

Inverse Design of Photonic Crystals through Automatic Differentiation

Momchil Minkov,^{*} Ian A. D. Williamson, Lucio C. Andreani, Dario Gerace, Beicheng Lou, Alex Y. Song, Tyler W. Hughes, and Shanhui Fan^{*}



Cite This: <https://dx.doi.org/10.1021/acsphotonics.0c00327>



Read Online

ACCESS |



Metrics & More



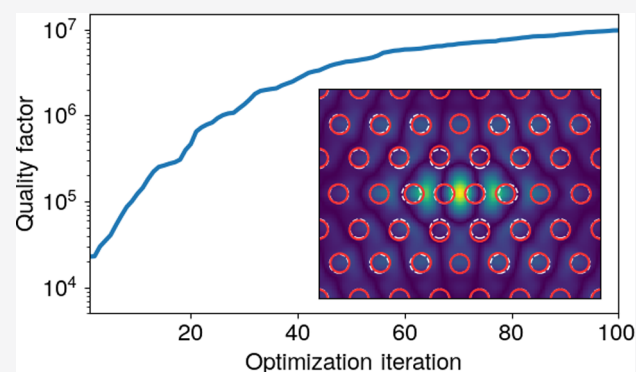
Article Recommendations



Supporting Information

ABSTRACT: Gradient-based inverse design in photonics has already achieved remarkable results in designing small-footprint, high-performance optical devices. The adjoint variable method, which allows for the efficient computation of gradients, has played a major role in this success. However, gradient-based optimization has not yet been applied to the mode-expansion methods that are the most common approaches to studying periodic optical structures such as photonic crystals. This is because, in such simulations, the adjoint variable method cannot be defined as explicitly as in standard finite-difference or finite-element time- or frequency-domain methods. Here, we overcome this gap through the use of automatic differentiation, which is a generalization of the adjoint variable method to arbitrary computational graphs. We implement the plane-wave expansion and the guided-mode expansion methods using an automatic differentiation library, and we show that the gradient of any simulation output can be computed efficiently and in parallel, with respect to all input parameters. We then use this implementation to optimize the dispersion of a photonic crystal waveguide, and the quality factor of an ultrasmall cavity in a lithium niobate slab. This extends photonic inverse design to an entirely new class of simulations, and more broadly highlights the importance that automatic differentiation could play in the future for tracking and optimizing complicated physical models.

KEYWORDS: *inverse design, photonic crystals, nanophotonics, resonators, waveguides*



INTRODUCTION

Tremendous flexibility in the control of the flow of light can be achieved by exploiting the vast number of degrees of freedom in photonic devices with wavelength-scale (or smaller) feature sizes. This flexibility promises the realization of compact and highly efficient integrated devices, which is becoming increasingly important for future photonic and optoelectronic technologies. To take advantage of the degrees of freedom in photonic devices, the field of photonic inverse design has emerged,¹ in which an optimization algorithm is used to automate the photonic design process toward a specified device performance, as characterized by an objective function. This has led to demonstrations of compact devices for routing, wavelength multiplexing, and spatial-mode converters.^{2–5} In addition, inverse design has been successfully extended to several nonlinear optical devices.^{6–8}

Gradient-based optimization is probably the most widely used technique in photonic inverse design. In such a technique, within each iteration, one first computes the gradient of the objective function, with respect to all the tunable parameters of the device. One then varies the parameters along the gradient direction to improve the performance of the device. Under-

pinning gradient-based inverse design is the adjoint variable method (AVM), which allows the gradient of a scalar objective function to be efficiently computed, with respect to many parameters of the device.^{9–11} AVM can be straightforwardly implemented when the optical devices are simulated by solving a linear system of equations of the form

$$\hat{A}\mathbf{e} = \mathbf{b}$$

where \hat{A} is the system matrix, \mathbf{e} the electromagnetic field distribution in the device to be solved for, and \mathbf{b} is the excitation source. In such a case, the gradient has a very straightforward physical interpretation as the interference between *forward* and *backward* fields. Such a physical interpretation is useful conceptually, but more importantly, it allows for the numerical implementation to use the same solver

Received: February 29, 2020

Published: June 9, 2020



for both the forward and backward simulations. Thus, once a simulation code is set up for the forward simulation, very little additional effort is required to implement the backward simulation. Therefore, the mathematical form of the backward simulation for computing the gradients is typically derived analytically and hard-coded.

However, there are scenarios in optical device simulations where the dependence of the objective function on the device parameters is more complex. For example, in the design of photonic crystals, one often solves an eigenvalue problem $\hat{A}\mathbf{e} = \lambda\mathbf{e}$, where λ is an eigenvalue and \mathbf{e} is the field of the corresponding eigenstate. The objective function is then expressed in terms of a mathematical function of some of the eigenvalues and eigenmodes. This has been done in the case when \hat{A} is just the finite-difference matrix describing the Maxwell problem on a spatial grid;^{12,13} however, in a mode-expansion method, the matrix \hat{A} itself will typically have some functional dependence on the structural parameters. This is determined by the definition of the expansion basis and could be highly nontrivial, and must be taken into account in the gradient. Other scenarios that can complicate the dependence of the final objective function on the physically tunable parameters include complex parametrizations of the simulation domain, or complex objective functions reflecting multiple goals and/or constraints. Considering the most general combinations of parametrization, simulation, and objective function definition, explicit by-hand implementation of the backward gradient computation can become difficult or even infeasible. Therefore, there is a need to systematically implement AVM for arbitrary dependency of the objective function on the device parameters.

In the context of computational science, automatic differentiation (AD) is the application of the adjoint variable method to arbitrary computational graphs. Within an AD-enabled programming framework, a software developer only needs to define the forward computation, while the backward computation is generated automatically by tracing program execution or ahead of time via source code analysis. At the heart of an AD framework are gradient-aware elementary functions, which, in essence, can each be represented in terms of their own individual adjoint variable problem, much like the explicitly defined backward simulations used in optical inverse design. However, the key advantage of an AD framework lies in its ability to flexibly compose such elementary functions to build far more complex computations with end-to-end gradient support.

Over the past several decades, automatic differentiation has been explored across many contexts.^{14–19} However, in recent years, the growing interest in machine learning, and particularly gradient-based model training, has driven the development of modern automatic differentiation libraries. Several examples include Autograd,²⁰ PyTorch,²¹ TensorFlow,²² Zygote,²³ and JAX.²⁴ While these libraries have been primarily applied to machine learning, their application to problems in computational physics has only recently been explored.^{25–29} In the context of optical inverse design, an AD-enabled finite difference frequency domain (FDFD) simulation framework was recently proposed,²⁸ which leveraged AD for flexible composition of optimization objective functions and device parametrizations. Similar goals have motivated the development of application-specific differentiable graph frameworks for optical inverse design.³⁰ However, an unexplored application of modern AD frameworks within optical inverse

design is to enable gradient computations through optical simulations that do not have straightforwardly defined backward problems. An example class of such simulations are mode expansion methods used for computing the photonic bands of periodic optical structures.

In this work, we present a differentiable implementation of the two-dimensional (2D) plane-wave expansion (PWE) method used for simulating a 2D photonic crystal (PhC), as well as of the guided-mode expansion (GME) method used for efficiently simulating photonic crystal slabs.³¹ We demonstrate that AD allows us to efficiently compute the gradient of a scalar-valued objective function derived from any output quantities (e.g., eigenmode dispersion, field profile, and/or loss rates) simultaneously, with respect to all the input parameters (e.g., position and size of holes, slab permittivity, and/or thickness). Our approach is equivalent to an adjoint variable method for the eigenmodes of periodic structures, without the need to derive the entire adjoint gradient computation by hand. Our implementation uses the open source AD package Autograd,²⁰ which allows for great flexibility both in the parametrization of the periodic structures and in the final objective function. Our open-source code has been made available online.³² We also show two examples of gradient-based optimizations performed using this method. In the first example, we optimize the dispersion of a PhC waveguide toward several target curves, which is an important problem for nonlinear optics³³ and slow light³⁴ applications. In the second example, we optimize the quality factor (Q) of a PhC cavity in a lithium niobate slab with an ultrasmall volume, which can be useful for many applications in integrated photonics.³⁵

■ THEORETICAL PRELIMINARIES

The results presented in this paper are at the intersection of two concepts: automatic differentiation and mode-expansion methods for electromagnetic simulations. In this section, we lay down a theoretical foundation for both concepts. We note that, throughout this paper, we denote column (“contravariant”) vectors in bold (e.g., \mathbf{v}) and 2D arrays (matrices) using a format such as \hat{M} . Derivatives of scalar functions, with respect to a vector ($d/d\mathbf{v}$), are therefore row (“covariant”) vectors. Similarly, derivatives of vectors, with respect to vectors, are matrices, and so on for tensors of higher rank.

Automatic Differentiation. The field of automatic differentiation (AD) has existed for more than five decades^{18,36,37} and has covered a wide range of applications. Our goal here is only to provide a summary targeted at readers with little or no prior knowledge of the topic, and hence many details are omitted for the sake of brevity. The first important point is that AD is a computer programming paradigm and not a purely mathematical construct. That is to say, unlike symbolic differentiation, AD is inextricably connected to an underlying computer program, which can be generically represented as a computational graph, as shown in Figure 1a. In the schematic, the rectangles denote generic functions included in a programming library, while vectors \mathbf{x}_i contain all the inputs and outputs of these functions. Thus, the *forward computation* of the program illustrated in Figure 1a performs the operations

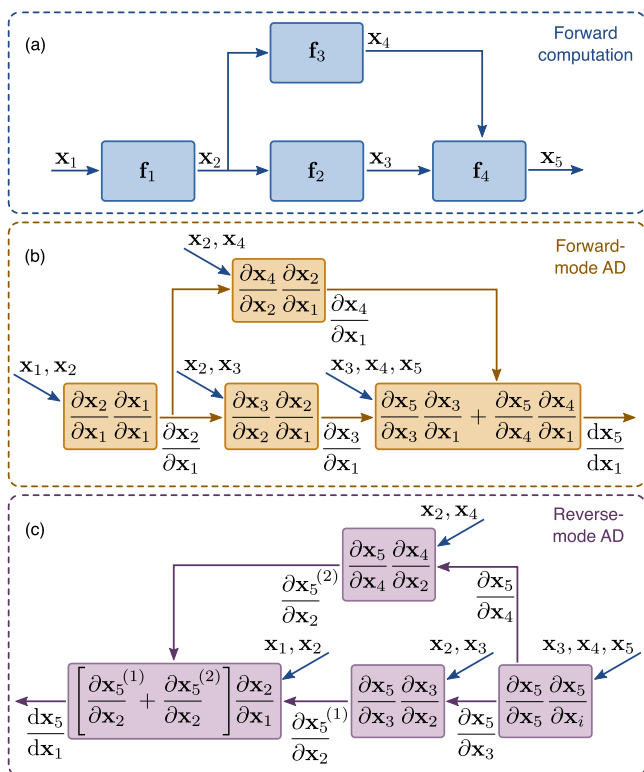


Figure 1. (a) An example computer program represented as a computational graph. (b) Forward-mode (FM) differentiation graph associated with panel (a). (c) Reverse-mode (RM) differentiation graph associated with panel (a). In panels (b) and (c), the blue arrows denote inputs from the forward computation of panel (a).

