Time reversal differentiation of FDTD for photonic inverse design

Rui Jie Tang[1,†,*], Soon Wei Daniel Lim[2,†,*], Marcus Ossiander[2,3], Xinghui Yin[4], Federico Capasso[2]

[1]University of Toronto, 27 King's College Circle, Toronto, Ontario M5S 1A1, Canada

[2]Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA

[3]Institute of Experimental Physics, Graz University of Technology, 8010 Graz, Austria

[4]LIGO – Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[†]Equal contribution

*Email: ruijie.tang@mail.utoronto.ca, lim982@g.harvard.edu

Abstract

Differentiable models enable the efficient computation of parameter gradients for continuous functions, greatly expediting the optimization of high-dimensional systems. This makes them an asset for the design of nanostructured metasurfaces. The adjoint variable method (AVM) is the workhorse for photonic gradient computation but can be challenging to implement with Finite Difference Time Domain (FDTD) electromagnetic for certain optimization problems. Automatically differentiable (AD) platforms remove the need for manual constructions while retaining favorable computational scaling but high memory consumption limits their application to small systems. Here we introduce a method of gradient calculation based on the direct differentiation of the FDTD update equations by leveraging the time-reversible nature of Maxwell's equations. We support open and closed systems by recording the time-dependent fields at lossy boundaries and playing them back during the time-reversed FDTD simulation. The method is generally applicable without the high memory consumption of AD by eliminating redundant memory operations performed at each timestep. We demonstrate this architecture in a three-dimensional FDTD simulation. Its computational cost is comparable to the adjoint method and it reduces memory requirements by 98% compared to an equivalent AD calculation for calculating a 900-element gradient vector. The differentiable simulator is applied to design two systems: a color sorter with a frequency-domain behavior and a resonant nanostructure array with a time-domain behavior. This approach to differentiate grid-based simulators is applicable to a broad range of physics simulators, thereby broadening the scope of inverse design topology optimization across fields.

Keywords: automatic differentiation, metasurface inverse design, topology optimization, electromagnetic simulation, adjoint variable method

Introduction

Nanostructured surfaces have been utilized to sense and manipulate numerous degrees of freedom of light. Metalenses, for instance, control the transmitted wavefront through an intricately structured surface to achieve diffraction-limited and aberration-corrected focusing (1,2). Modern lithography enables the creation of features far smaller than the wavelength of visible light, resulting in a vast space of possible nanostructures. Identifying high-performing devices that can be fabricated within equipment constraints

from this massive set of possible structures is essential. The two main approaches to address this challenge are forward design and inverse design.

Forward design involves constructing a large device from simpler components that have well-defined behavior. In flat optics, these components are discrete subwavelength nanostructures called meta-atoms. Each meta-atom in a "library" (*i.e.*, a collection) consists of elementary shapes and has its optical behavior (*e.g.*, transmitted phase delay) individually modeled with electromagnetic simulation techniques such as Finite Difference Time Domain (FDTD) (3) and finite element models (FEM). In a second step, these meta-atoms are assembled to a macroscopic optic based on a precalculated spatial distribution. For example, when designing a lens, meta-atoms would be arranged to induce a hyperbolic phase profile. Forward design can systematically generate large nanophotonic devices for applications with requirements that can be met by a library of meta-atoms with a limited number of degrees of freedom.

However, inverse design is required for applications without a well-defined desired spatial parameter distribution or those demanding wider-ranging behaviors. One example is designing different behaviors over a band of wavelengths, which requires control over the chromatic dispersion of the nanostructures and coverage of a much larger structure parameter space (2,4,5). Inverse design techniques such as topology optimization treat this as an optimization problem over a high-dimensional parameter space, *i.e.*, they maximize or minimize an objective function which describes the desired behavior. There are several techniques available for photonic inverse design (6,7); most high-dimensional optimization problems employ gradient descent directly onto the design parameters or train neural network generators (8,9) to discover high performing solutions. Beyond simple gradient descent, the information in these gradients can be incorporated with deep learning-based algorithms to attain globally-optimal devices (10–12). In gradient-based approaches, the most computationally intensive step is the calculation of the gradient of the objective function itself. Finite difference approximations can achieve this, but their computational complexity scales poorly with respect to the number of degrees of freedom (at least one objective function call per degree of freedom). A physics-informed neural network for approximating forward simulation and gradients can provide excellent computational scaling but may trade-off accuracy and generalizability (13).

The adjoint variable method (AVM), also known as sensitivity analysis, is the most widely used and computationally efficient technique to optimize systems with many degrees of freedom: AVM calculates all gradients simultaneously, at the computational cost of just two objective function calls (*i.e.*, two electromagnetic simulations in the photonics case). For the commonly-encountered case in photonics where the objective function is written in terms of frequency-domain complex-valued fields and the tunable parameters are dielectric permittivity distributions, AVM takes a simple form that can be easily implemented in existing FDTD suites (14–16). However, in situations that differ from the above case, such as systems with time-domain objective functions, time-varying materials, or tunable parameters that are associated with multiple pixels (*e.g.*, characteristic dimensions (17,18) or the rotation angle of a structure), the AVM requires the derivation of complicated operators that often have no closed form.

Automatic differentiation (AD) is another efficient technique for gradient calculation and thereby topology optimization (18–22). AD-based electromagnetic simulators based on rigorous coupled wave analysis (18,20) and finite difference frequency domain (FDFD) (23,24) are in the public domain. In its most common form, reverse mode (RM), AD maintains a record of all mathematical operations undertaken during a calculation and computes the exact numerical gradients using the chain rule. Therefore, the user does not need to perform analytical derivatives (21). However, applying AD to iterative grid-based simulation methods (*e.g.*, FDTD) is limited to small problems or large computer clusters due to the memory requirements of saving all intermediate values at every iteration timestep and pixel position, along with every mathematical operation involved.

In this paper, we introduce an alternative method to AVM and AD for obtaining arbitrary objective function gradients in FDTD: Direct Differentiation (DD) of the FDTD update steps. That is, we analytically differentiate the FDTD update equations and propagate the objective function gradients in a manner akin to an FDTD simulation running in reverse. This technique maintains the efficient computational complexity scaling of AVM and AD where the gradient calculation time is independent of the number of tunable parameters, has orders of magnitude less memory complexity compared to AD, and provides a systematic approach to accumulating parameter gradients. Our method is generalizable to many time- and frequency domain objective functions and optimization parameters. Our method provides – without changes – optimization gradients for objective functions including time-domain field components and broadband

responses. In these cases, our method is advantageous to the AVM, as it does not require the (mathematically challenging and sometimes impossible) derivation of closed-form adjoint formulations. For frequency-domain objective functions defined in a narrow frequency band, the AVM reduces to its computationally efficient form (see section AVM for frequency-domain objectives), allowing a more straightforward implementation into existing simulation tools.

Instead of storing intermediate parameter (field) values at each timestep, we perform backpropagation by running the FDTD simulation backwards, starting with the final system state and time-stepping in reverse towards the initial conditions. Since the backpropagation chain rule calculation proceeds in this same reverse direction, the intermediate parameter values are calculated precisely when they are needed at each timestep. This reverse simulation approach can also be separately applied in AVM adjoint simulations to remove the need for checkpointing (25,26). We then apply our architecture to design two devices: one that uses a frequency-domain objective function and another device with an objective function defined in the time-domain.

Methods

Overview of FDTD

FDTD is a grid-based, time-stepping method for solving partial differential equations. When applied to Maxwell's equations and electromagnetism, it is used to simulate light-matter interactions in the time-domain. FDTD is highly generalizable, well-suited for systems with multiple frequencies, time-varying properties, nonlinear materials, and can be integrated into multiphysics simulations. The electric and magnetic fields $\boldsymbol{E}$ and $\boldsymbol{H}$ are updated iteratively and alternately over a discretized simulation volume on the Yee grid, typically until the excitation fields are absorbed or escape the simulation volume. The space (∇) and time $\left(\frac{\partial}{\partial t}\right)$ derivatives in Maxwell's equations (Equations 1-2 for the lossless case) are first order and estimated using centered finite difference approximations. $\boldsymbol{J}_E$ is the electric current density, which typically acts as the input source for the simulation. μ is the permeability and ϵ is the permittivity.

$$\nabla \times \boldsymbol{E} = -\mu \frac{\partial \boldsymbol{H}}{\partial t} \tag{1}$$

$$\nabla \times \boldsymbol{H} = \epsilon \frac{\partial \boldsymbol{E}}{\partial t} + \boldsymbol{J}_E \tag{2}$$

For accuracy, FDTD requires a large number of spatial pixels and iteration timesteps. The spatial pixel size is typically chosen significantly smaller than the wavelength of interest λ (on the order of $\lambda/10$) (27). This directly constrains the maximum timestep through the Courant stability condition, which requires that the ratio between the timestep $c\Delta t$ and smallest spatial step Δx is smaller than $\frac{1}{\sqrt{3}}$ for the three-dimensional (3D) case with cubic pixels. The simulation then must be run for a sufficient time to let the excitation fields traverse the simulation volume or decay, which takes many calculation steps in the presence of resonances. The FDTD discretization and update equations used in this study are detailed in the **Supplementary Information**.

