons in the second hidden layer is shown in Equation (15):

\[
s_1^3 = \sum_{i=1}^{50} \omega_i^3 x_i^2 + b_3^3,
\]

where

\[
\omega_i^3, b_3^3
\]

are the ith weight and bias in the third hidden layer, respectively. The weighted summation is activated to get the output value of neuron as shown in Equation (16):

\[
x_1^3 = f(s_1^3),
\]

where

\[
f(\cdot)
\]

is the activation function which normalizes the output of each neuron to a finite range \([0, 1]\) or \([-1, 1]\). Activation functions are supposed to be computationally efficient, for there may exist abundant neurons in a DNN [84]. In this example, Rectified Linear Units (ReLU) activation function is adopted, which is expressed as in Equation (17):

\[
f(x) = \max(0, x).
\]

Figure 6. Training of an MLP to predict the coupling efficiency of grating couplers.

Through this feed-forward propagation, the relationship between input data feature \( \mathbf{x} \) and output label \( \mathbf{y} \) is shown in Equation (18):
where
\[ y = f(W^4 \cdot x^3 + b^4) = f(W^4 \cdot (W^3 \cdot x^2 + b^3) + b^4) = \ldots = F(x; W, b), \]

and
\[ W : [W^1, W^2, W^3, W^4] \]

\[ b : [b^1, b^2, b^3, b^4] \]

consists of weights and biases in all the layers. These weights and biases are determined by the training of MLP with known data samples, which is accomplished by backpropagation. Firstly, the loss function in the form of mean square error (MSE) is defined as in Equation (19):

\[ L(y^p, y) = \frac{1}{4} \sum_{i=1}^{4} (y^p_i - y_i)^2, \]

where
\[ y^p_i \]

is the predicted value of \( i \)th output neuron and is the \( i \)th label of a data sample. Apart from MSE, the loss function can be expressed, e.g., in the form of cross entropy loss function \(^{[95]}\). The loss function \( L(y^p, y) \) is optimized to its minimal by gradient descent, like stochastic gradient descent (SGD) or batch-size SGD. After the calculation of gradients for the \( t \)th sample, weights are updated as in Equation (20):

\[ W^{(t+1)} = W^{(t)} + \alpha \nabla_{w(t)} L(F(x^{(t)}; W^{(t)}), y^{(t)}), \]

where \( \alpha \) is the learning rate to control the update speed. Bias \( b \) is updated in a similar way. In the example of reference \(^{[74]}\), 9190 data samples are generated via 2D FDTD and separated with a ratio of 85:15 for training and validation. The trained MLP has 93.2% prediction accuracy and is 1830 times faster compared with FDTD simulation, hence this trained MLP is beneficial to surrogate EM simulations of grating couplers. This trained MLP is easy to connect with optimization algorithms for rapid development of new grating couplers. Gostimirovic et al. demonstrated that their trained MLP with brute-force sweeping could find an optimal design 61 times faster than FDTD simulations with PSO.

However, for increasing DOF, an MLP should be deep enough to map the data features to the data labels and the required training data samples will also increase. For example, Tahersima et al. trained an eight-layers ResNet with 20,000 data samples to predict the output spectra of power splitter with QR-code like structure \(^{[81]}\). Apart from increasing the layers of an MLP, reducing the dimension of data features is also an efficient way to map high-dimensional data features to the data labels, which could also alleviate the requirement for a large number of training data samples. Melati et al. have demonstrated that principal component analysis (PCA) is capable of compressing high-dimensional geometric parameters to a subspace without useful information loss \(^{[99]}\). To avoid the massive data samples requirement for devices with very high DOF like irregular structures, Liu et al. adopted Fourier transform together with a high frequency filter to diminish high-frequency geometry information and then use this compressed data in Fourier domain to train an MLP as a forward model \(^{[92]}\).

### 3.1.2. Convolutional Neural Network

As discussed in **Section 3.1.1**, MLP it is hard to process devices with high-DOF structures such as QR-code and irregular structures without introducing additional methods to reduce data dimensionality. Fortunately, CNN, another popular DNN, is innately capable of processing high-dimensional data features while requiring smaller training data samples than an MLP under the same conditions. Inspired by human visual information procedure illustrated by David (1981), CNN processes the data layer by layer from specific to abstract and has been widely used for image classification, separation and recognition \(^{[98]}\). A complete CNN usually consists of three types of layers, i.e., convolutional layers, pooling layers and fully-
connected layers. Convolutional layers extract local features of an input data matrix, while pooling layers, also called down-sampling layers, reduce the data dimensionalities greatly and avoid overfitting. The fully-connected layers are similar to an MLP, which project extracted data features to the data labels.