$$\begin{aligned} \mathbf{x}_2 &= \mathbf{f}_1(\mathbf{x}_1) \\ \mathbf{x}_3 &= \mathbf{f}_2(\mathbf{x}_2) \\ \mathbf{x}_4 &= \mathbf{f}_3(\mathbf{x}_2) \\ \mathbf{x}_5 &= \mathbf{f}_4(\mathbf{x}_3, \mathbf{x}_4) \end{aligned}$$

An AD library then allows the user to compute the *exact* Jacobian $\mathrm{d}\mathbf{x}_i/\mathrm{d}\mathbf{x}_j$ for any i, j , using the rules of differentiation and some knowledge of the partial derivatives of each operation (more discussion on this topic is given below). For example, in the case of the computational graph of Figure 1a, we have

$$\frac{\mathrm{d}\mathbf{x}_5}{\mathrm{d}\mathbf{x}_1} = \left[\frac{\partial \mathbf{x}_5}{\partial \mathbf{x}_4} \frac{\partial \mathbf{x}_4}{\partial \mathbf{x}_2} + \frac{\partial \mathbf{x}_5}{\partial \mathbf{x}_3} \frac{\partial \mathbf{x}_3}{\partial \mathbf{x}_2} \right] \frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} \quad (1)$$

which is, in itself, a computational graph that can be traced by the program. A crucial point in AD is that this tracing is dependent on the order in which the products in eq 1 are computed (the “accumulation” of the Jacobian), which also affects the computational complexity of evaluating eq 1. There are two main approaches. The first one is to work from right to left, which is called forward-mode (FM), because the Jacobian accumulation, illustrated in Figure 1b, follows the arrows of the original computational graph of Figure 1a. This is conceptually straightforward, as, for example, fan-out of the outputs (e.g., x_2 in Figure 1(a)) leads to fan-out of the derivative, while fan-in of the inputs (e.g., x_3 and x_4 in Figure 1a) leads to the addition of derivatives. The alternative approach is to work from left to right in eq 1, which is called reverse mode (RM), since the

computational graph (Figure 1c) flows in the opposite direction of the arrows in Figure 1a. In this case, everything is reversed, and fan-out of the output leads to the addition of the input derivatives ($\partial \mathbf{x}_5^{(1)}/\partial \mathbf{x}_2$ and $\partial \mathbf{x}_5^{(2)}/\partial \mathbf{x}_2$ in Figure 1c), while fan-in of the inputs leads to splitting of the output derivatives ($\partial \mathbf{x}_5/\partial \mathbf{x}_4$ and $\partial \mathbf{x}_5/\partial \mathbf{x}_3$ in Figure 1c). More generally, a combination of FM and RM is possible, but this is not typically used, because of the extra complexity.

The construction of the computational graph for evaluating the derivatives is one of the main components of an AD library, and there are various ways in which it can be implemented. The specifics of this lie beyond the scope of our discussion, but the important point is that once the graph is constructed, the derivative computation in both forward mode (FM) and RM accumulation becomes a sequence of elementary building-block operations, which we discuss in more detail below.

The FM derivative computation corresponding to Figure 1a is shown in Figure 1b. There, brown arrows indicate gradient flow (i.e., input and output of partial derivatives $\partial \mathbf{x}_i/\partial \mathbf{x}_j$), while blue arrows indicate inputs from the forward computation (i.e., \mathbf{x}_i). We see that the primitive building block of the forward-mode computation in Figure 1b is the operation

$$g_{\text{FM}}\left(\mathbf{x}_i, \mathbf{x}_j, \frac{\partial \mathbf{x}_j}{\partial \mathbf{x}_i}\right) = \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_j} \frac{\partial \mathbf{x}_j}{\partial \mathbf{x}_i} \quad (2)$$

Notice that, in eq 2, the right-most differentiation always has \mathbf{x}_1 in the denominator. Then, computing $\mathrm{d}\mathbf{x}_5/\mathrm{d}\mathbf{x}_1$ in our example requires N_1 separate evaluations of the graph of Figure 1b, where N_1 is the number of elements in \mathbf{x}_1 . As a general rule, the computational time of forward-mode AD scales linearly with the number of *input* parameters. In many implementations (including Autograd), this is made explicit by considering the forward accumulation of a single derivative vector $\partial \mathbf{x}_i/\partial \mathbf{x}$, and associating a Jacobian-vector product (JVP) function to every function $\mathbf{f}(\mathbf{x})$, such that

$$\text{jvp}(\mathbf{x}, \mathbf{f}(\mathbf{x}), \mathbf{v}) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{v} \quad (3)$$

This is essentially the directional derivative of a vector-valued function \mathbf{f} in the direction of \mathbf{v} . The vector \mathbf{v} that enters eq 3 is a single column of the Jacobian $\partial \mathbf{x}_i/\partial \mathbf{x}_1$ that appears in eq 2. Given a set of functions with correspondingly defined jvp-s, arbitrarily complex programs can be differentiated using forward-mode AD.

The RM derivative accumulation for the same computation is shown in Figure 1c. We see that the primitive building block is the operation

$$g_{\text{RM}}\left(\mathbf{x}_i, \mathbf{x}_j, \frac{\partial \mathbf{x}_5}{\partial \mathbf{x}_i}\right) = \frac{\partial \mathbf{x}_5}{\partial \mathbf{x}_i} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_j} \quad (4)$$

Here, the left-most differentiation always has \mathbf{x}_5 in the numerator. Then, computing $\mathrm{d}\mathbf{x}_5/\mathrm{d}\mathbf{x}_1$ in our example requires N_5 separate evaluations of the graph shown in Figure 1c, and, as a general rule, the computational time of RM AD scales linearly with the number of *outputs* of the program. One possible implementation of eq 4 is to associate a vector-Jacobian product (VJP) to every function $\mathbf{f}(\mathbf{x})$, such that

$$\text{vjp}(\mathbf{x}, \mathbf{f}(\mathbf{x}), \mathbf{v}) = \mathbf{v}^T \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \quad (5)$$

There are two crucial differences between FM and RM AD. First, regardless of the implementation, the computational time for $\mathrm{d}\mathbf{x}_o/\mathrm{d}\mathbf{x}_i$ scales with N_o when using RM and with N_i when using FM. Here, the “output” vector \mathbf{x}_o and the “input” vector \mathbf{x}_i can be any of the variables along the computational graph, i.e., any one of $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_5$ in the example of Figure 1a. Thus, when $N_o \gg N_i$, FM is much faster than RM, and vice versa if $N_i \gg N_o$. Since optimization problems typically have a large number of input parameters and a single scalar-valued objective function as output, RM AD is then substantially better for these problems. This explains the ubiquitous use of backpropagation in machine learning and of the adjoint variable method in engineering (both are specific examples of RM AD). However, the second difference works against RM AD, and can sometimes be a limiting factor. Specifically, note that all intermediate results from the forward computation (blue arrows in Figures 1b and 1c) are needed for the derivative computation in both FM AD and RM AD. Since the forward-mode AD can be done in parallel with the forward computation, these do not need to be stored beyond every individual step. In contrast, the reverse-mode computation can only start after the forward computation is complete, and it requires that all intermediate results are stored. This could lead to a significantly higher memory requirement, which has motivated the development of checkpointing schemes;³⁸ however, a detailed discussion of such schemes is beyond the scope of this paper. That being said, the computational time advantage of reverse-mode AD is so compelling that the typical approach is to accommodate its memory disadvantage in some way.

The Autograd library is API-compatible with a large subset of the NumPy and SciPy libraries.^{39,40} Thus, complex programs that include algebraic operations, array manipulations, and flow control can be traced and automatically differentiated with ease. The end user only needs to write the forward computation (as in Figure 1a), and the differentiation is handled automatically. This gives enormous flexibility in defining objective functions and parametrizations in numerical simulations and optimizations. In addition, external functions can also be included if the JVP and/or VJP can be defined. We will illustrate this below with a specific example of a function that will be needed for the guided-mode expansion. Furthermore, this same example captures the essence of the AVM for finite-difference frequency-domain electromagnetic simulations for both linear and nonlinear systems.^{9–11,28}

Assume that the output \mathbf{x}_o of an operation is defined through a set of constraints written most generally as

$$\mathbf{f}(\mathbf{x}_i, \mathbf{x}_o) = 0 \quad (6)$$

We assume that (i) \mathbf{f} has the same dimensionality as \mathbf{x}_o and (ii) a solution (not necessarily unique) exists and can be found through some iterative numerical method. Then, \mathbf{x}_o is an implicit function of the inputs \mathbf{x}_i ; using the differentiation rule for implicit functions, we get

$$\frac{\partial \mathbf{x}_o}{\partial \mathbf{x}_i} = - \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}_o} \right]^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{x}_i} \quad (7)$$

where $[\partial \mathbf{f} / \partial \mathbf{x}_o]^{-1}$ is the matrix inverse of the square Jacobian matrix $\partial \mathbf{f} / \partial \mathbf{x}_o$. Since we are mostly interested in RM AD, we will illustrate the VJP computation for this function, which is

$$\mathbf{v}_o = \mathbf{v}_i^T \frac{\partial \mathbf{x}_o}{\partial \mathbf{x}_i} = - \mathbf{v}_i^T \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}_o} \right]^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{x}_i} \quad (8)$$

Thus, the output \mathbf{v}_o of the VJP can be found using the auxiliary variable \mathbf{v}_a that is found as the solution to the linear system of equations

$$\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}_o} \right]^T \mathbf{v}_a = - \mathbf{v}_i \quad (9)$$

and reads

$$\mathbf{v}_o = \mathbf{v}_a^T \frac{\partial \mathbf{f}}{\partial \mathbf{x}_i} \quad (10)$$

Note that \mathbf{v}_o has the same dimension as \mathbf{x}_i , while \mathbf{v}_i and \mathbf{v}_a have the same dimension as \mathbf{x}_o , since input and output dimensions are switched in RM AD.

The essence of the adjoint variable method for an electromagnetic finite-difference frequency domain simulation is captured by eqs 9 and 10. In such a simulation, we have a discretized real-space domain and an input vector $\mathbf{x}_i = \epsilon$ defining the dielectric permittivity at every point. The output is the corresponding electric field $\mathbf{x}_o = \mathbf{e}$ at every point, which is obtained by solving for $\mathbf{f}(\epsilon, \mathbf{e}) = 0$, where \mathbf{f} defines the Maxwell problem. For a linear electromagnetic system, $\mathbf{f}(\epsilon, \mathbf{e}) = \hat{A}(\epsilon)\mathbf{e} - \mathbf{b}$, with $\hat{A}(\epsilon)$ being the matrix defining the single-frequency linear Maxwell's equations on a discretized spatial grid, and \mathbf{b} is a current source term.^{9–11} In the case of, e.g., Kerr-type nonlinearity, $\mathbf{f}(\epsilon, \mathbf{e}) = \hat{A}(\epsilon, \mathbf{e})\mathbf{e} - \mathbf{b}$ and a nonlinear numerical solver is required.²⁸ However, the VJP in both cases works in exactly the same way, as defined in eq 10. The auxiliary variable \mathbf{v}_a is called the adjoint field in the AVM terminology in photonics. This is because, for a linear system, $\partial \mathbf{f} / \partial \epsilon = \hat{A}$, and eq 9 can be interpreted as a Maxwell problem for fields \mathbf{v}_a propagating in an adjoint system described by \hat{A}^T (which is the same as \hat{A} for reciprocal materials). However, more generally, forward and adjoint variables would typically obey different equations and cannot be interpreted on the same footing.