The time and memory cost of a gradient calculation for an optimization problem is broadly characterized by their scaling with respect to the number of independent input (N_{input}) and dependent output (N_{output}) parameters. Most optimization problems have a large N_{input} due the number of tunable parameters or fitting parameters, and a small N_{output} for the objective function (in most cases, $N_{output}=1$). It is thus ideal for a gradient calculation algorithm to have a time and memory cost that scales favorably with N_{input}, preferably being independent of it. Two such algorithms are the AVM and AD.

Overview of AVM

Sensitivity analysis in electromagnetism has its roots in the optimal design of dynamic mechanical systems (28,29) and was applied to design waveguide structures in the late 1990s (30,31). While AVM was formulated for simulators that solve systems of equations captured by large system matrices like FEM simulations, Y.-S. Chung et al demonstrated that AVM could also be applied to FDTD, which does not use system matrices and is defined on a structured Yee grid (32–34).

In this case, the AVM is employed to calculate the gradient $\frac{dG}{d\boldsymbol{p}}$ of an objective function $G = \int_\Omega \int_0^T g[\boldsymbol{E}(\boldsymbol{x},t), \boldsymbol{H}(\boldsymbol{x},t)] dt d^3\boldsymbol{x}$ that is defined over a volume Ω and simulation time T, with respect to a vector of tunable parameters $\boldsymbol{p}$. The derivation of AVM is described in literature (32–36), here, we will only summarize the implementation and associated challenges. The first step of AVM is to solve the forward problem in the time-domain: that is, to solve Maxwell's equations for a nominal geometry and store the field results $\boldsymbol{E}(\boldsymbol{x},t), \boldsymbol{H}(\boldsymbol{x},t)$ for the simulation time T, and for all positions that are either used in the objective function or associated with the tunable parameters $\boldsymbol{p}$ (*e.g.*, pixels over which a shape

derivative is desired). Although this may consume a large amount of memory, especially when there are many timesteps or grid positions for which g is non-zero, this memory consumption can be alleviated by checkpointing (25), where a smaller subset of time-steps are stored, and the simulation is run from these stored checkpoints to the required time-step. The integrand $g[\boldsymbol{E}(\boldsymbol{x},t),\boldsymbol{H}(\boldsymbol{x},t)]$ is differentiated analytically to obtain $\partial g/\partial \boldsymbol{E}(\boldsymbol{x},t), \partial g/\partial \boldsymbol{H}(\boldsymbol{x},t)$, which are time-dependent vector-valued functions evaluated using the time-varying stored field values from the forward simulation. These derivatives are then used as current sources for the adjoint simulation over the same geometry to get the adjoint electric and magnetic fields $\boldsymbol{E}^A(\boldsymbol{x},t), \boldsymbol{H}^A(\boldsymbol{x},t)$ with the modified Maxwell equations (37) in Equations 3-4, with the initial conditions being $\boldsymbol{E}^A(\boldsymbol{x},\tau=0)=\boldsymbol{0}, \boldsymbol{H}^A(\boldsymbol{x},\tau=0)=\boldsymbol{0}$ in terms of the reverse time $\tau = T - t$.

$$\boldsymbol{\nabla}\times\boldsymbol{E^A}(\boldsymbol{x},\boldsymbol{T}-\boldsymbol{t}) = -\boldsymbol{\mu}\frac{\boldsymbol{\partial H^A}(\boldsymbol{x},\boldsymbol{T}-\boldsymbol{t})}{\boldsymbol{\partial t}} + \frac{\boldsymbol{\partial g}}{\boldsymbol{\partial H}(\boldsymbol{x},\boldsymbol{T}-\boldsymbol{t})} \tag{3}$$

$$\nabla\times\boldsymbol{H}^A(\boldsymbol{x},T-t) = \epsilon\frac{\partial \boldsymbol{E}^A(\boldsymbol{x},T-t)}{\partial t} + \frac{\partial g}{\partial \boldsymbol{E}(\boldsymbol{x},T-t)} \tag{4}$$

The integrand derivatives $\frac{\partial g}{\partial \boldsymbol{E}(\boldsymbol{x},\tau)}, -\frac{1}{\mu}\frac{\partial g}{\partial \boldsymbol{H}(\boldsymbol{x},\tau)}$ thus play the role of time-varying electric and magnetic current density sources, respectively.

The final step of AVM is the combination of the forward and adjoint fields $\boldsymbol{E},\boldsymbol{H},\boldsymbol{E}^A,\boldsymbol{H}^A$ to obtain the gradient $dG/d\boldsymbol{p}$. Equation 5 shows the gradient element for the i-th tunable parameter p_i.

$$\frac{dG}{dp_i} = \frac{\partial G}{\partial p_i} - \int_\Omega d^3\boldsymbol{x}\int_0^T dt\left\{\boldsymbol{E}^A(\boldsymbol{x},t)\cdot\frac{\partial \boldsymbol{R}^E[\boldsymbol{E}(\boldsymbol{x},t)]}{\partial p_i} + \boldsymbol{H}^A(\boldsymbol{x},t)\cdot\frac{\partial \boldsymbol{R}^H[\boldsymbol{H}(\boldsymbol{x},t)]}{\partial p_i}\right\} \tag{5}$$

In FDTD, $\boldsymbol{R}^E$ and $\boldsymbol{R}^H$ are operators that must be derived manually for each specific system geometry, boundary condition, and tunable parameter. $\partial \boldsymbol{R}^{E,H}/\partial p_i$ can be well approximated for photonic shape optimization (37–40). However, for arbitrary optimizations, $\boldsymbol{R}^E$ and $\boldsymbol{R}^H$ may not have an analytic form and require careful mapping onto the FDTD results, see (36,38,41) and the **Supplementary Information**.

AVM for frequency-domain objectives

When the objective function $G = \int_\Omega\int_{\Delta\omega} g[\boldsymbol{E}(\boldsymbol{x},\omega),\boldsymbol{H}(\boldsymbol{x},\omega),\boldsymbol{E}^*(\boldsymbol{x},\omega),\boldsymbol{H}^*(\boldsymbol{x},\omega)]d\omega d^3\boldsymbol{x}$ is only written in terms of the electromagnetic fields in the frequency-domain (*i.e.*, the complex fields $\boldsymbol{E}(\boldsymbol{x},\omega),\boldsymbol{H}(\boldsymbol{x},\omega)$ and their complex conjugates $\boldsymbol{E}^*(\boldsymbol{x},\omega),\boldsymbol{H}^*(\boldsymbol{x},\omega)$) over a frequency bandwidth $\Delta\omega$ of positive frequencies, and when the parameter vector $\boldsymbol{p}$ represents the dielectric permittivities or permeabilities over a subset of pixels,

the adjoint system reduces to a much simpler form that can also be derived by exploiting Lorentz reciprocity symmetry between time-harmonic current sources and their fields (14).

$$\nabla \times \boldsymbol{E}^A(\boldsymbol{x}, T-t) = -\mu \frac{\partial \boldsymbol{H}^A(\boldsymbol{x}, T-t)}{\partial t} + 2\mathrm{Re} \int_{\Delta\omega} \frac{\partial g}{\partial \boldsymbol{H}(\boldsymbol{x}, \omega)} \exp\big(i\omega(T-t)\big)\, d\omega, \tag{6}$$

$$\nabla \times \boldsymbol{H}^A(\boldsymbol{x}, T-t) = \epsilon \frac{\partial \boldsymbol{E}^A(\boldsymbol{x}, T-t)}{\partial t} + 2\mathrm{Re} \int_{\Delta\omega} \frac{\partial g}{\partial \boldsymbol{E}(\boldsymbol{x}, \omega)} \exp(i\omega(T-t))\, d\omega. \tag{7}$$

Furthermore, the objective function gradient with respect to the permittivity ϵ_i at grid point i at position $\boldsymbol{x}_i$ simplifies to

$$\frac{dG}{d\epsilon_i} = \frac{1}{\pi} \mathrm{Re} \int_{\Delta\omega} \omega^2 \boldsymbol{E}^A(\boldsymbol{x}_i, \omega) \cdot \boldsymbol{E}^*(\boldsymbol{x}_i, \omega) d\omega\,. \tag{8}$$

Such conditions are well-suited for nanophotonic inverse design, in which dielectric distributions are designed to achieve specific optical functions at well-defined frequencies and frequency bands (7,14–16), and is therefore the form that is broadly employed. From an implementation point of view, the source terms in Equations (6) and (7) have a straightforward interpretation: for every discretized frequency $\omega \in \Delta\omega$ of interest, one has to place a point electric dipole with an amplitude of $\partial g/\partial \boldsymbol{E}$ and a point magnetic dipole with an amplitude of $-(1/\mu)\partial g/\partial \boldsymbol{H}$ at every position $\boldsymbol{x} \in \Omega$ (15). Because the dipole sources are time-harmonic, one does not need to record the full time-domain field during the forward simulation – it suffices to accumulate the frequency-domain complex fields at the intended dipole source positions and frequencies of interest (42,43). This greatly reduces the memory requirements compared to that of the adjoint procedure for a time-domain objective function. The full derivation of AVM for frequency-domain objectives is included in the **Supplementary Information**.