We take reference [25] as an example to illustrate the process of building and training a CNN as a forward model to predict output spectra of a QR-code like power splitter. As shown in Figure 7a, this power splitter consists of one inverse tapered input waveguide, two tapered output waveguides and a $2.6 \times 2.6 \mu m^2$ middle square design area which is separated into $20 \times 20$ pixels. Each of these pixels has a binary state as etched (coded as 1) with a diameter of 90 nm circle and not-etched (coded as 0). This 0–1 coded $20 \times 20$ matrix works as the input data features $x$ of CNN, which goes through three convolutional layers with ReLu activation. In a convolutional layer, different convolutional kernels convolve with input matrix to generate different sub-matrices as shown in Figure 7b. After convolutional layers, the extracted features are condensed via pooling layers and flattened into a vector. This vector containing extracted features is then projected to data labels $y$ via fully-connected layers with Sigmoid activation function. The data labels are vectors with samples of transmission spectra as their elements shown in Figure 7c. After the training, this CNN predicted transmission performance with 85% accuracy compared with simulation results, which was much better than 16% accuracy of a 4-layered MLP using the same training data samples. Thus, a well-trained CNN is preferred for predicting the optical performance of devices with QR-code structures.

![Figure 7](image)

Figure 7. Training of a CNN to predict the efficiency of a power splitter based on QR-code like structure. (a) Initial QR-code structure with one input waveguide and two output waveguides. (b) Schematic process of CNN. (c) Optical performance of power splitter. (d) A simple example of convolutional and max pooling operations.

A simple example of convolutional and maximal pooling processes is demonstrated in Figure 7d. As shown in the red square, the pre-defined convolutional kernel is multiplexed to pixels in the original matrix to get a new value in sub-matrix. Moving the convolutional kernel along the original $5 \times 5$ matrix, we get a $3 \times 3$ sub-matrix. To further decrease the matrix dimension, maximal pooling operation is performed, where only the maximum value is picked up to represent the sub-matrix as marked by orange squares. These convolutional and pooling layers are widely used for high-dimensional image data processing and are easy to train.

### 3.2. Training Generative Deep Neural Networks as Inverse Models

A well-trained inverse model can predict the component geometries directly according to target optical responses without any iterative optimization procedures. However, it is difficult to build and train a DNN as an inverse model because a specific optical response corresponds to different component geometries, which is hard to model via discriminative neural networks. Luckily, generative neural networks can solve this one-to-many projection [29]. Instead of fitting data features to data labels directly, generative neural networks model the underlying distributions of data features (device geometry parameters) and data labels (optical response). However, training of generative neural networks is hard to converge without enough data samples. To ease the burden of massive requirement of training data samples, semi-
supervised generative neural networks like conditional variational autoencoder (CVAE) and conditional generative adversarial network (CGAN) have been proposed for building inverse models [80][92][100]. Furthermore, combined with an EM simulator such as FDTD, unsupervised generative neural networks have also been trained as inverse models, where prepared training data samples are not demanded [88][89].

### 3.2.1. Conditional Variational Autoencoder

Variational autoencoder (VAE), a kind of generative neural network, generates data features through a decoder with sampled normal distribution input [101]. The input of a decoder is supposed to have the same normal distributions with the output of an encoder. In the training process, an encoder projects original data features to their unique mean values and variances, which are supposed to obey normal distributions, while a decoder tries to reconstruct original data features through potential variables sampled from normal distributions. The VAE is trained by minimizing the difference between real features from data samples and generated features from VAE and by ensuring that the encoded data obeys normal distributions. After the training process, different groups of new data features can be generated by sampling different potential variables from the normal distributions and feeding them into encoder. However, VAE can only generate data features (e.g., device geometry) similar to their training data samples, which is not suitable for an inverse design problem where a data label (target optical response) is desired. Fortunately, CVAE is a kind of model which melts label information into a VAE so that the trained model could generate data features according to data labels.