Also note that any mathematical operation can be framed as a constraint, e.g., $\mathbf{x}_o = \mathbf{x}_i^2$ is equivalent to $\mathbf{f}(\mathbf{x}_i, \mathbf{x}_o) = \mathbf{x}_i^2 - \mathbf{x}_o = 0$. Therefore, any VJP can be derived through the implicit function theorem and eqs 6–10. In that sense, RM AD is sometimes derived within the framework of Lagrange multipliers, which end up corresponding to the adjoint variables.³⁷

In conclusion, we wish to point out that automatic differentiation, and specifically RM AD, is more than just a numerical subtlety with limited importance. On the contrary, when considering optimization problems, the method is computationally faster than *both* numerical differentiation (i.e., finite-difference methods) and symbolic differentiation (deriving by hand, and evaluating the final expression for the derivative). The improvement is due to the use of intermediate “adjoint” variables. The second important advantage of AD is splitting operations into elementary building blocks, which allows arbitrarily complex programs to be differentiated. For these reasons, the development and use of backpropagation, which corresponds to reverse-mode AD, has been the key driver of the recent machine learning revolution,¹⁸ and we believe that it will be a key component of tackling optimization problems in physics and engineering.

Mode Expansion Methods. In this section, we review the general idea of mode expansion methods applied to Maxwell's

equations. The plane-wave expansion and the guided-mode expansion are both specific examples of the formalism we introduce below.

In the absence of free charges and currents, Maxwell's equations can be written as an eigenvalue problem for the electric (\mathbf{E}) and magnetic (\mathbf{H}) fields with harmonic time-dependence,⁴¹ $\mathbf{E}(\mathbf{r}, t) = \mathbf{E}(\mathbf{r})e^{-i\omega t}$, $\mathbf{H}(\mathbf{r}, t) = \mathbf{H}(\mathbf{r})e^{-i\omega t}$. If we further assume a linear, isotropic, lossless, nondispersive, and nonmagnetic medium, which is a good approximation for many real-world materials, the eigenvalue equation can be written for \mathbf{H} alone:

$$\hat{\Theta}\mathbf{H}(\mathbf{r}) \equiv \left[\nabla \times \frac{1}{\epsilon(\mathbf{r})} \nabla \times \right] \mathbf{H}(\mathbf{r}) = \frac{\omega^2}{c^2} \mathbf{H}(\mathbf{r}) \quad (11)$$

together with the constraint $\nabla \cdot \mathbf{H}(\mathbf{r}) = 0$. In eq 11, $\epsilon(\mathbf{r})$ is the relative dielectric permittivity distribution that fully defines the system. For a known magnetic field profile, the electric field can be found through

$$\mathbf{E}(\mathbf{r}) = \frac{ic}{\omega\epsilon(\mathbf{r})} \nabla \times \mathbf{H}(\mathbf{r}) \quad (12)$$

For a closed system in which all fields decay as $|\mathbf{r}| \rightarrow \infty$, or for a periodic system, we can define an inner product on the space of magnetic field functionals as

$$(\mathbf{H}_\nu, \mathbf{H}_\mu) = \int d\mathbf{r} \mathbf{H}_\nu^\dagger(\mathbf{r}) \mathbf{H}_\mu(\mathbf{r}) \quad (13)$$

such that the operator $\hat{\Theta}\mathbf{H}(\mathbf{r})$ is Hermitian, i.e., $(\hat{\Theta}\mathbf{H}_\nu, \mathbf{H}_\mu)^* = (\mathbf{H}_\nu, \hat{\Theta}\mathbf{H}_\mu)$. For an open system, the integration of eq 13 is not well-defined, and correctly defining an inner product is an active area of study that goes beyond the scope of this work.⁴² Here, we will assume either periodic or decaying fields, and compute concrete examples of eq 13 in the sections on PWE and GME below.

The idea of a mode expansion method is to express the eigenstates of eq 11 for an arbitrary operator $\hat{\Theta}$ on the basis of the eigenstates of a different operator $\hat{\Theta}_0$, i.e.,

$$\mathbf{H}(\mathbf{r}) = \sum_\mu c_\mu \mathbf{H}_\mu(\mathbf{r}) \quad (14)$$

where $\hat{\Theta}_0\mathbf{H}_\mu = (\omega_\mu^2/c^2)\mathbf{H}_\mu$ and $\nabla \cdot \mathbf{H}_\mu = 0$ holds for all μ . In other words, the modes \mathbf{H}_μ are solutions to Maxwell's equations for a given structure, which is typically simple to solve, e.g., $\epsilon(\mathbf{r}) = 1$ everywhere (free space). Because the problem is Hermitian, the eigenmodes form an orthonormal set, such that

$$(\mathbf{H}_\nu, \mathbf{H}_\mu) = \delta_{\nu\mu} \quad (15)$$

Using this, plugging eq 14 into eq 11, multiplying by \mathbf{H}_ν^* and taking the inner product on both sides, we get

$$\left(\mathbf{H}_\nu, \hat{\Theta} \sum_\mu c_\mu \mathbf{H}_\mu \right) = \sum_\mu c_\mu \frac{\omega^2}{c^2} (\mathbf{H}_\nu, \mathbf{H}_\mu) \quad (16)$$

and so

$$\sum_\mu \mathcal{H}_{\nu\mu} c_\mu = \frac{\omega^2}{c^2} c_\nu \quad (17)$$

This is an eigenproblem for the expansion coefficients c_μ with matrix elements $\mathcal{H}_{\nu\mu} = (\mathbf{H}_\nu, \hat{\Theta}\mathbf{H}_\mu)$.

We note that the expansion of eq 14 is exact only when \mathbf{H}_μ forms a complete set, and all modes are included in the summation. However, any starting structure $\hat{\Theta}_0$ has an infinite number of eigenmodes, with increasing magnitude of the eigenfrequency. In practice, the summation will thus always be truncated to a finite subset of the basis functions. Apart from that, the set of plane waves used is complete, and can be thought of as Fourier decomposition of a vector-valued function. On the other hand, in the guided-mode expansion method, we restrict the summation to the fully guided slab modes, which is an *incomplete* set. The method is thus only approximate, but it has been shown to compare well to first-principles simulations for many different structures, with a much faster computational speed.^{43–45}

PLANE-WAVE EXPANSION

The plane-wave expansion method for simulating periodic structures is well-known both in photonics and in quantum mechanics and other domains. Here, we will nevertheless review the fundamentals, because this helps with the understanding of the guided-mode expansion, as well as the understanding of our automatic differentiation implementation of the two methods.

Method Description. In this paper, we focus on 2D PWE, in which the simulation domain is assumed to be periodic in two directions and translationally invariant in the third. One example of such a structure is illustrated in Figures 2a and 2b

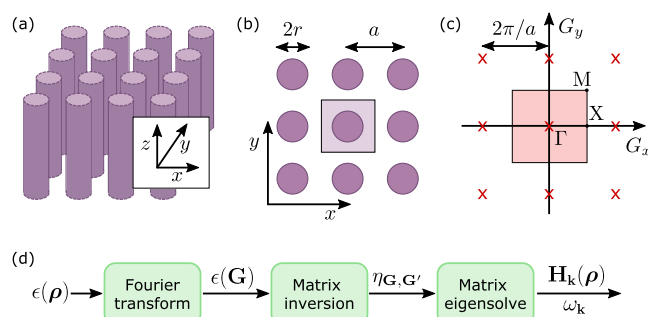


Figure 2. (a) A 2D photonic crystal composed of dielectric rods in air. The rods form a square lattice in the xy -plane, and are assumed to be infinite in the z -direction. (b) Cross-section in the xy -plane showing the lattice constant a and rod radius r . The shaded square shows the primitive cell of the crystal. (c) Reciprocal lattice for this crystal, with reciprocal lattice vectors separated by $2\pi/a$ in both directions. The shaded square shows the primitive cell, i.e., the first Brillouin zone, and the high-symmetry points Γ , X , and M are noted. (d) Computational graph for the 2D plane-wave expansion.

and consists of an array of dielectric rods in air, assumed to be infinitely extended in the z -direction. Most generally, the Bloch theorem then states that the eigenmodes of the structure can be written as

$$\mathbf{H}_\mathbf{k}(\rho) = e^{ik_z z} e^{i\mathbf{k} \cdot \rho} \tilde{\mathbf{H}}_{\mathbf{k},k_z}(\rho) \quad (18)$$

where the Bloch momentum \mathbf{k} and the position vector ρ lie in the xy -plane, and $\tilde{\mathbf{H}}_{\mathbf{k},k_z}(\rho)$ is periodic in xy with the 2D lattice periodicity. As is usual in the study of 2D PhCs, we will focus

on modes with $k_z = 0$ and drop that label. The method is easy to generalize for nonzero k_z .

The inner product (eq 13) for this 2D problem reads

$$(\mathbf{H}_\nu, \mathbf{H}_\mu) = \frac{1}{S} \int_S d\rho \mathbf{H}_\nu^\dagger(\rho) \mathbf{H}_\mu(\rho) \quad (19)$$

where S denotes the primitive cell in the xy -plane. With this, we can apply the general mode-expansion procedure. Because of the reflection symmetry, with respect to the xy -plane, all modes can be separated into transverse electric (TE) and transverse magnetic (TM) polarizations, even in the case of arbitrary in-plane permittivity $\epsilon(\rho)$. The plane-wave basis is just the solution to the free-space system with a relative permittivity of $\epsilon(\rho) = 1$ everywhere. This can be written as $\mathbf{H}_\mathbf{k}(\rho) = \mathbf{H}_\mathbf{k}^p e^{i\mathbf{k}\cdot\rho}$, where $\mathbf{H}_\mathbf{k}^p = \hat{z}$ for TE polarization, while $\mathbf{H}_\mathbf{k}^p = \hat{e}_\mathbf{k}$ for TM polarization. Here, \hat{z} and $\hat{e}_\mathbf{k}$ are unit vectors pointing in the z -direction and in the direction orthogonal to both z and \mathbf{k} , respectively. Thus, the p -polarization eigenstates for an arbitrary 2D PhC can be written as

$$\mathbf{H}_\mathbf{k}^p(\rho) = \sum_{\mathbf{G}} c_{\mathbf{k}}^p(\mathbf{G}) \mathbf{H}_{\mathbf{G}+\mathbf{k}}^p e^{i(\mathbf{G}+\mathbf{k})\cdot\rho} \quad (20)$$

where \mathbf{G} is a reciprocal lattice vector, as shown in Figure 2c. As derived in the Supporting Information (SI), the matrix elements defined in eq 17 for the two polarizations read

$$\mathcal{H}_{\mathbf{g},\mathbf{g}'}^{\text{TE}} = [(\mathbf{k} + \mathbf{G}) \cdot (\mathbf{k} + \mathbf{G}')] \eta_{\mathbf{G},\mathbf{G}'} \quad (21)$$

$$\mathcal{H}_{\mathbf{g},\mathbf{g}'}^{\text{TM}} = |\mathbf{k} + \mathbf{G}| |\mathbf{k} + \mathbf{G}'| \eta_{\mathbf{G},\mathbf{G}'} \quad (22)$$

where we define $\mathbf{g} = \mathbf{G} + \mathbf{k}$, and the 2D Fourier transform of the inverse of the permittivity distribution by the matrix elements

$$\eta_{\mathbf{G},\mathbf{G}'} = \frac{1}{S} \int_S d\rho \frac{1}{\epsilon(\rho)} e^{-i(\mathbf{G}-\mathbf{G}')\cdot\rho} \quad (23)$$

In the expansion of eq 20, we only include a discrete set of plane waves defined by the reciprocal lattice vectors, because the Fourier components in eq 23 are zero for any plane-wave combination that is noncommensurate with the lattice periodicity.