If the objective function or optimization parameters do not fall in this category, the full AVM described in Equation (5) must be used, which requires deep knowledge of its implementation and significant mathematical manipulation, limiting its application to specialists.

Methods

Overview of reverse mode automatic differentiation

Although AD has two modes of operation, forward and reverse mode, it is largely synonymous with the latter since RM AD scales linearly with N_{output} and is independent of N_{input}. Conversely, the forward mode

computational cost is independent of N_{output} but scales linearly with N_{input}. Therefore, we will focus on RM AD here. An intuitive explanation of RM AD's operation and scaling behavior is detailed in the **Supplementary Information**.

RM AD relies on the computation tree, also known as a tape or Wengert list (44). It contains all mathematical operations involved from the inputs to the outputs. Each node in the tree represents an intermediate elementary operation with a well-defined derivative associated with its own inputs and output. There are two passes in RM AD. The forward pass traverses the computational tree from the inputs to the outputs, storing all the intermediate values obtained. The backward pass, also known as backpropagation, performs the chain rule for differentiation from the outputs back towards the inputs, drawing upon the stored intermediate values. Backpropagation is the foundation of modern machine learning, as it provides the objective function gradients with respect to many tunable parameters (*e.g.*, weights and biases in neural networks) for iterative model training and refinement.

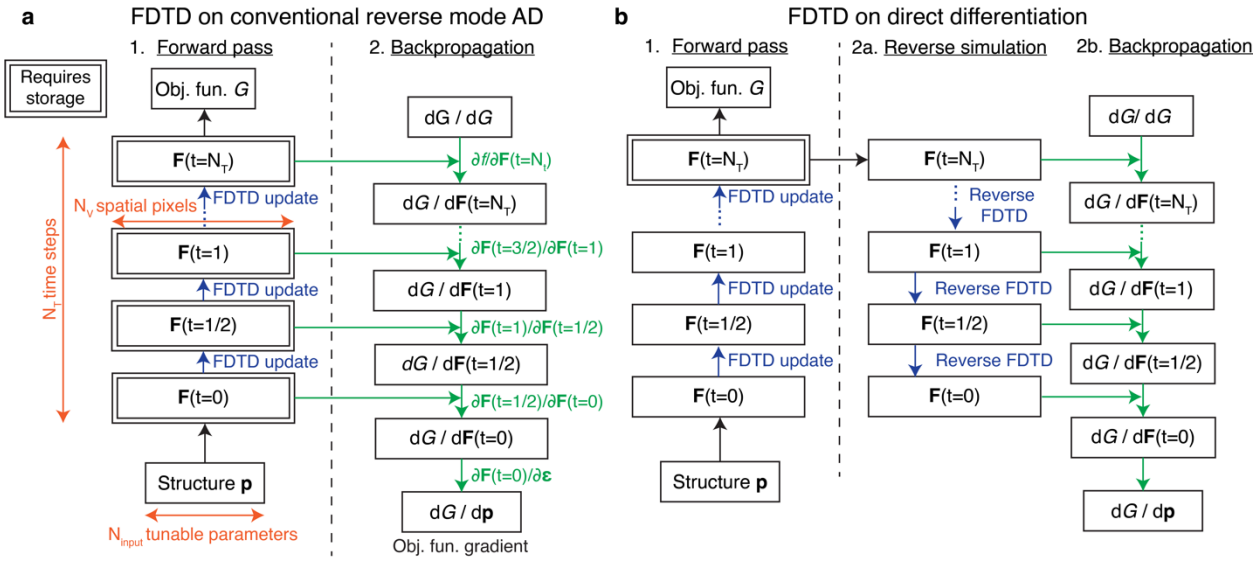

Figure 1: Comparison between conventional reverse mode automatic differentiation (AD) and direct differentiation (DD) | Both gradient calculation methods, which involve differentiating through FDTD update equations, are applied to the simulation of a structure (parametrized by vector $\boldsymbol{p}$) in a simulation volume comprising N_V pixels and N_T timesteps. In both cases, the desired output is the objective function gradient with respect to the structure vector $dG/d\boldsymbol{p}$. This is a simplified depiction: generally, the objective function is dependent on the fields at multiple timesteps and the parameter vector affects field behavior across multiple timesteps. **a** Conventional reverse mode AD has two steps: the forward pass and backpropagation. During the forward pass, all the electromagnetic field values $\boldsymbol{F}$ need to be stored (double box outline) for every space-time position so that they can be re-used during the backpropagation process, leading to substantial memory consumption. **b** For direct differentiation, only the fields at one

time step need to be stored. During backpropagation, a reverse simulation is run simultaneously to provide the necessary field values in the order required in backpropagation.

Consider the computation tree corresponding to an optimization calculation with an embedded FDTD calculation (**Figure 1a**). For ease of reading, from now on, we concatenate the electromagnetic fields $\boldsymbol{E}, \boldsymbol{H}$ to $\boldsymbol{F}$. The calculation proceeds from a parameter vector $\boldsymbol{p}$, which can be the vector of pixel fill factors, to the electromagnetic fields over the N_V spatial pixels at zero time $\boldsymbol{F}(0)$, through N_T timestep updates to the electromagnetic fields $\boldsymbol{F}(T)$ at the end time T, and finally to the objective function value G. The FDTD tree is simultaneously wide (due to the number of spatial pixels) and deep (due to the number of iteration timesteps). During the forward pass, every mathematical operation and intermediate field value is stored at considerable memory cost. This memory consumption peaks just after the objective value is computed. During backpropagation, the AD platform computes the derivative of the objective function G with respect to every intermediate parameter. This process proceeds in reverse order from the objective value back towards the structure vector at the root of the tree, and, importantly, draws upon the stored operations and intermediate field values in reverse order.

This layer-by-layer depiction of FDTD is a simplification but is valid even for objective functions that do not just depend on the fields at the final timestep $\boldsymbol{F}(T)$. In particular, to extract spectrally-resolved properties from FDTD, one explicitly accumulates partial sums of the discrete Fourier transform during the time-stepping updates (43). Thus, $\boldsymbol{F}$ can capture not only the electromagnetic fields over the simulation volume, but also partial Fourier transform sums, and it is valid to write the FDTD as a cascaded set of layers (each layer being associated with one timestep) with the same update equations between each layer.

Memory challenges in FDTD gradient calculations

Gradient calculation mode	Time complexity	Memory complexity
Finite difference	$O(N_{input} N_V N_T N_f)$	$O(N_{output} N_V N_f)$ *
Adjoint Variable Method	$O(N_{output} N_V N_T N_f)$ *	$O(N_{output} N_V N_f)$ *
Forward Mode AD	$O(N_{input} N_V N_T N_f)$	$O(N_{input} N_V N_f)$ *
Reverse Mode AD	$O(N_{output} N_V N_T N_f)$ *	$O(N_{output} N_V N_T N_f)$

Direct Differentiation	$O(N_{output} N_V N_T N_f) *$	$O(N_{output} N_V N_f + N_T \partial N_V) *$

Table 1: Time and memory scaling complexities for gradient calculation modes in FDTD simulations. Asterisks (*) indicate favorable scaling for high-dimensional inverse design, relative to the other gradient calculation modes. N_{input} and N_{output} represent the number of input and output parameters, respectively. N_V is the number of spatial pixels, N_T is the number of timesteps, and N_f is the number of frequency points. ∂N_V is the number of pixels on the recording layer at the spatial boundary of the simulation.

Table 1 displays the time and memory scaling behavior for various methods of gradient calculation applied to an FDTD system with N_{input} tunable parameters, N_{output} objective functions, N_V pixels, N_T timesteps, and N_f frequency points. Scaling behaviors that are favorable for high-dimensional inverse design (with its large N_{input}, N_V, N_T, small N_{output}) are labelled with asterisks (*). The scaling derivations are detailed in the **Supplementary Information.** The simplest gradient calculation, finite differences, exhibits poor time complexity as it scales linearly with N_{input}, but uses very little memory as it only needs to store the electromagnetic field values at the current timestep. Forward mode AD (45) has favorable memory scaling that is independent of the number of timesteps. However, it also suffers from poor N_{input} time complexity. RM AD has favorable time complexity independent of N_{input}, but has unfavorable memory complexity due to the large number of space-time pixels $N_V N_T$ for which the field values must be stored during the forward pass. For example, we find that a small ($N_V = 252 \times 180 \times 180$ pixels) 3D FDTD simulation based on the FDTD framework by Hughes et al (45) running on RM AD consumes 196 MB of memory per timestep when using the open-source AD tool Autograd (46). Typical FDTD simulations run beyond 1000 time steps, quickly raising memory needs into the hundreds of Gigabytes.