We take reference [80] as an example to illustrate how to train a CVAE as an inverse model to generate component geometry according to user-defined optical spectrum. The initial power splitter structure is similar to the structure in Figure 7 with 20 × 20 pixels in the middle square design area but the diameters of etched holes are tunable to increase the DOF of QR-code structure. 20 × 20 data features with values ranging from 0 to 1 are used to represent these holes, where the variables with value less than 0.3 represent no hole and variables with value [0.3~1] correspond to etched holes with diameters [44~77 nm]. As shown in Figure 8, features from real data samples are encoded through convolutional layers and fully-connected layers into mean values and variances. The latent vector which are sampled from Gaussian distributions described by and are concatenated with coded optical response to work as the input of autoencoder. After two de-convolutional layers and pooling layers, a geometry similar with training data samples is generated. The aim of the training process is to diminish the difference between original and generated geometric parameters and to decrease the Kullback-Leibler (KL) divergence between the latent and the Gaussian prior. Prepared 15,000 training data samples, including some semi-optimized power splitters with different splitting ratios, are used to train this CVAE. The trained CVAE then generates 1000 new geometrical patterns according to different splitting ratios within seconds, which are validated by FDTD and marked with labels to train the CVAE again for better performance. This trained CVAE is able to generate power splitters with 90% efficiencies for given splitting ratios.

![Figure 8. Training process of a CVAE to design a power splitter with QR-code like structure.](image)

### 3.2.2. Conditional Generative Adversarial Network
Different from VAE which encodes original data features into normal distributions and decodes these normal distributions to reconstruct original data features, a generative adversarial network (GAN) uses a kind of game theory for training \([102]\). A GAN consists of a generator and a discriminator which compete with each other. With random noise input, generator produces a group of generated data features (e.g., device geometry parameters) while the discriminator judges whether this group of generated data features is real or not. The training of GAN is a game process where the generator intends to fool discriminator with generated data features and the discriminator tries to distinguish generated features from the original ones. Firstly, the discriminator is trained several times to distinguish generated features from original features. Then, the discriminator works as a supervisor to train the generator until the generator can produce data features to deceive the discriminator itself. The training procedures of discriminator and generator are alternatively repeated several times until the generator can produce almost “real” data features. Like a CVAE, a CGAN also merges data labels into the model to generate new data features. Random noise and data labels are concatenated and then applied as input to the generator.

We take reference \([103]\) as an example to illustrate the process of training a CGAN as an inverse model to predict the irregular geometries of metagratings. The target optical properties are diffraction angle \(\theta\) and diffraction wavelength \(\lambda\). As shown in Figure 9, the label \(y\) which describes the optical properties is combined with random noise to work as the input of generator. The random noise \(n\) is utilized to improve the robustness of the generator as well as to generate different irregular geometries. The generator, which consists of two fully-connected layers, four de-convolutional layers and a Gaussian filter, produces new irregular geometries. The discriminator consists of convolutional layers and fully-connected layers. It works as a supervisor to determine whether a generated irregular geometry shares the same properties with the prepared irregular geometries. The generator and discriminator are trained alternately. 600 high-resolution irregular half-optimized metagratings are used as the initial training data samples. After training, the CGAN generates thousands of irregular metagratings within seconds. They are used as new training data samples to retrain CGAN for better performance. Most of the generated metagratings from the trained CGAN have efficiencies over 75\%, much better than randomly generated metagratings whose best efficiency is less than 30\%. The efficiency of irregular metagratings generated by CGAN together with a few iterative optimization procedures could reach up to over 90\%, comparable to that of iterative-only optimization methods.

Using a CGAN, Liu et al. further merged a pre-trained forward neural network with CGAN to calculate the optical response of a generated device \([103]\). The generator was trained with two aims: the first to diminish the difference between generated structure and real structures under the supervision of discriminator, and the second to reduce the difference between targeted optical response and generated optical response which is calculated via pre-trained forward neural network. In this way, apart from making the generated device geometry similar to real device structures, the transmission efficiency of generated device is also supposed to be similar to the target optical response.

### 3.2.3. Unsupervised Generative Neural Network
In addition to learning from the data samples, DNN can also work as a trainable function and merge with EM-simulator based optimization algorithms. In this way, training data samples are not required.

We take reference [100] as an example to illustrate the process of building and training an unsupervised neural network as an inverse model to generate irregular metagratings. The irregular geometries of metagrating are also produced by a well-trained generator for the given optical response as well as random noise. As shown in Figure 10, the device geometry $G$ is the function of diffraction wavelength $\lambda$, angle $\theta$ and random noise $n$ as shown in Equation (21):

$$G = f_1(\lambda, \theta, n; W),$$  \hspace{1cm} (21)

where $f_1(.)$ is the function described by the generator with trainable parameters matrix $W$. With known metagrating geometry $G$, the optical response $E_G$ can be calculated by forward EM simulation as in Equation (22):

$$E_G = f_2(G),$$  \hspace{1cm} (22)

where $f_2(.)$ is the non-analytical function computed by EM simulation. The loss function is an analytical function defined by designer as in Equation (23):

$$L = f_3(E_T, E_G).$$  \hspace{1cm} (23)

Figure 10. Merging DNN to optimization algorithms.