Numerically, we truncate the expansion up to some maximum value G_{max} such that $|\mathbf{G}| \leq G_{\text{max}}$. Furthermore, we note that it has been shown that, because of the discontinuous nature of the permittivity, a significantly better convergence of the expansion is achieved if the Fourier transform $\epsilon(\mathbf{G})$ of $\epsilon(\rho)$ is computed first, and $\hat{\eta}$ is computed as the matrix inverse of the matrix $\epsilon_{\mathbf{G},\mathbf{G}'} = \epsilon(\mathbf{G} - \mathbf{G}')$.^{46,47} A high-level computational graph for the full plane-wave expansion is then presented in Figure 2d, showing the three main operations needed to compute the eigenmodes of an arbitrary 2D PhC. These are Fourier transform of the real-space permittivity to compute $\epsilon(\mathbf{G})$, matrix inversion to compute $\eta_{\mathbf{G},\mathbf{G}'}$, and matrix diagonalization of $\hat{\mathcal{H}}$ as defined in eqs 21 and 22 for the two polarizations. Next, we describe the automatic differentiation aspect of the PWE method.

Automatic Differentiation. As emphasized in our discussion on automatic differentiation, RM AD is the method of choice for optimization problems, which is why, from this point forward, we only focus on this method. Thus, we discuss the derivative propagation through the operations of Figure 2d in reverse order, starting with the matrix eigensolve.

Generically, we denote the forward operation as a function $\text{eigh}(\hat{\mathcal{H}})$ that takes a Hermitian matrix $\hat{\mathcal{H}}$ as an input and returns a set of eigenvalues E_i and associated eigenvectors \mathbf{c}_i , such that

$$\hat{\mathcal{H}} \mathbf{c}_i = E_i \mathbf{c}_i, \quad \forall i \quad (24)$$

In the current version of Autograd (1.3), the eigh operation is only implemented for real symmetric matrices. Therefore, we extended the library by defining the RM AD step for the eigensolve of a Hermitian matrix. This operation can be derived using matrix algebra,^{48,49} or using eqs 6 and 7 and treating eq 24 as a constraint. Below, we give yet another, more intuitive, derivation, using perturbation theory from quantum mechanics.

We assume that there are no degenerate eigenvalues. Degenerate cases can also be handled by perturbation theory, but this requires extra considerations and is neglected for simplicity. Then, the eigenvalues can be labeled (e.g., sorted by magnitude), and a derivative of a given eigenvalue, with respect to a given matrix element, is

$$\frac{\partial E_i}{\partial \mathcal{H}_{\mu\nu}} = \lim_{\lambda \rightarrow 0} \frac{E_i(\hat{\mathcal{H}} + \lambda \hat{1}_{\mu\nu}) - E_i(\hat{\mathcal{H}})}{\lambda} \quad (25)$$

where $E_i(\mathcal{H})$ is the i th eigenvalue of matrix $\hat{\mathcal{H}}$, and $\hat{1}_{\mu\nu}$ is a matrix with the same dimension as $\hat{\mathcal{H}}$ with zeros everywhere apart from the μ, ν element, which is set to 1. Since we take a limit of $\lambda \rightarrow 0$, we can use first-order perturbation theory as an exact result:

$$E_i(\hat{\mathcal{H}} + \lambda \hat{1}_{\mu\nu}) = E_i(\hat{\mathcal{H}}) + \lambda \mathbf{c}_i^\dagger \hat{1}_{\mu\nu} \mathbf{c}_i \quad (26)$$

Therefore,

$$\frac{\partial E_i}{\partial \mathcal{H}_{\mu\nu}} = \lim_{\lambda \rightarrow 0} \frac{\lambda \mathbf{c}_i^\dagger \hat{1}_{\mu\nu} \mathbf{c}_i}{\lambda} = c_{i,\mu}^* c_{i,\nu} \quad (27)$$

where $c_{i,\mu}$ is the μ th element of \mathbf{c}_i . The VJP for the eigensolve can then be written as a matrix with elements

$$\text{vjp}(\hat{\mathcal{H}}, \text{eigh}(\hat{\mathcal{H}}), \mathbf{v})_{\mu\nu} = \sum_i v_i c_{i,\mu}^* c_{i,\nu} \quad (28)$$

We can also use first-order perturbation theory to compute derivatives of the eigenvectors:

$$\begin{aligned} \frac{\partial \mathbf{c}_i}{\partial \mathcal{H}_{\mu\nu}} &= \lim_{\lambda \rightarrow 0} \frac{\mathbf{c}_i(\mathcal{H} + \lambda \hat{1}_{\mu\nu}) - \mathbf{c}_i(\mathcal{H})}{\lambda} \\ &= \lim_{\lambda \rightarrow 0} \sum_{j \neq i} \frac{\mathbf{c}_j^\dagger \lambda \hat{1}_{\mu\nu} \mathbf{c}_i}{\lambda(E_i - E_j)} \mathbf{c}_j = \sum_{j \neq i} \frac{c_{j,\mu}^* c_{i,\nu}}{E_i - E_j} \mathbf{c}_j \end{aligned} \quad (29)$$

and write the corresponding VJP.

Something very important to notice is the fact that the derivative of an eigenvalue is dependent only on its corresponding eigenvector. On the other hand, the derivative of an eigenvector as defined in eq 29 is dependent on all other eigenvectors of the matrix, which could have important implications. In particular, if an objective function is dependent on any of the eigenvectors, *all* of them are needed to propagate the gradient *exactly*. However, computing all eigenvectors is difficult when using iterative methods that return only a few eigenvectors.⁵⁰ Still, the dependence of the denominator in eq

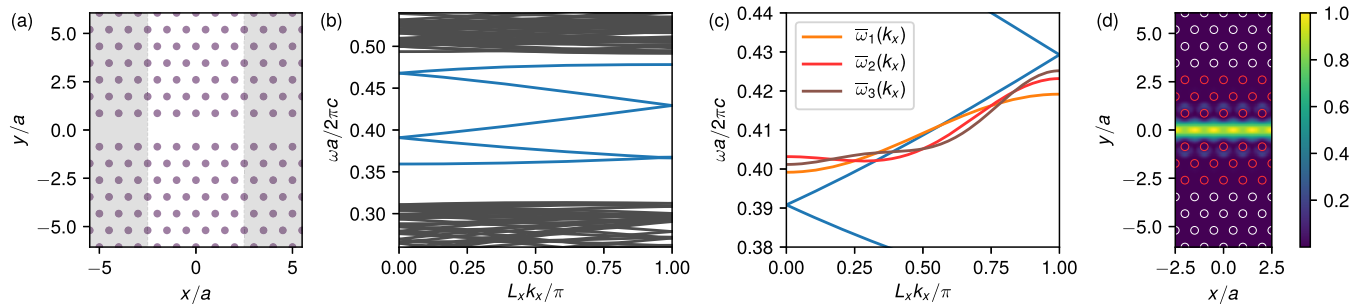


Figure 3. (a) Photonic crystal waveguide made by a missing row of pillars in a hexagonal lattice of dielectric pillars of relative permittivity $\epsilon = 9$ and radius $r = 0.2a$, with a being the lattice constant. The white region of width $L_x = 5a$ in the x -direction denotes the supercell used in the plane-wave expansion simulations. (b) Photonic bands of the structure; waveguide bands are shown in blue. (c) Expanded view over the middle of the waveguide-band region, and three different target dispersion curves that we optimize for. (d) Electric field intensity for the mode at $L_x k_x / \pi = 0.5$ for the blue band in panel (c). The positions and radii of the pillars marked in red are optimized to achieve the target dispersion curves in panel (c).

29 on $E_i - E_j$ means that eigenvectors with very different eigenvalues should contribute less to the gradient. It is also possible to define the gradient propagation for the eigenvectors in a way that is exact without the need for other eigenvectors,⁵¹ and we have also included this option in our software package.³²

The Hermitian eigenvalue decomposition is the only operation that required extending the existing Autograd API. All other operations, including several low-level algebraic operations that are not shown in Figure 2d, are already supported by Autograd. For completeness, here, we briefly discuss the other two main operations of the computational graph shown in Figure 2d. First, the AD step through the matrix inverse can be derived starting from $d\hat{A}^{-1}/dp = -\hat{A}^{-1}d\hat{A}/dp\hat{A}^{-1}$, and the VJP reads⁴⁸

$$\text{vjp}(\hat{A}, \hat{A}^{-1}(\hat{A}), \hat{V}) = -\hat{A}^{-1}\hat{V}^T\hat{A}^{-1} \quad (30)$$

where \hat{V} is now a matrix of the same size as \hat{A} . This is a convenient way to write the expression that can be assumed without loss of generality—strictly speaking, we can always “flatten” \hat{A} and \hat{V} into vectors to more rigorously match the formalism that we defined for AD. Second, regarding the Fourier transform, one way that this can be implemented in practice is to define $\epsilon(\rho)$ on a discrete grid in real space, and use a discrete Fourier transformation. In that case, we simply have $\epsilon(\mathbf{G}) = \hat{D}\epsilon(\rho)$, where \hat{D} is the corresponding Fourier transform matrix, and the VJP through this linear operation is straightforward, $\mathbf{v}_o^T = \mathbf{v}_i^T \hat{D}$.

In our implementation,³² we take a different approach to the parametrization of the permittivity. Namely, we define the structure through *shape* primitives, such as circles and polygons. Then, the Fourier transform can be computed through simple algebraic operations^{31,52} that are straightforward to differentiate through. For circles, the Fourier transform also includes the Bessel function of the first kind, whose derivative is analytic and the function is already included in Autograd. In the following section, we demonstrate this parametrization in practice, as well as the entire formalism developed thus far.

Waveguide Optimization. In this section, we apply the PWE and AD formalism developed in the previous sections to a practical design problem. Namely, we optimize the dispersion $\omega(\mathbf{k})$ of a PhC waveguide to match various predefined target forms. Dispersion engineering is generally important for many practical applications, including nonlinear phase matching³³

and generation of frequency combs.⁵³ Here, for illustrative purposes, we use generic target dispersion curves. The starting structure is shown in Figure 3a and consists of a row of missing holes in a hexagonal lattice of dielectric rods in air. The underlying bulk PhC has a broad band gap for TM-polarized modes, and the line defect introduces a guided band inside the band gap, as shown in Figure 3b. Note that the guided band (blue) is folded because we use a supercell of size $L_x = 5a$ in the propagation direction, such that, within a supercell, there are sufficient degrees of freedom for optimization purposes.