Summarizing, the adjoint method yields favorable time and memory complexity but requires significant mathematical manipulation for non-standard objective functions, as discussed earlier. Reverse mode automatic differentiation can be generally deployed by non-specialists, but requires access to the simulation source code at runtime (for knowledge of every mathematical operation) and is restricted to small problem sizes. In the following, we present Direct Differentiation, which offers simplified deployment and reasonable memory use.

Description of Direct Differentiation Operation

The structure of the FDTD computational tree motivates two key changes that can substantially reduce the memory cost of a gradient calculation relative to the RM AD architecture, forming the core of our Direct Differentiation (DD) platform (**Figure 1b**). First, the mathematical operations that update the field $\boldsymbol{F}(t), 1 \leq t \leq N_T$ from one timestep to the next are the same at every timestep. Similarly, the backpropagation operations which propagate the parameter gradients from $dG/d\boldsymbol{F}(t)$ to $dG/d\boldsymbol{F}(t-1/2)$ using the chain rule are time-independent. Instead of storing an individual copy of the update equations (and backpropagation operations) for every timestep, it suffices to store them once. This leads to a factor of N_T reduction in memory usage for storing the information encoded in the computation tree.

Second, we observe that Maxwell's equations are time-reversible (47) and that backpropagation during gradient calculation proceeds backwards from the objective function. The time-reversibility of Maxwell's equations (not to be confused with time inversion symmetry $t \to -t$) arises from the conservation of information under time evolution. This implies that the FDTD update equations can be rearranged to timestep fields in reverse, computing $\boldsymbol{F}\left(t - \frac{1}{2}\right)$ from $\boldsymbol{F}(t)$. Although FDTD on a finite domain does not conserve information due to the presence of lossy boundary conditions like Perfectly Matched Layers (PMLs), local information loss at boundaries can be prevented, as described below.

In DD, instead of storing all electromagnetic fields during the forward pass, we only store the fields at the final timestep $\boldsymbol{F}(T)$, reducing the peak memory required during the forward pass to that of a regular FDTD simulation. During backpropagation, we run a time-reversed simulation of the same FDTD domain (using $\boldsymbol{F}(T)$ as the initial conditions) in parallel with the chain rule gradient calculations, beginning at the final timestep $t = T$ and proceeding backwards in time. This time-reversed simulation supplies the required field values to the chain rule calculations at every timestep. Specifically, after a time-reversed simulation from $\boldsymbol{F}(t)$ to $\boldsymbol{F}(t-1/2)$, we can use $\boldsymbol{F}(t-1/2)$ to calculate $\frac{\partial \boldsymbol{F}(t)}{\partial \boldsymbol{F}\left(t-\frac{1}{2}\right)}$ and find $\frac{dG}{d\boldsymbol{F}\left(t-\frac{1}{2}\right)}$ using the chain rule $\frac{dG}{d\boldsymbol{F}\left(t-\frac{1}{2}\right)} = \frac{dG}{d\boldsymbol{F}(t)}\frac{\partial \boldsymbol{F}(t)}{\partial \boldsymbol{F}\left(t-\frac{1}{2}\right)} + \frac{\partial G}{\partial \boldsymbol{F}\left(t-\frac{1}{2}\right)}$. The $\frac{\partial G}{\partial \boldsymbol{F}\left(t-\frac{1}{2}\right)}$ term arises from the functional form of the objective function and changes for different simulations, though it can be performed using conventional RM AD if the objective function G is a complex functional of the field. This method of generating the intermediate

fields on-demand accounts for the bulk of the memory saving in DD relative to RM AD. The explicit equations for FDTD time stepping, time reversal, and gradient propagation for nondispersive media are detailed in the **Supplementary Information**. These time-reversal techniques can be generalized to dispersive media using techniques which parametrize frequency-dependent behavior using local variables that are updated at each timestep, such as the recursive convolution method (3).

In order to implement DD, the analytical $\frac{\partial g}{\partial \boldsymbol{E}(\boldsymbol{x},t)}, \frac{\partial g}{\partial \boldsymbol{H}(\boldsymbol{x},t)}, \frac{\partial \boldsymbol{E}(\boldsymbol{x},t)}{\partial \boldsymbol{p}}, \frac{\partial \boldsymbol{H}(\boldsymbol{x},t)}{\partial \boldsymbol{p}}$ derivatives must be known as a function of the fields at each timestep. This is similar to AVM which requires the analytical form of the objective function gradients $\frac{\partial g}{\partial \boldsymbol{E}(\boldsymbol{x},t)}, \frac{\partial g}{\partial \boldsymbol{H}(\boldsymbol{x},t)}$ and the operator derivatives $\frac{\partial \boldsymbol{R}^E}{\partial \boldsymbol{p}}, \frac{\partial \boldsymbol{R}^H}{\partial \boldsymbol{p}}$ to be known.

Handling lossy boundaries with a recording layer

In an infinitely large or periodically continued system the information content is conserved under the evolution dictated by Maxwell's equations. It is possible to start at the final system configuration and perform reverse time-stepping to perfectly re-attain the initial conditions. However, electromagnetic simulations are finite and use idealizations to capture the essential behavior without unnecessary computational cost. After the light-matter interaction of interest, the source fields are absorbed by infinitely lossy boundary conditions such as PMLs, which serve as an approximation of fields exiting to free space. At first glance, such an "empty" final configuration cannot be time-reversed, as the field information incident at these infinitely lossy boundaries is lost when the field amplitude decreases below machine precision. To address this issue, we introduce a single-pixel-thick recording layer at the surface of these infinitely lossy boundaries to capture the escaping time-dependent field. We therefore encode the system's 3D spatial initial state in the two-dimensional simulation boundary and one time dimension. During the time-reversal simulation, we replay these recorded fields as field sources from the recording layers. Full details of the recording layer implementation are found in the **Supplementary Information**. Non-infinitely lossy dispersive or absorptive media do not need a recording layer as long as the field amplitude remains larger than the machine precision.

Table 1 exhibits the time and memory complexity of DD, demonstrating that it has the same time complexity as RM AD and AVM but substantially less memory consumption than RM AD. In the limit of

large N_T, the peak memory usage of DD calculations scales linearly with the number of simulation time steps. This memory consumption arises from the storage operation of the recording boundary. Since the number of field positions associated with the recording boundary ∂N_V scales with the surface area of the simulation volume, the additional memory consumption per timestep scales with the square of the simulation box side length, instead of the cube as in conventional RM AD. The memory saving relative to RM AD thus becomes more pronounced for larger simulation volumes.

While undergoing peer review, we identified a patent application which describes a similar approach to propagating adjoint gradients through an FDTD system by exploiting time reversal, however, we did not find a related scientific publication (48).