The aim of $f_3(.)$ is to diminish the difference between target optical response $E_T$ and generated optical response $E_G$. To train this deep generator, back-propagation is applied as in Equation (24):

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial E_G} \cdot \frac{\partial E_G}{\partial G} \cdot \frac{\partial G}{\partial W} = \frac{\partial f_3}{\partial E_G} \cdot \frac{\partial f_2}{\partial G} \cdot \frac{\partial f_1}{\partial W}. \hspace{1cm} (24)$$

The first term $\frac{\partial f_3}{\partial E_G}$ is an explicit function, which is easy to calculate analytically. The second term $\frac{\partial f_2}{\partial G}$, which is an implicit function, can be solved through adjoint method with an inverse simulation as derived in Section 2. The last term...
\[
\frac{\partial f_1}{\partial W}
\]

is the loss function of deep generator which can be trained via back-propagation similar to other DNNs.

In this way, the generator is no longer trained by prepared data samples but with the assistance of EM simulation in a chain rule. After the training process, the generator is able to produce multiple metagratings with different geometric parameter sets for a target optical response.

Reinforcement learning which merges DNN with EM simulator is another approach without the need of training data samples. DNNs based on reinforcement learning generate new data features which are evaluated by EM simulator and fed back for the training of DNNs. Sajedian et al. utilized reinforcement learning for silicon-based color printing design (90). Different from examples discussed above which make use of gradient of EM simulator, reinforcement learning makes decisions on the rewards of short-term and long-term gains.

3.3. Comparision of DNNs for the Design of Nanophotonics on Silicon Platform

In this section, we discussed some popular DNNs used as forward models and inverse models to accelerate the design process of silicon photonics. Here, we will classify and compare the advantages and disadvantages of these DNNs for the design of nanophotonics on silicon platform.

Forward model built via DNN serves as a boosted EM simulator for devices with similar initial structures to the training data samples. A well-trained forward model can predict optical response for given geometric parameters at almost no time cost. Combining forward model with iterative optimization algorithms, geometric parameters for a target optical response can be quickly optimized. MLP and CNN are both discriminative neural networks and have been widely used for the forward models. MLP is the simplest and most efficient DNN for geometries with low DOF such as empirical structures. With increasing DOF, both the number of training data samples and the depth of MLP structure have to be increased greatly. Data dimensionality reduction approaches can also be used to map high-DOF devices to low data dimension, but tricky data preprocessing is needed in that case. Another method is to merge data preprocessing directly into DNNs like CNN, whose convolutional layers and pooling layers are innately capable of processing high-dimensional data features. CNN is widely used for components with QR-code like structure or irregular structure.

Inverse models are more attractive for that they generate component geometry parameters for a target optical response almost without iterative optimization processes involved. However, an inverse model is much more difficult to build and train. Firstly, one optical response corresponds to not only one group of geometric parameters. Secondly, optical response usually has lower dimensionality as compared to geometry parameters. Luckily, generative neural networks such as CVAE and CGAN are capable of projecting low dimension to high dimension. Both of them try to monitor the underlying distributions of geometric parameters and optical response. The difference is that CVAE gets the underlying distribution via encoder and CGAN utilizes a discriminator to judge whether generated geometric parameters share the same distribution with data samples. This structural difference also brings training difference: CVAE trains the encoder and decoder simultaneously while CGAN trains two DNNs as generator and discriminator in an alternate way. After training, with optical response and potential variables as input, a CVAE generates device geometric parameters via a decoder while a CGAN produces geometric parameters via a generator.

Inverse models based on unsupervised generative neural networks are also appealing because they do not require training data samples. However, the time-consuming EM simulator has to be inserted into the generative neural networks to guarantee that they obey physical laws. Once trained, unsupervised generative neural networks can also generate new geometric parameters according to target optical response without any simulator involved.
References


Keywords

silicon photonics;inverse design;deep neural networks;optical neural networks

Retrieved from https://encyclopedia.pub/10403