In Figure 3c, we show three different target dispersion curves for which we will optimize the waveguide. These are defined by a set of increasing Fourier components in k -space as

$$\bar{\omega}_1(k_x) = -0.01 \cos(k_x L_x) \quad (31)$$

$$\bar{\omega}_2(k_x) = -0.01 \cos(k_x L_x) + 0.004 \cos(2k_x L_x) \quad (32)$$

$$\bar{\omega}_3(k_x) = -0.01 \cos(k_x L_x) + 0.004 \cos(2k_x L_x) - 0.002 \cos(3k_x L_x) \quad (33)$$

where we define $\bar{\omega}$ as the dimensionless reduced frequency $\omega a / 2\pi c$. We take only cosine components, because, due to time-reversal invariance, the dispersion must have zero derivative at the high-symmetry Brillouin zone points. The three target curves are illustrated in Figure 3c, on top of the middle of the five guided bands of the waveguide, which is the one we optimize to match the targets. We allow for an arbitrary offset in frequency, and define the objective function as a mean-square error (MSE):

$$\mathcal{L} = \int_0^{L_x} dk_x |\bar{\omega}(k_x) - \langle \bar{\omega} \rangle - \bar{\omega}_i(k_x)|^2 \quad (34)$$

where $\bar{\omega}(k_x)$ is the reduced frequency of the waveguide band, and $\langle \bar{\omega} \rangle$ denotes $\bar{\omega}(k_x)$ averaged over k_x . As optimization parameters, we take the position and radii of the three rows of rods around the waveguide, as illustrated in Figure 3d. The shifts are applied symmetrically, with respect to the $y = 0$ plane, such that there are 45 free parameters total (Δx , Δy , and Δr for 15 rods on one side of the waveguide). The electric field intensity ($|E_z|^2$) for one of the guided modes of the starting waveguide is also shown in Figure 3d.

The results of the three separate optimizations for $\bar{\omega}_i(k_x)$ are shown in Figure 4. In Figure 4a, we show the dispersion optimized to match $\bar{\omega}_1(k_x)$, and in Figure 4b, we show the corresponding final structure and the electric field intensity of one of the guided modes. We use the Limited-memory

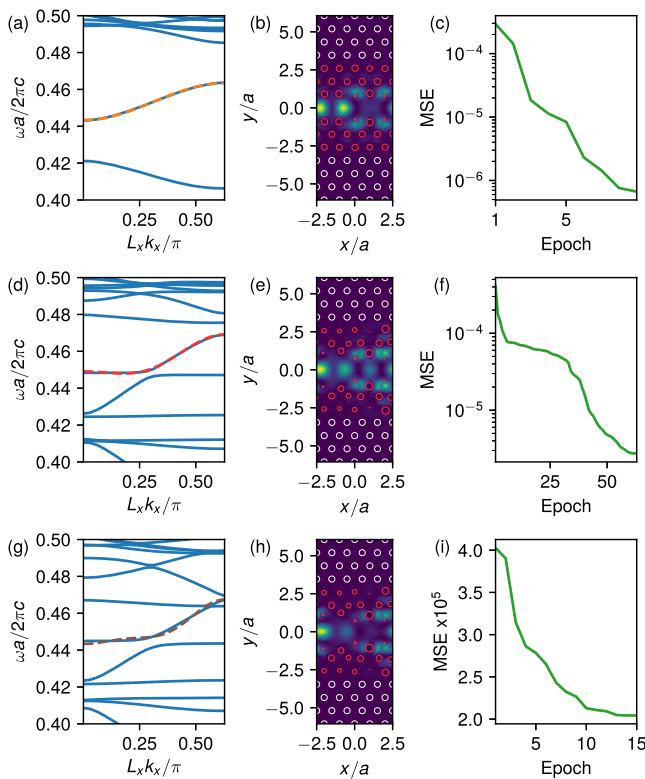


Figure 4. (a) Optimized waveguide band with a target band shape given by $\bar{\omega}_1(k_x)$ in Figure 3c (represented by a dashed line here). (b) The optimized structure and the electric field intensity of the waveguide mode at $L_x k_x/\pi = 0.5$. (c) Mean-square error vs optimization epoch. Panels (d)–(f) are the same as panels (a)–(c), for the target band shape given by $\bar{\omega}_2(k_x)$ in Figure 3(c). Panels (g)–(i) are the same as panels (a)–(c), for the target band shape given by $\bar{\omega}_3(k_x)$ in Figure 3c.

Broyden–Fletcher–Goldfarb–Shanno (LBFGS) algorithm^{54,55} to perform the optimization, and the evolution of the objective function with epochs (iterations) is shown in Figure 4c. We note that, as discussed previously, the “backward” propagation of the gradient, with respect to *all* 45 free parameters, requires approximately the same time as the “forward” computation of the photonic bands of the structure. The optimization converges quickly to the target dispersion, with a final MSE below 10^{-6} .

In Figures 4d–f, we show the same results, but for $\bar{\omega}_2(k_x)$. This more-complicated dispersion pattern requires a greater number of optimization iterations (Figure 4f), but the final result also matches the target function very well, as seen in Figure 4d. Finally, in Figures 4g–i, we show the same results for $\bar{\omega}_3(k_x)$. In the previous two optimizations, we used the unmodified structure of Figure 3d as the initial structure. However, as the target dispersion becomes harder to achieve, this yields a poor convergence in the optimization for $\bar{\omega}_3(k_x)$, with a final MSE of 8.9×10^{-5} . Thus, in Figures 4g–i, we show the result obtained using, as a starting configuration, the optimal structure obtained for $\bar{\omega}_2(k_x)$, i.e., the one shown in Figure 4e. This illustrates the fact that, as the optimization problem becomes more complicated, the results of a gradient-based optimization can be strongly dependent on the starting parameters. This can be remedied by gradually increasing the complexity of the optimization, as done here, or by running the

optimization with several randomly initialized starting structures.

Initializing the optimization with the structure of Figure 4e brings the MSE down to 2.0×10^{-5} (panel (i)). Still, as shown in Figure 4g, the waveguide band gets close to the target one, but the match is not perfect. This is to be expected, as the higher- k components correspond to smaller- and smaller-scale fluctuations of the real-space permittivity. This could be improved by allowing for rods of arbitrary shapes, or by increasing the supercell size L_x , therefore decreasing the maximum k_x in units of $1/a$.

GUIDED-MODE EXPANSION

The guided-mode expansion is a method for simulating three-dimensional (3D) layered periodic structures (PhC slabs) that, in many cases, has a computational complexity similar to the 2D plane-wave expansion, as it was formulated in ref 31. The method is approximate since the basis modes do not form a complete set, but it has already been shown to agree very well with first-principle methods for modes that are well-confined in the PhC region (weakly radiating in the claddings).^{31,43–45} The key idea is to use a smart basis that captures the z -dependence of such modes analytically. The periodic dielectric permittivity in the xy -plane then enters through its Fourier components, just like in the PWE described earlier. Below, we review the method in detail, and we discuss our AD implementation. As an original application, we then show an example optimization of the quality factor (Q) of a PhC slab cavity in a lithium niobate slab.

Method Description. Our implementation of the GME generalizes the seminal work of ref 31 to a multilayer structure,⁵⁶ as shown in Figure 5a. Along the z -direction, there are two semi-infinite layers: a lower cladding extending from z_0 to $-\infty$, and an upper cladding extending from z_{N+1} to $+\infty$. Between these two semi-infinite layers, we assume N layers with thickness d_j . The claddings and the layers are described by in-plane permittivity distribution $\epsilon_j(\rho)$, with $j = 0, \dots, N+1$ ($j = 0$ for the lower and $j = N+1$ for the upper cladding, and vector ρ denotes the in-plane coordinate). We allow for arbitrary $\epsilon_j(\rho)$, with the condition that the periodicity in all layers is the same, as shown by the blue shaded elementary cell in Figure 5a. The basis set that we use in the guided-mode expansion is then derived from a multilayer structure with homogeneous permittivity in each layer, as shown in Figure 5b. Specifically, each layer has the same thickness d_i as the starting structure of Figure 5a, and a constant relative permittivity given by

$$\bar{\epsilon}_j = \frac{1}{S} \int_S d\rho \epsilon_j(\rho) \quad j = 0, \dots, N+1 \quad (35)$$

where S is the in-plane elementary cell.

As shown in Figure 5c, the effective homogeneous structure supports a discrete set of guided bands, as well as a continuum of modes that are radiative in at least one of the claddings. Because of the reflection symmetry in the gz -plane, where \mathbf{g} is the in-plane wave-vector, the modes can again be classified as TE (electric field perpendicular to the gz -plane, odd reflection symmetry) and TM (magnetic field perpendicular to the gz -plane, even reflection symmetry). Notice that this classification is consistent with the PWE definition and the general photonic crystal literature,⁴¹ as TE modes have \mathbf{E} -field in the xy -plane and $E_z = 0$ everywhere, while TM modes have \mathbf{H} -field in the xy -plane and $H_z = 0$ everywhere.

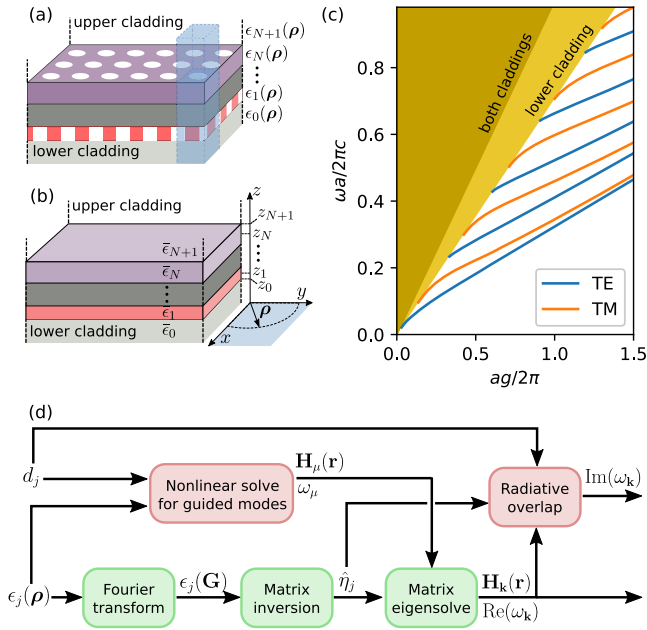


Figure 5. (a) A layered photonic crystal structure consisting of lower and upper semi-infinite claddings, and N layers of thickness d_1, \dots, d_N between. Each layer is described by relative permittivity $\epsilon_j(\rho)$, where ρ is the position in the xy -plane (cf. panel (b)). The permittivity is assumed to have the same in-plane periodicity in every layer, and the blue shaded square denotes the common elementary cell. (b) Effective homogeneous system corresponding to panel (a), where the permittivity $\bar{\epsilon}_j$ of each layer is the elementary-cell average of $\epsilon_j(\rho)$. (c) Frequency ω vs wave vector magnitude g of the photonic bands of a homogeneous structure as in panel (b), with two layers having a thickness of $d_1/a = 0.3$, $d_2/a = 0.5$ (a is a length unit), and permittivities of $\bar{\epsilon}_0 = 2$, $\bar{\epsilon}_1 = 10$, $\bar{\epsilon}_2 = 12$, $\bar{\epsilon}_3 = 1$. Discrete guided bands can be found below the lower-cladding light line $\omega/c = g/\sqrt{2}$. Above that line, there is a continuum of modes radiating either in the lower cladding only, or in both. (d) Computational graph of the guided-mode expansion method.