Results

Validation of approach in 3D FDTD and Performance

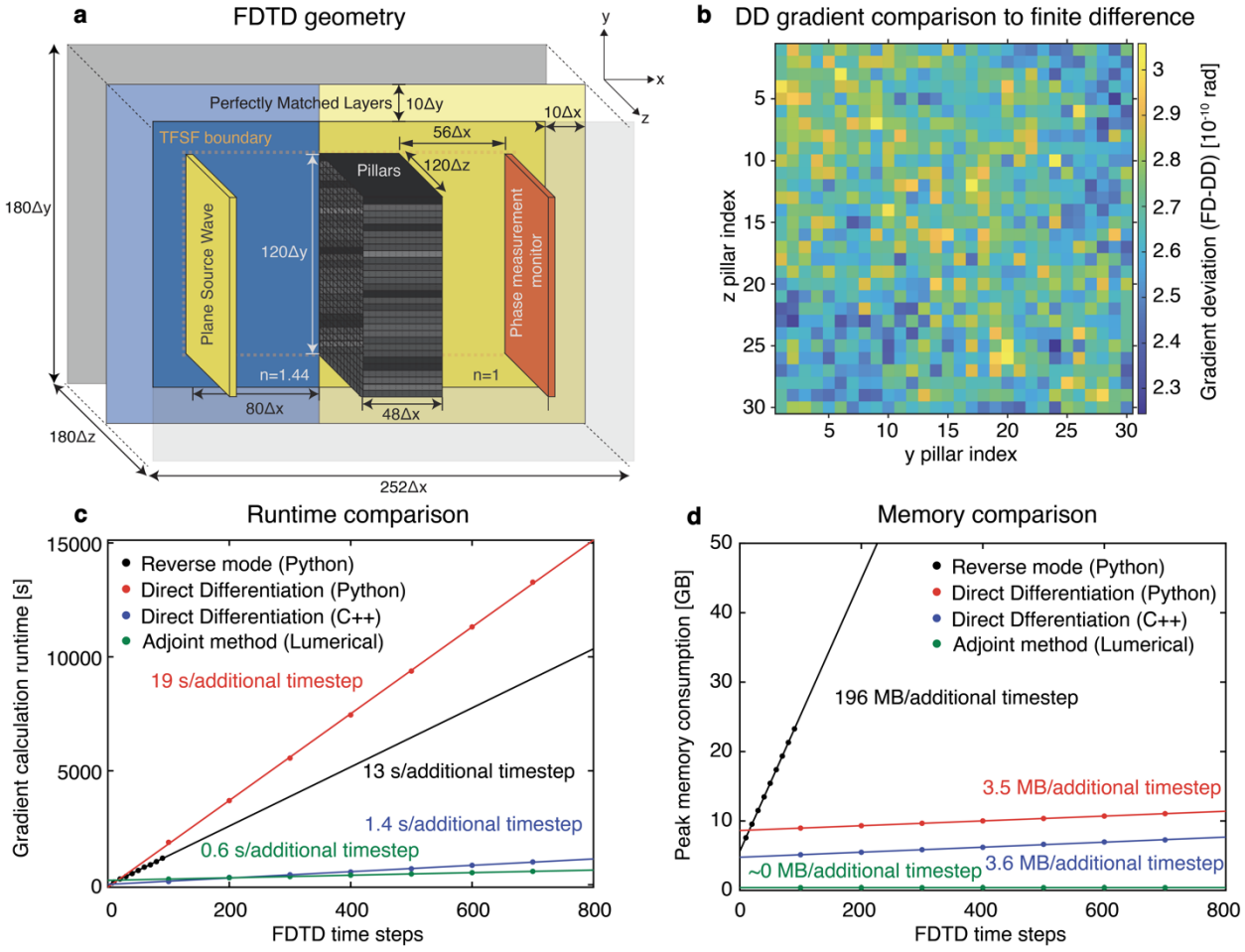

Figure 2: Implementation of 3D differentiable FDTD for Direct Differentiation | **a** 3D FDTD simulation geometry: The $252 \times 180 \times 180$ pixel domain ($6.5\lambda \times 4.7\lambda \times 4.7\lambda$) is partitioned into a glass volume (blue) and air volume (yellow). A single mode plane wave with central wavelength λ = 532 nm (yellow plane) is generated inside the glass medium. It is incident on a volume of pillars volume (gray volume) with permittivities $\epsilon_r(y,z)$, and the zeroth order phase ϕ of the transmitted field is recorded (orange plane). The simulation volume is surrounded by a Total Field Scattered Field (TFSF) boundary and perfectly matched layers. **b** Validation of the computed gradients $\frac{d\phi}{d\epsilon_r}$ from DD against that of finite difference calculations, for different pillar positions on the *y*-*z* plane. **c** Conventional reverse mode AD and DD do not have significant runtime differences when run on the same single-core Python platform. A speedup is obtained when DD is run on a compiled platform (C++), also on a single core, which has comparable runtime scaling with the adjoint variable method (AVM) run on a commercial FDTD suite (Lumerical FDTD, Ansys Inc.). **d** DD yields a 98% reduction in peak memory consumption per timestep compared to conventional reverse mode AD, when run on the same single core Python platform. AVM on a commercial FDTD simulation uses less memory as far fewer field locations need to be stored. The runtime and memory scaling with respect to the number of FDTD time steps are obtained by linear regression.

We validate the DD approach by differentiating a 3D FDTD and evaluating its gradient calculation accuracy, runtime, and memory scaling. Without loss of generality, we choose a 3D simulation region of 252×180×180 pixels and spatial pixel size $\Delta x = \Delta y = \Delta z = 13.75$ nm (**Figure 2a**, size $6.5\lambda \times 4.7\lambda \times 4.7\lambda, \lambda = 532$ nm). The structure under test is a $30 \times 30 = 900$ square pillar array (660 nm tall) on the *y*-*z* plane. Each of the 900 pillars has an individual real-valued relative permittivity (normalized to the permittivity of free space ϵ_0) and the 30 × 30-element matrix of the relative permittivities $\epsilon_r(\boldsymbol{x})$ serves as the structure vector of tunable parameters. The pillars are positioned on a glass substrate (n=1.44) in the *y*-*z* plane and are surrounded by air (n=1). The source field is a plane wave located within the glass substrate (vacuum wavelength $\lambda = 532$ nm, *z*-polarized, *x*-propagating). The objective function is the frequency-domain phase $\phi = \arg \sum_\Omega E$ of the complex electric field averaged over a 149 × 149 pixel monitor plane located 770 nm above the 900 pillars, which corresponds to the zeroth-order transmitted phase of the nanostructures. The objective function gradient is thus $\frac{d\phi}{d\epsilon_r(\boldsymbol{x})}$. The source plane, nanostructures, and monitor plane are surrounded by Total Field Scattered Field boundaries (49), which are in turn surrounded by PMLs on all sides. The recording boundary is coincident with the PML boundaries. We validated the accuracy of the FDTD simulation against a commercial FDTD suite (Lumerical FDTD, Ansys Inc.); details of this comparison are in the **Supplementary Information**.

We validate the gradients obtained through the DD approach using 900 single-sided finite difference calculations for a structure with random permittivity (**Figure 2b**) and find that the root-mean-squared absolute difference of the objective function gradient is negligible.

Although DD performs an additional time-reversal simulation during backpropagation compared to RM AD, runtime comparisons on a single CPU core using the same platform (Python) as a function of the number of FDTD timesteps shows similar scaling (**Figure 2c**), 19 s/timestep and 13 s/timestep for DD and RM AD, respectively. In terms of peak memory consumption, DD achieves a 98% memory usage reduction from 196 MB/timestep to 3.5 MB/timestep when compared to RM AD (**Figure 2d**). We demonstrate further gradient calculation speed-up with similar peak memory consumption by implementing DD on a compiled language (C++) instead of an interpreted language (Python), even for single CPU core computations (1.4 s/timestep and 3.6 MB/timestep). This two order-of-magnitude improvement in memory consumption in moving from RM AD to DD is consistent with the number of pixels along one spatial dimension of the 3D simulation volume.

We further compare the time and memory scaling of these differentiable platforms to AVM (run on a commercially optimized FDTD – Ansys Lumerical on a single CPU core) and derive the adjoint equations for the total field phase objective function using the Lorentz reciprocity approach in the **Supplementary Information**. The adjoint system is an array of dipoles at each of the phase measurement monitor pixels, each with a complex amplitude of $\frac{(\sum_\Omega E)^*}{|\sum_\Omega E|^2}$. The objective function derivative $\frac{d\phi}{d\epsilon_r(\boldsymbol{x})}$ for pixels with volume ΔV is obtained from the electric fields there from the forward simulation $E(\boldsymbol{x})$ and the adjoint simulation $E_A(\boldsymbol{x})$:

$$\frac{d\phi}{d\epsilon(\boldsymbol{x})} = \Delta V \ \mathrm{Im}[E^A(\boldsymbol{x})E(\boldsymbol{x})] \tag{9}$$

To reduce the required 149 × 149 = 22201 dipole sources at the monitor plane, which would take several hours for the software just to set up the simulation, we down-sampled the dipole source array to a 30 × 30

array. AVM achieves quicker runtime scaling by a factor of 2.3 (0.6 s/timestep vs 1.4 s/timestep) of the C++ implementation of DD (**Figure 2c**) and less memory consumption (4 kB/timestep vs 3.6 MB/timestep, **Figure 2d**). AVM for frequency-domain objective functions does not need to store values for each timestep and can accumulate partial sums, thus using very little additional memory with incremental timesteps.

Application to the inverse design of nanophotonic devices

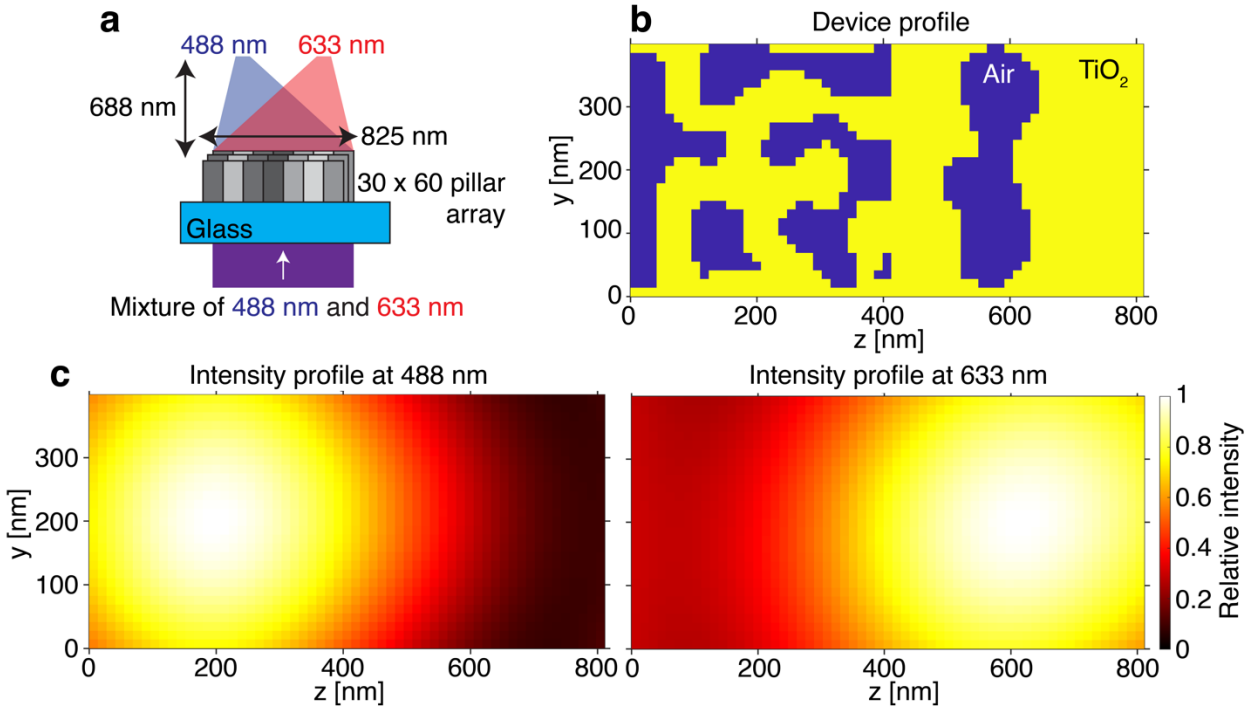

Figure 3: Design and performance of simple color sorter using frequency-domain optimization | **a** Schematic of color sorting function: incident illumination comprising a mixture of 488 nm and 633 nm wavelengths is spatially split into two distinct regions by a compact 30x60 array of pillars. The permittivity distribution over the pillar array is determined by inverse-design using the 3D direct differentiable FDTD. **b** Optimized binary titanium dioxide distribution over the 30x60 pillar array. **c** Intensity distribution $|E_x|^2+|E_y|^2+|E_z|^2$ at the monitor plane placed 50 pixels (687.5 nm) above the pillar array, under two illumination wavelengths, showing how the different wavelengths are deflected to two different regions. The intensity profiles are individually normalized to the maximum intensity on each plane.

With the performance and accuracy of DD established for FDTD, we deploy the direct differentiation model to inverse design multifunctional nanophotonic devices. We do this for two systems: one with frequency-domain behavior and another in the time-domain. While time-domain optical behavior can be equivalently

represented in the frequency-domain, it can capture dynamics across a wide frequency range without employing a large number of discrete frequency points (50).

For frequency-domain optimization, we aim to design an isolated passive structure that sorts incident illumination at two different wavelengths $\lambda_1 = 488$ nm, $\lambda_2 = 633$ nm into two distinct spatial regions, *i.e.*, it could act as a meta-optical color sorter that redirects incident light to different photodetector pixels placed close to the device (51). The desired behavior of the device is illustrated in **Figure 3a**: *x*-directed, *z*-polarized illumination is incident on a glass substrate on which a compact 30×60 pixel array of pillars lies in the *y*-*z* plane. The tunable parameters are the dielectric permittivities of the pixelated pillar array, and the objective function to be maximized for each wavelength is the overlap integral of the transmitted electric field profile with a target Gaussian field profile. We chose the target field profiles to ensure that 488 nm and 633 nm light are deflected in opposite *z*-directions, allowing the device to act as a near-field color sorter for these two wavelengths when evaluated in a plane placed 688 nm (50 pixels) above the pillars. The total system objective function to be maximized is the minimum of the two individual wavelength objective functions (see above), which ensures that the performance of the device at the two wavelengths remains comparable. We compute the objective function gradient with respect to the dielectric permittivity in the pixelated pillar array using the DD FDTD simulator and perform gradient descent optimization at a fixed learning rate. We apply a binarization term in the second half of the optimization to push the permittivities to the upper and lower bounds, which correspond to titanium dioxide and air, respectively. The optimized titanium dioxide profile is plotted in **Figure 3b**. Due to the proximity of the target plane to the top of the pillar array (around one wavelength away), the optimized permittivity profile reflects the left/right partitioning of the target plane. The transmitted intensity profiles for both wavelengths are plotted in **Figure 3c**, which demonstrates that the incident illumination is sorted into two spatial regions based on the wavelength. The optimization details are described in the **Supplementary Information**.

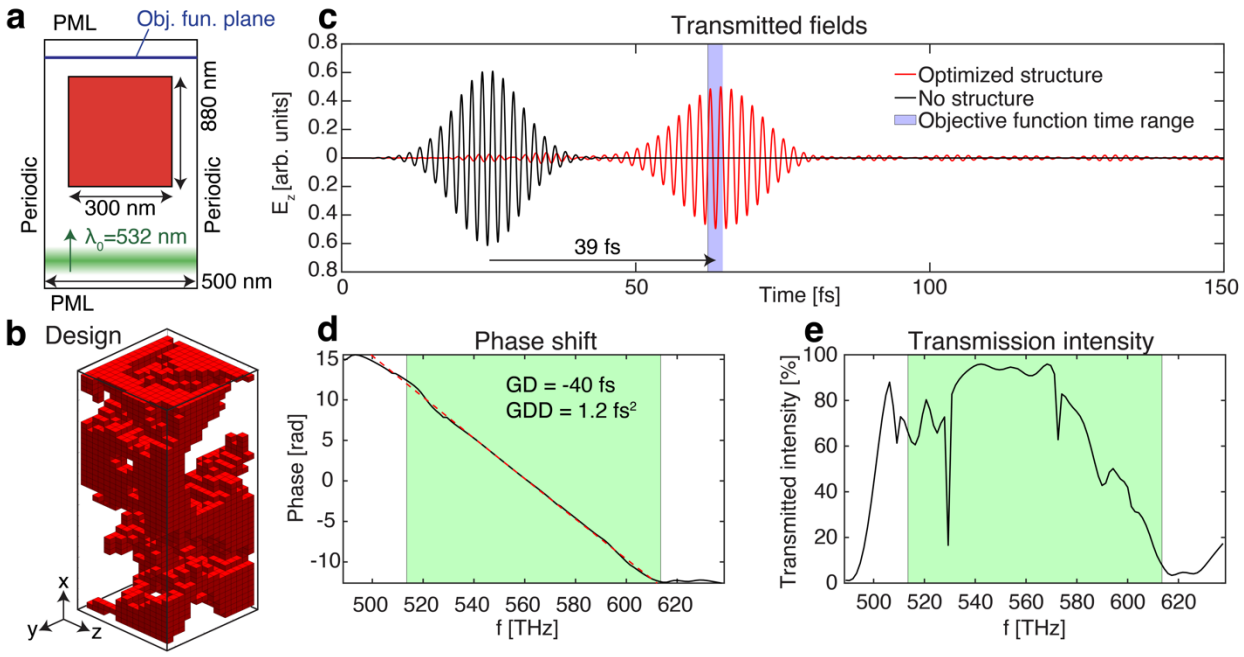

Figure 4: Design and performance of a resonator array using time-domain optimization. A unit cell resonator within a periodic array can be inverse-designed based on the time-domain profile of its transmitted field. **a** Schematic of the system to be optimized. *z*-polarized light with a central wavelength of 532 nm is incident on a block of dimensions $880 \times 300 \times 300$ nm ($44 \times 15 \times 15$ pixels) with periodic boundary conditions in the transverse plane. The objective function to be maximized is the transmitted field across the transverse plane, integrated over a subset of timesteps. **b** Rendering of unit cell optimized design, where the red solid blocks represent pixels filled with titanium dioxide. **c** Time-domain variation of the *z*-polarized field placed 500 nm from the array surface, averaged over the transverse plane, for the situation with and without the optimized structure, on the same normalized axis scale. The subset of timesteps used for objective function maximization is shaded, and is displaced 39 fs from the envelope peak of the nominal field without the structure. **d** Phase and **e** transmission intensity profiles of the optimized design based on Fourier transformation of the transmitted fields. The group delay (GD) and group delay dispersion (GDD) values over a 100 THz bandwidth is obtained by fitting the phase to a quadratic polynomial.