In the guided-mode expansion, we write an eigenmode of the PhC structure of Figure 5a as

$$\mathbf{H}_{\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{G}} \sum_{\alpha, p} c_{\mathbf{G}, \alpha, p} \mathbf{H}_{\mathbf{g}, \alpha, p}^{\text{guid}}(\mathbf{r}) \quad (36)$$

where we restrict the summation only to the fully guided modes that lie below the light lines of the claddings (Figure 5c). In eq 36, $\mathbf{g} = \mathbf{G} + \mathbf{k}$, the \mathbf{G} -summation is over reciprocal lattice vectors, just like in the plane-wave expansion case (see Figure 2c), α is a band index, and $p = \text{TE/TM}$ denotes polarization. Below, we label the guided mode by a single index μ that includes \mathbf{g} , α and p . Note that the two polarizations are generally mixed in the photonic crystal because of the ρ -dependent permittivity, and it is not possible to classify the PhC modes strictly as TE or TM, since all of the components of both \mathbf{E} and \mathbf{H} can be nonzero.

The expressions for the guided mode computation are derived in detail in the Supporting Information. This expansion basis also must be normalized, with respect to the scalar product for this problem, which is defined as

$$(\mathbf{H}_\nu, \mathbf{H}_\mu) = \frac{1}{S} \int_S d\rho \int_{-\infty}^{\infty} dz \mathbf{H}_\nu^\dagger(\rho, z), \mathbf{H}_\mu(\rho, z) \quad (37)$$

The normalization condition $(\mathbf{H}_\mu, \mathbf{H}_\mu) = 1$ for both polarizations is also given in the Supporting Information. The elements of the matrix for diagonalization, $\mathcal{H}_{\mu\nu} = (\mathbf{H}_\mu, \hat{\mathcal{O}}\mathbf{H}_\nu)$, can then be computed, and are also given in the Supporting Information for all the possible polarization combinations. Just like in the plane-wave expansion, the permittivity of each layer enters through the inverse matrix $\hat{\eta}_{j, \mathbf{G}, \mathbf{G}'}$, as defined in eq 23. The eigenvalues of $\hat{\mathcal{H}}$ give the frequencies $\omega_{\mathbf{k}}$ of the Bloch bands of the photonic crystal, while the eigenvectors correspond to the c_μ coefficients of eq 36, from which the magnetic field $\mathbf{H}_{\mathbf{k}}$ can be computed. At this stage, the frequencies $\omega_{\mathbf{k}}$ are purely real, because we have only included purely guided modes in the system, and $\hat{\mathcal{H}}$ is Hermitian. The final step of the method is to compute the coupling of the photonic crystal modes to the radiative modes of the homogeneous structure that lie above the light line (Figure 5c).

We use perturbation theory to compute the (small) imaginary components of the eigenmode frequencies of the PhC structure. These imaginary components correspond to a loss rate due to resonant coupling to radiating waves outgoing in the claddings. The decay rate of an eigenvalue $\omega_{\mathbf{k}}^2/c^2$ of matrix $\hat{\mathcal{H}}$ can be computed in first-order time-dependent perturbation theory^{31,57} as

$$-I\left(\frac{\omega_{\mathbf{k}}^2}{c^2}\right) = \pi \sum_{\mathbf{G}', p'} \sum_{o=l, u} |\mathcal{H}_{\mathbf{k}r}|^2 \rho_o(\mathbf{k} + \mathbf{G}', \omega_{\mathbf{k}}) \quad (38)$$

where ρ_o is a density of states of the radiative modes,⁶⁹ and $\mathcal{H}_{\mathbf{k}r}$ is a matrix element between the PhC mode and a radiative mode \mathbf{H}_r given by

$$\mathcal{H}_{\mathbf{k}r} = (\mathbf{H}_{\mathbf{k}}, \hat{\mathcal{O}}\mathbf{H}_r) = \sum_{\mathbf{G}, \alpha, p} c_\mu^* \times (\mathbf{H}_{\mu}, \hat{\mathcal{O}}\mathbf{H}_r) \quad (39)$$

For radiative modes, the index r includes \mathbf{G} , p , and o , where o denotes if the mode is outgoing in the lower cladding or the upper cladding, respectively. We note again that, for guided modes, the index μ includes \mathbf{G} , α , and p . Further details on the radiative rate computation are given in the Supporting Information. Once the imaginary part $I(\omega^2/c^2)$ is computed, the imaginary part of ω can be determined through $I(\omega/c) = I((\mathcal{R}(\omega^2/c^2) + iI(\omega^2/c^2))^{1/2})$, and the quality factor associated with a Bloch mode $\mathbf{H}_{\mathbf{k}}$ is $Q = \mathcal{R}(\omega_{\mathbf{k}})/(2I(\omega_{\mathbf{k}}))$.

Automatic Differentiation. A high-level computational graph for the GME is presented in Figure 5d. In our discussion thus far, we have already covered everything that is needed for the AD of the GME computation. The three green blocks in Figure 5d are the same as those for the PWE of Figure 2d. The only difference here is that the Fourier transform and matrix inversion are done for every layer $j = 0, \dots, N+1$. The AD step through the nonlinear solve for the guided modes is done as defined in eq 7 and is the only nontrivial operation that we added to Autograd to enable the GME.

The guided-mode frequency ω_μ is the solution to the implicit function

$$D_{22}(\omega, \epsilon, \mathbf{d}) = 0 \quad (40)$$

(defined in the Supporting Information) where ϵ is a vector containing all average permittivity values $\bar{\epsilon}_j$, and \mathbf{d} is a vector containing all layer thickness values d_j . This implicit function is

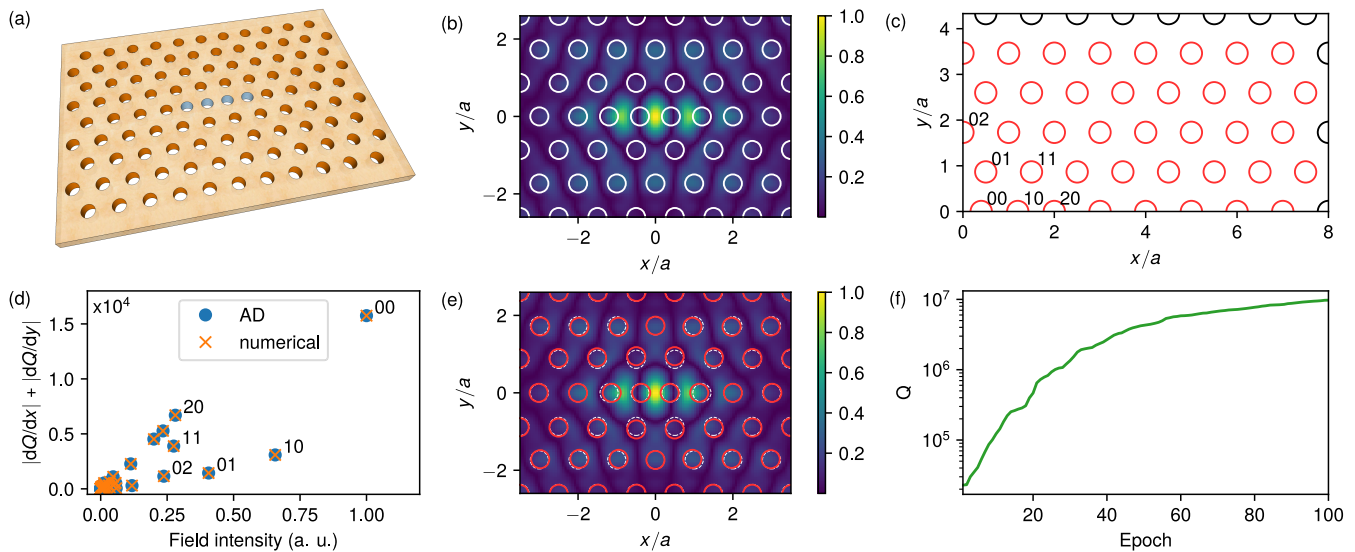


Figure 6. (a): Schematic of a $L4/3$ photonic crystal cavity in a lithium niobate slab in air. The cavity is defined by four holes (marked in blue) introduced in place of the three holes of the regular lattice. (b): Electric field $|E_y|^2$ of the fundamental mode of the starting cavity. (c): One quadrant of the supercell used in the simulation. The positions of the red holes are the free parameters used in the optimization. The holes closest to the center of the cavity are labeled by their x, y position index. (d): Magnitude of the gradient of the quality factor with respect to hole position vs integrated field intensity on the surface of the corresponding hole. The labels refer to the corresponding holes in panel (c). (e): Optimized high- Q cavity. The position of the holes of the starting structure are shown in white dashed lines. (f): Quality factor vs optimization epoch.

actually a simple version of eq 7, in that D_{22} and ω are scalars and no matrix inversion is needed. Thus, once a solution ω_μ is found, the reverse-mode AD step, with respect to ϵ , is defined as

$$\frac{\partial \omega_\mu}{\partial \epsilon} = - \left[\frac{\partial D_{22}}{\partial \omega_\mu} \right]^{-1} \frac{\partial D_{22}}{\partial \epsilon} \quad (41)$$

and similarly for \mathbf{d} . To implement this, we use Autograd to define a differentiable computation of D_{22} , which is computed as a product of matrices, as defined in eq (S15) in the Supporting Information. We then use Autograd to automatically compute $\partial D_{22} / \partial \omega_\mu$, $\partial D_{22} / \partial \epsilon$, and $\partial D_{22} / \partial \mathbf{d}$, using these quantities to perform the computation defined in eq 41 and continue the gradient accumulation through the main part of the computation.

The computation of the matrix for diagonalization from the guided modes and the Fourier components of the permittivity involves a large number of algebraic operations, which we outlined in the Supporting Information. However, these are conceptually simple to differentiate and are all supported by Autograd. The same is true for the radiative overlap elements, which is the final step of the forward computation shown in Figure 5d. However, this once again highlights the importance of using AD in handling complex computational algorithms: apart from the fundamental speed advantage offered by RM differentiation, we also profit immensely from the fact that the differentiation is split into elementary building blocks, and the extremely long and complicated derivative accumulation is done automatically. In the following section, we finally apply this to a practical problem: optimizing a photonic crystal cavity.

Cavity Optimization. Photonic crystal cavities are attractive because they can confine light to volumes close to the diffraction limit.⁵⁸ Furthermore, through small modifications of the structure, quality factors exceeding 10 million have been experimentally demonstrated in silicon PhC slabs

suspended in air.⁵⁹ Recently, lithium niobate (LN) has emerged as an extremely versatile material for integrated photonics applications. However, LN has a much lower refractive index than silicon ($n = 2.21$ vs $n = 3.48$, respectively) at wavelengths of $\lambda \approx 1.55 \mu\text{m}$. The significantly lower index contrast, with respect to the air holes and claddings, makes the design of high- Q PhC cavities in LN much more challenging. Recently, a design with a theoretical $Q = 1.5 \times 10^6$ was proposed,⁶⁰ with an associated mode volume is $V = 2.43(\lambda/n)^3$, which is defined as

$$V = \frac{\int d\mathbf{r} |\mathbf{E}_m(\mathbf{r})|^2 \epsilon(\mathbf{r})}{\max_{\mathbf{r}} [|\mathbf{E}_m(\mathbf{r})|^2 \epsilon(\mathbf{r})]} \quad (42)$$

Here, we demonstrate the optimization of a LN photonic crystal cavity with a simultaneously higher Q and lower V than the design described in ref 60.

The starting cavity is shown in Figure 6a and is based on a triangular lattice of circular holes in an LN slab. We use the $L4/3$ design,⁵⁸ in which four holes (marked in blue) are introduced in place of the three holes of the regular lattice. This type of defect results in a tightly confined cavity mode, as shown in Figure 6b. For the underlying PhC, we use the same parameters as in ref 60: lattice constant $a = 620 \text{ nm}$, slab thickness $d = 270 \text{ nm}$, and hole radius $r = 145 \text{ nm}$. The starting cavity of Figure 6b has $Q = 6100$, and a mode volume $V = 0.61(\lambda/n)^3$. In Figure 6c, we show one-quarter of the supercell used in the simulation (the positive- x and positive- y quadrant). As optimization parameters, we use the positions of the holes marked in red, and all holes are moved symmetrically, with respect to the x - and y -axes. Note that this also means that the holes that lie on the x -axis are not shifted in the y -direction, and vice versa for the holes on the y -axis. Thus, there are a total of 70 free parameters $\Delta \mathbf{x}$, $\Delta \mathbf{y}$ to optimize, and the objective function to be maximized is simply the quality factor of the resonator, $\mathcal{L}(\Delta \mathbf{x}, \Delta \mathbf{y}) = Q$.

Before optimizing, we inspect the gradient of the quality factor, with respect to each hole for the starting structure. This is done both as a check that the gradients computed using automatic differentiation are correct, and to investigate the sensitivity of the Q to the position of each of the holes. In Figure 6d, we show the combined magnitude of the gradient $|dQ/dx| + |dQ/dy|$ for every hole, plotted versus the field intensity at that hole,

$$I_h = \oint_h d\rho |\mathbf{E}(\rho, z_1)|^2 \quad (43)$$

where the integration is along the contour of the hole, and the field $\mathbf{E}(\rho, z_1)$ is taken at the center of the slab. As expected, there is a clear correlation between the sensitivity of the Q -factor to the displacement of a hole and the proximity of that hole to the center of the cavity, where the field intensity of the eigenmode is the highest. In the same panel, we also compare the AD gradient with values computed numerically using the finite-difference method, i.e., by slightly perturbing each parameter one by one. We observe that the values match perfectly. The difference is that 70 extra simulations were required to get the numerical values, while the RM AD computation after the initial Q is computed required approximately the same time as the FM simulation.

In Figure 6e, we show the final optimized cavity, and in Figure 6f, we show the evolution of the quality factor with optimization steps. The optimal values of all 45 parameters ($\Delta x, \Delta y$) are given in the Supporting Information. The LBFGS optimization is close to converged after 100 epochs, and it produces a cavity mode with a GME-computed quality factor of $Q = 9.7 \times 10^6$ and a mode volume of $V = 0.49(\lambda/n)^3$. The mode volume is smaller than that of the starting structure, because we restricted the two holes closest to the center (labeled "00" and "10" in Figure 6c) to only shift inward toward the cavity center. This can be seen in Figure 6e, where we also show the positions of the holes for the unoptimized cavity. As is common in PhC cavity optimization and can be seen there, very small changes to the structure can result in drastic changes in the quality factor. We also computed the Q of the final cavity design using a first-principles finite-difference time-domain simulation in Lumerical FDTD Solutions, and found $Q = 2.4 \times 10^6$. This is slightly lower than the GME-computed result, and the discrepancy can be attributed to the fact the GME is an approximate method. In any case, this Q value is still higher than that of ref 60, and it confirms the success of the GME optimization.

DISCUSSION AND CONCLUSION

The guided-mode expansion is an invaluable tool in the study of photonic crystal slabs. Although approximate, the method has been shown to agree well in comparison to first-principle simulations in several different structures, while being computationally faster.^{31,43–45} Here, we used the method—and our differentiable implementation—to improve the quality factor of a small-volume photonic crystal cavity by more than 2 orders of magnitude, compared to the starting structure. In the past, there have been various approaches to PhC cavity optimization. Heuristic optimizations of the quality factor have been extremely successful in silicon slabs,⁵⁹ but they become challenging in lower-index materials, or when other features, such as the mode volume, must also be taken into account. Traditional gradient-based inverse design techniques have been tried on the problem of designing high- Q small-volume

cavities, both using topology optimization^{61,62} and by optimizing the hole positions of a PhC resonator.⁶³ However, in both cases, the result was only moderately successful, especially with respect to the quality factor. This is fundamentally linked to the use of finite-difference frequency-domain simulations, which have significant difficulties in resolving sharp resonances that shift in frequency during the course of optimization. In contrast, global optimization methods using the GME have already proven extremely useful when applied to PhC cavities,^{43,58,64,65} both, because of the speed of the individual computation and because high- Q resonances actually represent the condition under which the approximation of the method works best. Here, we have moved one step further, leveraging our differentiable implementation of GME to perform a gradient-based optimization. This leads to a significantly lower number of simulations needed to obtain an optimal structure. For comparison, a recent work⁶⁵ reports on the application of GME to a similar cavity design by using a global optimization procedure on a comparable number of parameters, which converged to a result that is comparable with ours (18 times larger Q but 3.5 times larger V , in a structure with a slightly larger refractive index contrast in ref 65). Crucially, the global optimization required 800 000 evaluations, while our gradient-based optimization converges after 131 function and gradient evaluations (equivalent to 262 forward-only evaluations in terms of computational time), which highlights the computational advantage of AD for this problem.

In periodic structures, the guided-mode expansion is particularly useful for the study of quasi-guided modes above the light line, which are difficult to isolate in first-principles finite-difference methods (time domain or frequency domain). Therefore, we anticipate that one direction in which our software package will prove invaluable is the study of bound states in the continuum,⁶⁶ which can find applications for on-chip zero-index metamaterials,⁴⁵ or for enhanced nonlinear frequency conversion.⁶⁴ Another potential application is the optimization of the dispersion of exciton-polaritons in nanostructures.⁶⁷ We can also envision using GME to optimize the dispersion of a PhC slab waveguide—similarly to what we demonstrated here with plane-wave expansion, but for a fully 3D structure. This has recently been done using inverse design applied to the eigenmodes of the real-space finite-difference matrix defining Maxwell's equations.⁶⁸ However, this has a significant computational cost, and it was furthermore only limited to modes that lie below the light line. In contrast, the GME method is extremely fast. For example, in our GME optimization, we used a computational space of size 16×10 elementary cells, and the converged simulation requires a time period on the order of 10 min on a personal computer. A waveguide simulation would have a smaller supercell and, thus, be even faster. In addition, the optimization is not restricted to modes below the light line, and the radiation losses can be included in the objective function, as was done for example in ref 44.

In conclusion, we have extended the paradigm of gradient-based inverse design to two mode expansion methods that are widely used in the study of photonic crystals. Namely, we have implemented the 2D plane-wave expansion and the guided-mode expansion methods in a way that allows for efficient gradient computation using reverse-mode (RM) automatic differentiation (AD). Furthermore, this naturally brings great flexibility in defining structure parametrizations and objective

functions. We have also made our software publicly available³² for future use by the community. More broadly, our paper demonstrates the power of AD for generic physical simulations, and it highlights the fact that the sophisticated AD libraries that have been developed in the past decade^{20–24} hold great promise for future large-scale inverse design of physical structures.

■ ASSOCIATED CONTENT

SI Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acsphotonics.0c00327>.

Details of the mathematical derivations of some of the equations presented in this work (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Momchil Minkov – Department of Electrical Engineering, and Ginzton Laboratory, Stanford University, Stanford, California 94305, United States; orcid.org/0000-0003-0665-8412; Email: mminkov@stanford.edu

Shanhui Fan – Department of Electrical Engineering, and Ginzton Laboratory, Stanford University, Stanford, California 94305, United States; Email: shanhui@stanford.edu

Authors

Ian A. D. Williamson – Department of Electrical Engineering, and Ginzton Laboratory, Stanford University, Stanford, California 94305, United States; orcid.org/0000-0002-6699-1973

Lucio C. Andreani – Dipartimento di Fisica, Università di Pavia, 27100 Pavia, Italy

Dario Gerace – Dipartimento di Fisica, Università di Pavia, 27100 Pavia, Italy

Beicheng Lou – Department of Applied Physics, and Ginzton Laboratory, Stanford University, Stanford, California 94305, United States

Alex Y. Song – Department of Electrical Engineering, and Ginzton Laboratory, Stanford University, Stanford, California 94305, United States

Tyler W. Hughes – Department of Applied Physics, and Ginzton Laboratory, Stanford University, Stanford, California 94305, United States; orcid.org/0000-0001-7989-0891

Complete contact information is available at:

<https://pubs.acs.org/doi/10.1021/acsphotonics.0c00327>

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by a MURI grant from the U.S. Air Force Office of Scientific Research (No. FA9550-17-1-0002), by the Department of Defense Joint Directed Energy Transition Office (DE-JTO) (under Grant No. N00014-17-1-2557), and by the Gordon and Betty Moore Foundation (No. GBMF4744).

■ REFERENCES

(1) Molesky, S.; Lin, Z.; Piggott, A. Y.; Jin, W.; Vuckovic, J.; Rodriguez, A. W. Outlook for inverse design in nanophotonics. *Nat. Photonics* **2018**, *12*, 659–670.

(2) Wang, F.; Jensen, J. S.; Mørk, J.; Sigmund, O. Systematic design of loss-engineered slow-light waveguides. *J. Opt. Soc. Am. A* **2012**, *29*, 2657–66.

(3) Piggott, A. Y.; Lu, J.; Lagoudakis, K. G.; Petykiewicz, J.; Babinec, T. M.; Vucković, J. Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer. *Nat. Photonics* **2015**, *9*, 374–377.

(4) Frelsen, L. F.; Ding, Y.; Sigmund, O.; Frandsen, L. H. Topology optimized mode multiplexing in silicon-on-insulator photonic wire waveguides. *Opt. Express* **2016**, *24*, 16866.

(5) Lin, Z.; Christakis, L.; Li, Y.; Mazur, E.; Rodriguez, A. W.; Lončar, M. Topology-optimized dual-polarization Dirac cones. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2018**, *97*, No. 081408(R).

(6) Lin, Z.; Pick, A.; Lončar, M.; Rodriguez, A. W. Enhanced Spontaneous Emission at Third-Order Dirac Exceptional Points in Inverse-Designed Photonic Crystals. *Phys. Rev. Lett.* **2016**, *117*, 107402.

(7) Lin, Z.; Liang, X.; Lončar, M.; Johnson, S. G.; Rodriguez, A. W. Cavity-enhanced second-harmonic generation via nonlinear-overlap optimization. *Optica* **2016**, *3*, 233.

(8) Hughes, T. W.; Minkov, M.; Williamson, I. A. D.; Fan, S. Adjoint Method and Inverse Design for Nonlinear Nanophotonic Devices. *ACS Photonics* **2018**, *5*, 4781–4787.

(9) Veronis, G.; Dutton, R. W.; Fan, S. Method for sensitivity analysis of photonic crystal devices. *Opt. Lett.* **2004**, *29*, 2288.

(10) Jensen, J. S.; Sigmund, O. Topology optimization for nanophotonics. *Laser Photon. Rev.* **2011**, *5*, 308–321.

(11) Lalau-Keraly, C. M.; Bhargava, S.; Miller, O. D.; Yablonovitch, E. Adjoint shape optimization applied to electromagnetic design. *Opt. Express* **2013**, *21*, 21693.

(12) Dobson, D. C.; Cox, S. J. Maximizing band gaps in two-dimensional photonic crystals. *SIAM J. Appl. Math.* **1999**, *59*, 2108–2120.

(13) Kao, C. Y.; Osher, S.; Yablonovitch, E. Maximizing band gaps in two-dimensional photonic crystals by using level set methods. *Appl. Phys. B: Lasers Opt.* **2005**, *81*, 235–244.

(14) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. In *Parallel Distributed Processing*, Vol. 1; Rumelhart, D. E., McClelland, R. J., Eds.; MIT Press, 1986; Chapter 8.

(15) Bischof, C.; Carle, A.; Corliss, G.; Griewank, A.; Hovland, P. ADIFOR-Generating Derivative Codes from Fortran Programs. *Sci. Program.* **1992**, *1*, 11–29.

(16) Sambridge, M.; Rickwood, P.; Rawlinson, N.; Sommacal, S. Automatic Differentiation in Geophysical Inverse Problems. *Geophys. J. Int.* **2007**, *170*, 1–8.

(17) Enciu, P.; Gerbaud, L.; Wurtz, F. Automatic Differentiation Applied for Optimization of Dynamical Systems. *IEEE Trans. Magn.* **2010**, *46*, 2943–2946.

(18) Baydin, A. G.; Pearlmutter, B. A.; Radul, A. A.; Siskind, J. M. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.* **2018**, *18*, 1–43.

(19) Rackauckas, C.; Ma, Y.; Dixit, V.; Guo, X.; Innes, M.; Revels, J.; Nyberg, J.; Ivaturi, V. A Comparison of Automatic Differentiation and Continuous Sensitivity Analysis for Derivatives of Differential Equation Solutions. *arXiv*, **2018**, arXiv:1812.01892 [cs].

(20) Autograd: Efficiently computes derivatives of numpy code. Available via the Internet at: <https://github.com/HIPS/autograd> (accessed June 7, 2020).

(21) Paszke, A. et al. *Advances in Neural Information Processing Systems*, Vol. 32; 2019; pp 8024–8035.

(22) Agrawal, A.; Modi, A. N.; Passos, A.; Lavoie, A.; Agarwal, A.; Shankar, A.; Ganichev, I.; Levenberg, J.; Hong, M.; Monga, R.; Cai, S. TensorFlow Eager: A Multi-Stage, Python-Embedded DSL for Machine Learning. *arXiv*, **2019**, arXiv:1903.01855 [cs].

(23) Innes, M. Don't Unroll Adjoint: Differentiating SSA-Form Programs. *arXiv*, **2019**, arXiv:1810.07951 [cs].

(24) Bradbury, J.; Frostig, R.; Hawkins, P.; Johnson, M. J.; Leary, C.; Maclaurin, D.; Wanderman-Milne, S. *JAX: composable transformations*

of Python+NumPy programs, 2018. Available via the Internet at: <http://github.com/google/jax>, (accessed June 7, 2020).

(25) Richardson, A. Seismic Full-Waveform Inversion Using Deep Learning Tools and Techniques. *arXiv*, **2018**, arXiv:1801.07232 [physics].

(26) Hoyer, S.; Sohl-Dickstein, J.; Greydanus, S. Neural Reparameterization Improves Structural Optimization. *arXiv*, **2019**, arXiv:1909.04240 [cs, stat].

(27) Rackauckas, C.; Ma, Y.; Martensen, J.; Warner, C.; Zubov, K.; Supekar, R.; Skinner, D.; Ramadhan, A. Universal Differential Equations for Scientific Machine Learning. *arXiv*, **2020**, arXiv:2001.04385 [cs, math, q-bio, stat].

(28) Hughes, T. W.; Williamson, I. A. D.; Minkov, M.; Fan, S. Forward-Mode Differentiation of Maxwell's Equations. *ACS Photonics* **2019**, *6*, 3010–3016.

(29) Hughes, T. W.; Williamson, I. A. D.; Minkov, M.; Fan, S. Wave physics as an analog recurrent neural network. *Sci. Adv.* **2019**, *5*, No. eaay6946.

(30) Su, L.; Vercruysse, D.; Skarda, J.; Sapra, N. V.; Petykiewicz, J. A.; Vuckovic, J. Nanophotonic inverse design with SPINS: Software architecture and practical considerations. *Appl. Phys. Rev.* **2020**, *7*, 011407.

(31) Andreani, L. C.; Gerace, D. Photonic-crystal slabs with a triangular lattice of triangular holes investigated using a guided-mode expansion method. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2006**, *73*, 235114.

(32) Legume: Guided Mode Expansion supporting automatic differentiation with autograd. Available via the Internet at: <https://github.com/fancompute/legume> (accessed June 7, 2020).

(33) Boyd, R. W. *Nonlinear Optics*; Elsevier, 2003.

(34) Baba, T. Slow light in photonic crystals. *Nat. Photonics* **2008**, *2*, 465–473.

(35) Arizmendi, L. Photonic applications of lithium niobate crystals. *Phys. status solidi* **2004**, *201*, 253–283.

(36) Wengert, R. E. A. Simple Automatic Derivative Evaluation Program. *Commun. ACM* **1964**, *7*, 463–464.

(37) Griewank, A.; Walther, A. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, Vol. 105; SIAM, 2008, DOI: 10.1137/1.9780898717761.

(38) Dauvergne, B.; Hascoët, L. The data-flow equations of checkpointing in reverse automatic differentiation. *Int. Conf. Comput. Sci.* **2006**, 3994, 566–573.

(39) van der Walt, S.; Colbert, S. C.; Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Comput. Sci. Eng.* **2011**, *13*, 22–30.

(40) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272.

(41) Joannopoulos, J. D.; Johnson, S. G.; Winn, J. N.; Others *Photonic Crystals: Molding the Flow of Light*; Princeton University Press, 2008.

(42) Lalanne, P.; Yan, W.; Vynck, K.; Sauvan, C.; Hugonin, J. P. Light Interaction with Photonic and Plasmonic Resonances. *Laser Photonics Rev.* **2018**, *12*, 1700113.

(43) Minkov, M.; Savona, V. Automated optimization of photonic crystal slab cavities. *Sci. Rep.* **2015**, *4*, 5124.

(44) Minkov, M.; Savona, V. Wide-band slow light in compact photonic crystal coupled-cavity waveguides. *Optica* **2015**, *2*, 631–634.

(45) Minkov, M.; Williamson, I. A. D.; Xiao, M.; Fan, S. Zero-index bound states in the continuum. *Phys. Rev. Lett.* **2018**, *121*, 263901.

(46) Ho, K. M.; Chan, C. T.; Soukoulis, C. M. Existence of a photonic gap in periodic dielectric structures. *Phys. Rev. Lett.* **1990**, *65*, 3152.

(47) Li, L. Use of Fourier series in the analysis of discontinuous periodic structures. *J. Opt. Soc. Am. A* **1996**, *13*, 1870–1876.

(48) Giles, M. B. Collected matrix derivative results for forward and reverse mode algorithmic differentiation. *Lect. Notes Comput. Sci. Eng.* **2008**, *64*, 35–44.

(49) Lee, T. H. Adjoint method for design sensitivity analysis of multiple eigenvalues and associated eigenvectors. *AIAA J.* **2007**, *45*, 1998–2004.

(50) Johnson, S.; Joannopoulos, J. Block-iterative frequency-domain methods for Maxwell's equations in a planewave basis. *Opt. Express* **2001**, *8*, 173.

(51) Nelson, R. B. Simplified calculation of eigenvector derivatives. *AIAA J.* **1976**, *14*, 1201–1205.

(52) Lee, S.-W.; Mittra, W. L. Fourier Transform of a Polygonal Shape Function and Its Application in Electromagnetics. *IRE Trans. Antennas Propag.* **1983**, *31*, 99–103.

(53) Moss, D. J.; Morandotti, R.; Gaeta, A. L.; Lipson, M. New CMOS-compatible platforms based on silicon nitride and Hydex for nonlinear optics. *Nat. Photonics* **2013**, *7*, 597.

(54) Byrd, R. H.; Lu, P.; Nocedal, J.; Zhu, C. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing* **1995**, *16*, 1190–1208.

(55) Zhu, C.; Byrd, R. H.; Lu, P.; Nocedal, J. Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization. *ACM Trans. Math. Softw.* **1997**, *23*, 550–560.

(56) Zabelin, V. *Numerical Investigations of Two-Dimensional Photonic Crystal Optical Properties, Design and Analysis of Photonic Crystal Based Structures*. Ph.D. Thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2009 (DOI: 10.5075/epfl-thesis-4315).

(57) Ochiai, T.; Sakoda, K. Nearly free-photon approximation for two-dimensional photonic crystal slabs. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2001**, *64*, 45108.

(58) Minkov, M.; Savona, V.; Gerace, D. Photonic crystal slab cavity simultaneously optimized for ultra-high Q/v and vertical radiation coupling. *Appl. Phys. Lett.* **2017**, *111*, 131104.

(59) Asano, T.; Ochi, Y.; Takahashi, Y.; Kishimoto, K.; Noda, S. Photonic crystal nanocavity with a Q factor exceeding eleven million. *Opt. Express* **2017**, *25*, 1769–1777.

(60) Li, M.; Liang, H.; Luo, R.; He, Y.; Lin, Q. High-Q 2D Lithium Niobate Photonic Crystal Slab Nanoresonators. *Laser Photonics Rev.* **2019**, *13*, 1800228.

(61) Lu, J.; Boyd, S.; Vučković, J. Inverse design of a three-dimensional nanophotonic resonator. *Opt. Express* **2011**, *19*, 10563–10570.

(62) Liang, X.; Johnson, S. G. Formulation for scalable optimization of microcavities via the frequency-averaged local density of states. *Opt. Express* **2013**, *21*, 30812.

(63) Wang, F.; Zheng, D.; Decker, J.; Wu, X.; Essertel, G. M.; Rompf, T. Demystifying differentiable programming: shift/reset the penultimate backpropagator. *Proc. ACM Program. Lang.* **2019**, *3*, 1–31.

(64) Minkov, M.; Gerace, D.; Fan, S. Doubly resonant $\chi^{(2)}$ nonlinear photonic crystal cavity based on a bound state in the continuum. *Optica* **2019**, *6*, 1039–1045.

(65) Vasco, J. P.; Savona, V. Global optimization of an encapsulated Si/SiO₂ L3 cavity for ultra-high quality factor. *arXiv*, **2019**, arXiv:1911.03361.

(66) Hsu, C. W.; Zhen, B.; Stone, A. D.; Joannopoulos, J. D.; Soljacic, M. Bound states in the continuum. *Nat. Rev. Mater.* **2016**, *1*, 16048.

(67) Dang, N. H. M.; Gerace, D.; Drouard, E.; Trippé-Allard, G.; Lédée, F.; Mazurczyk, R.; Deleporte, E.; Seassal, C.; Nguyen, H. S. Tailoring Dispersion of Room-Temperature Exciton-Polaritons with Perovskite-Based Subwavelength Metasurfaces. *Nano Lett.* **2020**, *20*, 2113–2119.

(68) Vercruysse, D.; Sapra, N. V.; Su, L.; Trivedi, R.; Vučković, J. Analytical level set fabrication constraints for inverse design. *Sci. Rep.* **2019**, *9*, 8999.

(69) Carniglia, C. K.; Mandel, L. Quantization of Evanescent Electromagnetic Waves. *Phys. Rev. D: Part. Fields* **1971**, *3*, 280–296.