To demonstrate that DD can be straightforwardly extended to objective functions beyond the conventional frequency domain AVM, we perform a time-domain optimization. We design an array of resonators that impose a group delay on an incident pulse. Nanostructures with spatially-variant group delays are frequently employed in meta-optics to engineer behavior over a frequency band, such as achromatic metalenses which focus light within a given frequency range to a single focal point (52). Such nanostructures are designed through dispersion engineering, in which the transmission phase is designed to have a specific dependence on the illumination frequency, a process which typically entails simulation of nanostructure behavior over a dense set of frequencies, followed by polynomial regression on the transmitted spectral phase profile

(2,5,53). A linear phase dependence in the spectral phase represents a group delay on the pulse envelope, enabling such behavior to be engineered more directly in the time-domain. To our knowledge, such time-domain group delay topology optimization has only been done in the context of metasurfaces by Yasuda and Nishiwaki (54). To simulate an array of nanostructures, we replace the TFSF and PML boundary conditions in the transverse *y*-*z* directions with periodic boundary conditions (**Figure 4a**). Since these periodic boundary conditions are not lossy, they do not require recording layers. We define a time-domain objective function written in terms of the average *z*-polarized transmitted electric field across the transverse cross-section, which corresponds to the on-axis far-field projection of the transmitted fields. The objective value to be maximized is the average electric field for timesteps delayed between 37.4 to 39.9 fs from the peak of the nominal field without any nanostructure present (**Figure 4c** shaded area) and the tunable parameters are the dielectric permittivities for a $880 \times 300 \times 300$ nm block ($44 \times 15 \times 15 = 9900$ pixels, 20 nm pixel size) in the periodic unit cell. This temporal delay is chosen to be substantially larger than the 4.1 fs that obtained by a uniform slab of titanium dioxide (n=2.404) of the same thickness. The total simulation domain is $2560 \times 500 \times 500$ nm ($128 \times 25 \times 25$ pixels). The illumination source used is a pulse with central wavelength of 532 nm and a 13 fs full-width-at-half-maximum of the field amplitude envelope. The objective function gradients are used to update the pixel permittivities in the latent space using a fixed step size, with 3D gaussian blurring performed periodically to eliminate isolated pixels. Optimization details are in the **Supplementary Information** and the optimized structure is visualized in **Figure 4b**. The optimized structure successfully delays the incident pulse (**Figure 4c**) and the frequency-domain phase and amplitude profiles of the structure are plotted in **Figure 4d-e**, respectively. The phase exhibits the desired linear decrease over a 100 THz bandwidth with an estimated group delay of 39.7 fs and group delay dispersion of 1.25 fs^2. The transmission intensity profile indicates the existence of several resonances in the frequency band that are responsible for the large group delay.

The inverse design framework presented here is an initial proof-of-concept and can be augmented using other topology optimization tactics such as the incorporation of fabrication tolerances and level-set representations for curved structures (7,9,55,56). These modifications will remove features that are difficult to fabricate using conventional lithography techniques, such as the small L-shaped island in the lower center region of **Figure 3b**.

Conclusion

The DD FDTD architecture enables nanophotonic devices to be modelled differentiably at substantially lower memory cost compared to conventional RM AD. This potentially allows these computations to be performed rapidly and in parallel on a single graphics processing unit (GPU) instead of dedicated large-memory enterprise computing clusters. The DD architecture can be more generally applied to any grid-like simulator that can be reverse time-stepped, such as quantum mechanical wavefunctions in space, low Reynolds number fluid dynamics, and dissipationless solid mechanics. DD paves the way for the creation of differentiable simulators incorporating multiple coupled physical influences and performing high-dimensional optimization over the control parameters governing these dynamics.

Supporting Information

Residual operators in AVM, AVM for frequency-domain objectives, Reverse mode automatic differentiation description, Time and memory scaling derivations for gradient calculation methods, Direct Differentiation Key equations, Recording layer implementation, Validation against commercial FDTD software suite, Adjoint method derivations for field phase, Design of color sorter, Design of resonator array.

Acknowledgements

S.W.D.L. is supported by A*STAR Singapore through the National Science Scholarship Scheme. The authors thank Noah Rubin from Harvard University for helpful conversations.

Funding Sources

This project is supported by the Defense Advanced Research Projects Agency (DARPA), grant HR00111810001. We acknowledge financial support from the Air Force Office of Scientific Research (AFOSR) MURI program, grant number FA9550-21-1-0312. The computations in this paper were run on the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University.

Author Contributions

RJT and SWDL developed the DD algorithm. RJT implemented and validated the algorithm with input from SWDL and MO. XY and MO clarified the applications of the algorithm. RJT and SWDL wrote the manuscript with input from MO and XY. MO and FC supervised the research.

Competing Interests statement

A provisional patent application based on this work is pending. The authors declare no other competing interests.

Data Availability Statement

The source data for all manuscript figures is provided with the paper.

Code Availability Statement

The source code of the differentiable FDTD is provided with the paper.

References

(1) Yu, N.; Capasso, F. Flat Optics with Designer Metasurfaces. *Nat Mater* **2014**, *13* (2), 139–150.

(2) Chen, W. T.; Zhu, A. Y.; Capasso, F. Flat Optics with Dispersion-Engineered Metasurfaces. *Nat Rev Mater* **2020**, *5* (8), 604–620.

(3) Taflove, A.; Susan C. Hagness. *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 3rd ed.; Artech House Publishers: Norwood, MA, 2000.

(4) Bayati, E.; Pestourie, R.; Colburn, S.; Lin, Z.; Johnson, S. G.; Majumdar, A. Inverse Designed Extended Depth of Focus Meta-Optics for Broadband Imaging in the Visible. *Nanophotonics* **2022**, *11* (11), 2531–2540.

(5) Shrestha, S.; Overvig, A. C.; Lu, M.; Stein, A.; Yu, N. Broadband Achromatic Dielectric Metalenses. *Light Sci Appl* **2018**, *7* (1), 85.

(6) Campbell, S. D.; Sell, D.; Jenkins, R. P.; Whiting, E. B.; Fan, J. A.; Werner, D. H. Review of Numerical Optimization Techniques for Meta-Device Design [Invited]. *Opt Mater Express* **2019**, *9* (4), 1842.

(7) Li, Z.; Pestourie, R.; Lin, Z.; Johnson, S. G.; Capasso, F. Empowering Metasurfaces with Inverse Design: Principles and Applications. *ACS Photonics* **2022**, *9* (7), 2178–2192.

(8) Liu, D.; Tan, Y.; Khoram, E.; Yu, Z. Training Deep Neural Networks for the Inverse Design of Nanophotonic Structures. *ACS Photonics* **2018**, *5* (4), 1365–1369.

(9) Jiang, J.; Fan, J. A. Global Optimization of Dielectric Metasurfaces Using a Physics-Driven Neural Network. *Nano Lett* **2019**, *19* (8), 5366–5372.

(10) Kudyshev, Z. A.; Kildishev, A. V.; Shalaev, V. M.; Boltasseva, A. Machine-Learning-Assisted Metasurface Design for High-Efficiency Thermal Emitter Optimization. *Appl Phys Rev* **2020**, *7* (2), 021407.

(11) Ma, W.; Liu, Z.; Kudyshev, Z. A.; Boltasseva, A.; Cai, W.; Liu, Y. Deep Learning for the Design of Photonic Structures. *Nat Photonics* **2021**, *15* (2), 77–90.

(12) Kudyshev, Z. A.; Kildishev, A. V.; Shalaev, V. M.; Boltasseva, A. Machine Learning–Assisted Global Optimization of Photonic Devices. *Nanophotonics* **2020**, *10* (1), 371–383.

(13) Peurifoy, J.; Shen, Y.; Jing, L.; Yang, Y.; Cano-Renteria, F.; DeLacy, B. G.; Joannopoulos, J. D.; Tegmark, M.; Soljačić, M. Nanophotonic Particle Simulation and Inverse Design Using Artificial Neural Networks. *Sci Adv* **2018**, *4* (6), 1–8.

(14) Lalau-Keraly, C. M.; Bhargava, S.; Miller, O. D.; Yablonovitch, E. Adjoint Shape Optimization Applied to Electromagnetic Design. *Opt Express* **2013**, *21* (18), 21693.

(15) Miller, O. D. Photonic Design: From Fundamental Solar Cell Physics to Computational Inverse Design, University of California, Berkeley, 2013.

(16) Molesky, S.; Lin, Z.; Piggott, A. Y.; Jin, W.; Vucković, J.; Rodriguez, A. W. Inverse Design in Nanophotonics. *Nat Photonics* **2018**, *12* (11), 659–670.

(17) Mansouree, M.; Kwon, H.; Arbabi, E.; McClung, A.; Faraon, A.; Arbabi, A. Multifunctional 2.5D Metastructures Enabled by Adjoint Optimization. *Optica* **2020**, *7* (1), 77.

(18) Colburn, S.; Majumdar, A. Inverse Design and Flexible Parameterization of Meta-Optics Using Algorithmic Differentiation. *Commun Phys* **2021**, *4* (1), 65.

(19) Minkov, M.; Williamson, I. A. D.; Andreani, L. C.; Gerace, D.; Lou, B.; Song, A. Y.; Hughes, T. W.; Fan, S. Inverse Design of Photonic Crystals through Automatic Differentiation. *ACS Photonics* **2020**, *7* (7), 1729–1741.

(20) Jin, W.; Li, W.; Orenstein, M.; Fan, S. Inverse Design of Lightweight Broadband Reflector for Relativistic Lightsail Propulsion. *ACS Photonics* **2020**, *7* (9), 2350–2355.

(21) Nørgaard, S. A.; Sagebaum, M.; Gauger, N. R.; Lazarov, B. S. Applications of Automatic Differentiation in Topology Optimization. *Structural and Multidisciplinary Optimization* **2017**, *56* (5), 1135–1146.

(22) Zhelyeznyakov, M.; Fröch, J.; Wirth-Singh, A.; Noh, J.; Rho, J.; Brunton, S.; Majumdar, A. Large Area Optimization of Meta-Lens via Data-Free Machine Learning. *Communications Engineering* **2023**, *2* (1), 60.

(23) Jesse Lu; Jelena Vuckovic. Jaxwell: GPU-Accelerated, Differentiable 3D Iterative FDFD Electromagnetic Solver. 2020.

(24) Su, L.; Vercruysse, D.; Skarda, J.; Sapra, N. V.; Petykiewicz, J. A.; Vučković, J. Nanophotonic Inverse Design with SPINS: Software Architecture and Practical Considerations. *Appl Phys Rev* **2020**, *7* (1), 011407.

(25) Wang, Q.; Moin, P.; Iaccarino, G. Minimal Repetition Dynamic Checkpointing Algorithm for Unsteady Adjoint Calculation. *SIAM Journal on Scientific Computing* **2009**, *31* (4), 2549–2567.

(26) Lazarov, B. S.; Matzen, R.; Elesin, Y. Topology Optimization of Pulse Shaping Filters Using the Hilbert Transform Envelope Extraction. *Structural and Multidisciplinary Optimization* **2011**, *44* (3), 409–419.

(27) Zheng, F.; Chen, Z.; Zhang, J. A Finite-Difference Time-Domain Method without the Courant Stability Conditions. *IEEE Microwave and Guided Wave Letters* **1999**, *9* (11), 441–443.

(28) Haug, E. J.; Arora, J. S. Design Sensitivity Analysis of Elastic Mechanical Systems. *Comput Methods Appl Mech Eng* **1978**, *15* (1), 35–62.

(29) Michaleris, P.; Tortorelli, D. A.; Vidal, C. A. Tangent Operators and Design Sensitivity Formulations for Transient Non-Linear Coupled Problems with Applications to Elastoplasticity. *Int J Numer Methods Eng* **1994**, *37* (14), 2471–2499.

(30) Hong-Bae Lee; Hyun-Kyo Jung; Song-yop Hahn; Changyul Cheon; Ki-Sik Lee. Shape Optimization of H-Plane Waveguide Tee Junction Using Edge Finite Element Method. *IEEE Trans Magn* **1995**, *31* (3), 1928–1931.

(31) Youngho Lee; Changyul Cheon; Hyeong-seok Kim. Shape Optimization of W/G Structure for Uniform Field Illumination. *IEEE Trans Magn* **1998**, *34* (5), 3584–3587.

(32) Young-Seek Chung; Changyul Cheon; Il-Han Park; Song-Yop Hahn. Optimal Shape Design of Microwave Device Using FDTD and Design Sensitivity Analysis. *IEEE Trans Microw Theory Tech* **2000**, *48* (12), 2289–2296.

(33) Young-Seek Chung; Jechung Ryu; Changyul Cheon; Il-Han Park; Song-Yop Hahn. Optimal Design Method for Microwave Device Using Time Domain Method and Design Sensitivity Analysis. I. FETD Case. *IEEE Trans Magn* **2001**, *37* (5), 3289–3293.

(34) Young-Seek Chung; Changyul Cheon; Il-Han Park; Song-Yop Hahn. Optimal Design Method for Microwave Device Using Time Domain Method and Design Sensitivity Analysis. II. FDTD Case. *IEEE Trans Magn* **2001**, *37* (5), 3255–3259.

(35) Zhang, Y.; Bakr, M. H. FDTD-Based Transient Adjoint Sensitivity Analysis. In *2015 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*; IEEE, 2015; pp 1–3.

(36) Guoqiang Shen; Tam, H. W. W.; Nikolova, N. K.; Bakr, M. H. Adjoint Sensitivity Technique for FDTD Methods on Structured Grids. In *IEEE Antennas and Propagation Society International Symposium. Digest. Held in conjunction with: USNC/CNC/URSI North American Radio Sci. Meeting (Cat. No.03CH37450)*; IEEE, 2003; Vol. 3, pp 746–749.

(37) Yasuda, H.; Yamada, T.; Nishiwaki, S. A Design Method of Spatiotemporal Optical Pulse Using Level-Set Based Time Domain Topology Optimization. *Int J Numer Methods Eng* **2019**, *117* (6), 605–622.

(38) Nikolova, N. K.; Tam, H. W.; Bakr, M. H. Sensitivity Analysis With the FDTD Method on Structured Grids. *IEEE Trans Microw Theory Tech* **2004**, *52* (4), 1207–1216.

(39) Swillam, M. A.; Bakr, M. H.; Nikolova, N. K.; Li, X. Adjoint Sensitivity Analysis of Dielectric Discontinuities Using FDTD. *Electromagnetics* **2007**, *27* (2–3), 123–140.

(40) Bakr, M. H.; Nikolova, N. K. An Adjoint Variable Method for Frequency Domain TLM Problems with Conducting Boundaries. *IEEE Microwave and Wireless Components Letters* **2003**, *13* (9), 408–410.

(41) Nikolova, N. K.; Ying Li; Yan Li; Bakr, M. H. Sensitivity Analysis of Scattering Parameters with Electromagnetic Time-Domain Simulators. *IEEE Trans Microw Theory Tech* **2006**, *54* (4), 1598–1610.

(42) Yunpeng Song; Nikolova, N. K. Memory-Efficient Method for Wideband Self-Adjoint Sensitivity Analysis. *IEEE Trans Microw Theory Tech* **2008**, *56* (8), 1917–1927.

(43) Sullivan, D. M. *Electromagnetic Simulation Using the FDTD Method*; IEEE Press: New York, 2000.

(44) Wengert, R. E. A Simple Automatic Derivative Evaluation Program. *Commun ACM* **1964**, *7* (8), 463–464.

(45) Hughes, T. W.; Williamson, I. A. D.; Minkov, M.; Fan, S. Forward-Mode Differentiation of Maxwell's Equations. *ACS Photonics* **2019**, *6* (11), 3010–3016.

(46) Maclaurin, D.; Duvenaud, D.; Johnson, M.; Townsend, J. *Autograd*. https://github.com/HIPS/autograd

(47) Sorrentino, R.; Roselli, L.; Mezzanotte, P. Time Reversal in Finite Difference Time Domain Method. *IEEE Microwave and Guided Wave Letters* **1993**, *3* (11), 402–404.

(48) Brian Adolf; Martin Schubert; Jesse Lu. Physical Device Optimization with Reduced Memory Footprint via Time Reversal at Absorbing Boundaries. US 2021/0357554 A1, July 2021.

(49) Merewether, D. E.; Fisher, R.; Smith, F. W. On Implementing a Numeric Huygen's Source Scheme in a Finite Difference Program to Illuminate Scattering Bodies. *IEEE Trans Nucl Sci* **1980**, *27* (6), 1829–1833.

(50) Yang, L.; Lavrinenko, A. V.; Hvam, J. M.; Sigmund, O. Design of One-Dimensional Optical Pulse-Shaping Filters by Time-Domain Topology Optimization. *Appl Phys Lett* **2009**, *95* (26), 1–3.

(51) Camayd-Muñoz, P.; Ballew, C.; Roberts, G.; Faraon, A. Multifunctional Volumetric Meta-Optics for Color and Polarization Image Sensors. *Optica* **2020**, *7* (4), 280.

(52) Chen, W. T.; Zhu, A. Y.; Sisler, J.; Huang, Y. W.; Yousef, K. M. A.; Lee, E.; Qiu, C. W.; Capasso, F. Broadband Achromatic Metasurface-Refractive Optics. *Nano Lett* **2018**, *18* (12), 7801–7808.

(53) Ossiander, M.; Huang, Y.-W.; Chen, W. T.; Wang, Z.; Yin, X.; Ibrahim, Y. A.; Schultze, M.; Capasso, F. Slow Light Nanocoatings for Ultrashort Pulse Compression. *Nat Commun* **2021**, *12* (1), 6518.

(54) Yasuda, H.; Nishiwaki, S. A Design Method of Broadband Metalens Using Time-Domain Topology Optimization. *AIP Adv* **2021**, *11* (5), 055116.

(55) Sell, D.; Yang, J.; Doshay, S.; Yang, R.; Fan, J. A. Large-Angle, Multifunctional Metagratings Based on Freeform Multimode Geometries. *Nano Lett* **2017**, *17* (6), 3752–3757.

(56) Li, Z.; Pestourie, R.; Park, J.-S.; Huang, Y.-W.; Johnson, S. G.; Capasso, F. Inverse Design Enables Large-Scale High-Performance Meta-Optics Reshaping Virtual Reality. *Nat Commun* **2022**, *13* (1), 2409.

For Table of Contents Use Only

Manuscript title: Time reversal differentiation of FDTD for photonic inverse design

Authors: Rui Jie Tang, Soon Wei Daniel Lim, Marcus Ossiander, Xinghui Yin, Federico Capasso

Brief synopsis describing graphic: Unit cell of a resonator array designed using time-domain topology optimization with direct differentiation.

TOC Graphic: