Massively parallel pixel-by-pixel nanophotonic optimization using a Green’s function formalism

Jiahui Wang,1,2,* Alfred K. C. Cheung,1,*,† Aleksandra Spyra,1 Ian A. D. Williamson,1 Jian Guan,1 and Martin F. Schubert1
1) X, 100 Mayfield Ave, Mountain View, CA 94043, USA
2) Department of Applied Physics, Stanford University, 348 Via Pueblo Mall, Stanford, CA 94305, USA

We introduce an efficient parallelization scheme to implement pixel-by-pixel nanophotonic optimization using a Green’s function based formalism. The crucial insight in our proposal is the reframing of the optimization algorithm as a large-scale data processing pipeline, which allows for the efficient distribution of computational tasks across thousands of workers. We demonstrate the utility of our implementation by exercising it to optimize a high numerical aperture focusing metalens at problem sizes that would otherwise be far out of reach for the Green’s function based method. Finally, we highlight the connection to powerful ideas from reinforcement learning as a natural corollary of reinterpreting the nanophotonic inverse design problem as a graph traversal enabled by the pixel-by-pixel optimization paradigm.

I. INTRODUCTION

The promise of photonic inverse design is to enable the optimization of non-intuitive photonic structures that achieve superior device performance (e.g. lower losses, larger bandwidths) within substantially more compact device footprints. This requires the effective exploration of complex and high dimensional design spaces to ultimately arrive at designs which are both performant and fabricable by modern foundry manufacturing processes. The fabricability requirements are crucial and broadly entail the enforcement of: (a) a binary condition on the value of the permittivity at each design pixel (assuming, as in typical examples, two possible materials), and (b) compliance with minimum feature size constraints for printed features as required by manufacturing processes. As such, optimization schemes that are able to maintain one or both fabricability requirements throughout the entire course of an optimization are especially attractive.

An always-feasible design is notably not a property of the most widely used class of optimization methods for photonic inverse design, which rely on continuous optimization algorithms and the adjoint variable method (AVM)1–5. In the AVM, changes to the design are made based on the gradient of the device figure of merit (FOM) with respect to the permittivity at each design pixel. The gradient provides the response of the FOM with respect to infinitesimally small, continuous changes in the permittivity at each pixel. The gradient at a pixel will not, in general, reflect the actual change in the FOM that results from discrete, arbitrarily large changes in the permittivity. Therefore, while formalisms have been developed that utilize the gradient in ways that try to drive the design towards fabricability4–8, there are no strict guarantees, and there is typically a strong trade off between driving towards fabricability and device performance.

An alternative optimization paradigm is to instead frame the optimization problem as a pixel-by-pixel graph traversal scheme, where every node of a graph is a possible design. Specifically, the root node design could be a featureless design, and the children of any design are all possible modifications to the design. Conceptually, as an implementation of this strategy, one might at each optimization step simulate a set of designs that result from discrete variations in the permittivity at one or multiple pixels, calculate the resulting changes in FOM, and greedily update the design to the variation achieving the largest positive change in FOM. The pixel-by-pixel paradigm possesses the advantage of always satisfying the binary condition. Furthermore, by imposing additional rules on how the actions are selected, minimum feature size design rules can also be easily enforced as a byproduct of the graph traversal9. Finally, the reinterpretation of the photonic inverse design problem as a graph traversal naturally lends itself to application of ideas from reinforcement learning (RL)—a strategy which has proven to be quite fruitful across a broad spectrum of domains in recent years10,11. We seek to showcase this idea with a demonstration later in this work.

A brute force implementation of the pixel-by-pixel optimization paradigm based on numerous (full-wave) simulations of the design variations at each optimization step is extremely computationally intensive and is limited to the optimization of small systems12. In Ref. 13, Boutami and Fan introduce a pixel-by-pixel optimization formalism based on the Green’s function technique14,15 for solving the electromagnetic scattering problem that circumvents the need for any full-wave simulations. The authors demonstrate that if the Green’s function is known for a given structure, the change in FOM for discrete variations of the structure can be efficiently evaluated. Subsequently, after a design variation is selected based on the change in FOM information, the Green’s function can

* These authors contributed equally to this work.
† Corresponding author: alfredkcc@x.team
also be updated efficiently. Thus, as long as the Green’s function for the initial structure is known, the pixel-by-pixel optimization scheme can proceed without the need for any full-wave simulations.

The Green’s function formalism is difficult to scale to even moderately sized systems due to the formidable memory cost (scaling as $\mathcal{O}(N^2)$ where $N$ is the total number of pixels in the design region) of storing the full two-point Green’s function throughout the optimization. The authors of Ref. 13 recognized this and concurrently introduced a modified formalism in Ref. 16 where, instead of having to store the full Green’s function for all pairwise combinations of positions, only the position diagonal elements are required in exchange for a small number of full-wave simulations needed at each optimization step. This drastically reduces the memory cost to $\mathcal{O}(N)$. This alternative implementation was exercised to optimize for 3D silicon-on-insulator waveguide bends which were fabricated and measured in Ref. 17.

While the memory-reduced alternative implementation of Ref. 16 is undoubtedly promising, it comes with the cost of additional computational complexity associated with the reintroduction of full-wave simulations at each step. Indeed, there is a unique appeal to the original implementation of Ref. 13 in forgoing any full-wave simulations. This sets the formalism apart from practically all other topology optimization schemes.

In this work, we demonstrate that it is possible to scale to large design problems ($\sim 10^6$ design pixels) using the formalism introduced in Ref. 13 in spite of the $\mathcal{O}(N^2)$ memory scaling for storing the full Green’s function. The crucial insight is the observation that the structure of the optimization scheme fits naturally within large scale data processing models$^{18,19}$ wherein the storage, change in FOM calculations, and Green’s function updates are all trivial to chunk and parallelize across thousands of workers or more on computer clusters. We will show that our massively parallel implementation of the Green’s function based pixel-by-pixel optimization scheme is a viable option for photonic inverse design. We further provide reasons for why it is particularly well suited for application to the inverse design of metalenses.

This paper is organized as follows. In Sec. II, we review the formalism behind the Green’s function pixel-by-pixel optimization method. In Sec. III, we then demonstrate how the optimization scheme can be massively parallelized through an implementation that uses common data processing frameworks such as the open source Apache Beam$^{19,20}$ which conforms to the Dataflow$^{21}$ model. In Sec. IV, we turn to an application of our implementation to the design of a high numerical aperture (NA) metalens. Section V provides a summary and further discussion of the results, with a particular emphasis on the connections of pixel-by-pixel optimization with fundamental concepts in RL.

II. FORMALISM

Consider a three-dimensional photonic inverse design problem in which the design degrees of freedom are the discrete set of permittivity values (corresponding to a set of possible materials) at $N$ pixel locations within a design region $\mathcal{D}$, where each pixel has volume $\Delta V$. For simplicity, let us restrict our attention to the case of two materials. We refer to the first material as the patterned material with relative permittivity $\epsilon$ and the second material as the background material with relative permittivity $\epsilon_0$, and define $\Delta \epsilon \equiv \epsilon - \epsilon_0$. Let $\lambda$ be the operating wavelength in the background material and $k_0 = 2\pi/\lambda$. Finally, let the FOM be a scalar function $F$ of the electric field at $N_m$ monitor positions $\{\mathbf{r}_{m,i} : i = 1, \ldots, N_m\}$:

$$\text{FOM} = F(\mathbf{E}(\mathbf{r}_{m,1}), \ldots, \mathbf{E}(\mathbf{r}_{m,N_m})).$$

The goal of the optimization is to maximize the value of the FOM by sequentially proposing discrete modifications to the design and selecting the modifications based on exact $\Delta$FOM values. The main idea behind the Green’s function formalism is that, if the Green’s function $G(\mathbf{r},\mathbf{r'})$ is known for all pairs of positions $\mathbf{r}, \mathbf{r'} \in \mathcal{D} \cup \{\mathbf{r}_{m,i}\}$ for a structure, then $\Delta$FOM values for single or multiple pixel modifications to the structure can be calculated efficiently without the need for any full-wave simulations.

The starting point is the Green’s function technique for solving the electromagnetic scattering problem$^{14,15}$. Consider a reference structure (i.e. a distribution of background and patterned material) for which the Green’s function $G_0(\mathbf{r},\mathbf{r'})$ is known. The Green’s function $G(\mathbf{r},\mathbf{r'})$ corresponding to a modification of the reference structure where background material in a region $V \subset \mathcal{D}$ is flipped to patterned material can be self-consistently related to $G_0(\mathbf{r},\mathbf{r'})$ via Dyson’s equation$^{13,14}$:

$$G(\mathbf{r},\mathbf{r'}) = G_0(\mathbf{r},\mathbf{r'}) + \int_V d\mathbf{r}'' G_0(\mathbf{r},\mathbf{r}'') \cdot k_0^2 \Delta \epsilon(\mathbf{r}''),$$

(2)

The electric fields for the reference structure, $\mathbf{E}_0(\mathbf{r})$, and for the modified structure, $\mathbf{E}(\mathbf{r})$, obey the Lippman-Schwinger equation$^{13,14}$:

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}_0(\mathbf{r}) + \int_V d\mathbf{r}'' G_0(\mathbf{r},\mathbf{r}'') \cdot k_0^2 \Delta \epsilon(\mathbf{r}'').$$

(3)

A. Calculating changes in the FOM

We now reinterpret Eqs. (2) and (3) within the context of pixel-by-pixel optimization by regarding $\mathbf{E}_0(\mathbf{r})$ and $G_0(\mathbf{r},\mathbf{r'})$ as the known field and Green’s function for the structure at the start of a given optimization step. We propose a modified structure where pixels at positions $\mathbf{r}''$ within a region $V$ are flipped. Discretizing Eq. (3) and
The region that results from this structural modification is:

\[ E(\mathbf{r}) = E_0(\mathbf{r}) + b \sum_{\mathbf{r}'' \in V} G_0(\mathbf{r}', \mathbf{r}'') \cdot E(\mathbf{r}''), \]

where, for convenience, we have defined the constant \( b \equiv k_0^2 \Delta \varepsilon \Delta V \). To solve Eq. (4) for the field at general positions \( \mathbf{r} \), we must first solve for the fields at the positions of the flipped pixels in \( V \). Evaluating the field at the position \( \mathbf{r}'' \in V \), we obtain:

\[ E(\mathbf{r}'') = E_0(\mathbf{r}'') + b \sum_{\mathbf{r}'' \in V} G_0(\mathbf{r}', \mathbf{r}'') \cdot E(\mathbf{r}''), \]

which describes a system of \( 3N_V \) linear equations for three-dimensional problems, where \( N_V \) denotes the number of pixels in the flipped region \( V \).

Eq. (5) can be expressed in matrix form as:

\[ [\mathbf{I}_{3N_V} - b \mathbf{G}_{0,V}] \cdot \mathbf{E}_V = \mathbf{E}_{0,V}, \]

where we have defined the length \( 3N_V \) vector as:

\[ \mathbf{E}_V = \begin{bmatrix} E(\mathbf{r}'') \\ \vdots \\ E(\mathbf{r}'' \in V) \end{bmatrix}, \]

and \( \mathbf{E}_{0,V} \) is analogously defined for the fields in the unmodified structure. \( \mathbf{I}_{3N_V} \) is the \( 3N_V \times 3N_V \) identity matrix. Finally, \( \mathbf{G}_{0,V} \) is a \( 3N_V \times 3N_V \) matrix consisting of block matrices \( \mathbf{G}_0 \) associated with the flipped region \( V \):

\[ \mathbf{G}_{0,V} = \begin{bmatrix} G_0(\mathbf{r}', \mathbf{r}'') & \cdots & G_0(\mathbf{r}', \mathbf{r}'' \in V) \\ \vdots & \ddots & \vdots \\ G_0(\mathbf{r}'' \in V, \mathbf{r}'') & \cdots & G_0(\mathbf{r}'' \in V, \mathbf{r}'' \in V) \end{bmatrix}. \]

Therefore, we can solve for the resulting field at all flipped pixel locations \( \mathbf{r}'' \in V \) via Eq. (6) upon inverting the matrix \( \mathbf{I}_{3N_V} - b \mathbf{G}_{0,V} \):

\[ \mathbf{E}_V = [\mathbf{I}_{3N_V} - b \mathbf{G}_{0,V}]^{-1} \cdot \mathbf{E}_{0,V}. \]

We now have all the quantities needed on the right hand side of Eq. (4) for evaluating the resulting field \( \mathbf{E}(\mathbf{r}) \) at any position \( \mathbf{r} \in D \cup \{ \mathbf{r}_{m,1} \} \), including, most importantly, the set of monitor positions \( \{ \mathbf{r}_{m,1} \} \). The change in FOM that results from the proposed structure modification can in turn be evaluated by Eq. (1). This completes the first phase of an optimization step where the “rewards” (i.e., changes in the FOM) for arbitrary modifications to the structure are evaluated. The information can then be used to select an optimal modification in the graph traversal.

Before proceeding further to update the field and Green’s function once a modification is chosen, we comment briefly on relevant shapes and sizes for region \( V \). There are a number of benefits for considering proposed flipped regions that consist of multiple pixel locations.

For example, if there is a minimum feature size that patterned structures must obey due to fabrication constraints, then one way to impose such constraints is by only considering modified regions that satisfy the minimum feature size. Along similar lines, fabricability may require that etched patterns (e.g., in a lithographic process) have a specific or minimum depth. This can also be imposed by considering “pillars” of modified material that are of a given height.

B. Updating fields and Green’s functions

Once a particular structural modification is selected, \( \mathbf{E}_0(\mathbf{r}) \) and \( \mathbf{G}_0(\mathbf{r}, \mathbf{r}') \) must be updated before starting the next optimization step. In the following discussion, we shall use superscripts “old” and “new” to denote the \( \mathbf{E}_0 \) and \( \mathbf{G}_0 \) quantities before and after updating. The updated field \( \mathbf{E}_{0,\text{new}}(\mathbf{r}) \) can be obtained by exactly following the equations in Sec. II A, with the substitutions \( \mathbf{E}_0 \rightarrow \mathbf{E}_{0,\text{old}} \), \( \mathbf{E} \rightarrow \mathbf{E}_{0,\text{new}} \), and \( \mathbf{G}_0 \rightarrow \mathbf{G}_{0,\text{old}} \). The region \( V \) and its constituent pixel locations \( \mathbf{r}'' \) should also be reinterpreted as a selected modified region rather than a proposed modified region.

The update equations for the Green’s function are derived in a similar manner as those for the field. The starting point is now Eq. (2), which after discretizing and substituting \( \mathbf{G}_0 \rightarrow \mathbf{G}_{0,\text{old}} \) and \( \mathbf{G} \rightarrow \mathbf{G}_{0,\text{new}} \), becomes:

\[ \mathbf{G}_{0,\text{new}}(\mathbf{r}, \mathbf{r}') = \mathbf{G}_{0,\text{old}}(\mathbf{r}, \mathbf{r}') + b \sum_{\mathbf{r}'' \in V} \mathbf{G}_{0,\text{old}}(\mathbf{r}, \mathbf{r}'') \cdot \mathbf{G}_{0,\text{new}}(\mathbf{r}'', \mathbf{r}'). \]

We first solve the system of equations for positions \( \mathbf{r} \) equal to the flipped pixels \( \mathbf{r}'' \):

\[ \mathbf{G}_{0,\text{new}}(\mathbf{r}, \mathbf{r}') = \mathbf{G}_{0,\text{old}}(\mathbf{r}, \mathbf{r}') + b \sum_{\mathbf{r}'' \in V} \mathbf{G}_{0,\text{old}}(\mathbf{r}, \mathbf{r}'') \cdot \mathbf{G}_{0,\text{new}}(\mathbf{r}'', \mathbf{r}'). \]

For a fixed \( \mathbf{r}' \), we then have a matrix equation analogous to Eq. (6) for the field:

\[ [\mathbf{I}_{3N_V} - b \mathbf{G}_{0,\text{old}}] \cdot \mathbf{G}_{0,\text{new}}(\mathbf{r}') = \mathbf{G}_{0,\text{old}}(\mathbf{r}'), \]

where \( \mathbf{G}_{0,\text{new}}(\mathbf{r}') \) are \( 3N_V \times 3 \) matrices defined as:

\[ \mathbf{G}_{0,\text{new}}(\mathbf{r}') = \begin{bmatrix} G_{0,\text{new}}(\mathbf{r}', \mathbf{r}'') \\ \vdots \\ G_{0,\text{new}}(\mathbf{r}', \mathbf{r}'' \in V) \end{bmatrix}. \]

The definition of \( \mathbf{G}_{0,\text{old}}(\mathbf{r}') \) is given by Eq. (8). Thus, we have:

\[ \mathbf{G}_{0,\text{new}}(\mathbf{r}') = [\mathbf{I}_{3N_V} - b \mathbf{G}_{0,\text{old}}]^{-1} \cdot \mathbf{G}_{0,\text{old}}(\mathbf{r}'). \]
III. PARALLELIZATION

Consider an optimization problem where the design region $D$ consists of $N = n_x n_y n_z$ pixels, where $n_x$ and $n_y$ are the in-plane sizes, and $n_z$ gives the out-of-plane thickness. Note that we distinguish between sizes in the in-plane and out-of-plane directions because the design region in many applications has a thickness much smaller than the sizes of its lateral dimensions, i.e. $n_z \ll n_x, n_y$.

One example is that of integrated photonic devices where the etch depth\textsuperscript{6,23} is typically much smaller than the design region. Another class of examples are metalenses\textsuperscript{19,23–25} for which the thickness of the lens is also much smaller than the diameter. For clarity of presentation, we will also restrict our attention to the case of a single monitor position $r_m$ which we assume to lie outside of the design region. It is trivial to extend to the case of multiple monitor positions.

The quantities that must be stored are $G(r, r')$ and $E(r)$. Note however that the problematic $\mathcal{O}(N^2)$ memory complexity in the algorithm originates only from the part of $G(r, r')$ corresponding to the pairs of positions $r$ and $r'$ within the design region. It is therefore convenient for accounting purposes to separate the design region only Green’s function $G_{dr}(r, r')$, where $r, r' \in D$, from the monitor point Green’s function $G_m(r) \equiv G(r, r_m)$ for $r \in D$. For consistency, we also separate the design region field $E_{dr}(r)$, where $r \in D$, from the single field value at the monitor position $E_m \equiv E(r_m)$.

Our first task is to partition $G_{dr}(r, r')$ into divisions that fit into the random-access memory (RAM) of a single worker (note: the amount of RAM available per worker is specific to the computational platform used). Because there are two position variables involved, there are potentially many valid partitioning schemes. Our strategy is to chunk the space of the first position variable $r$ while keeping the full size of the second position variable $r'$. Picking this particular strategy keeps the implementation relatively simple while the chunk memory requirements remain tractable for the problems considered in this paper. Arbitrary (more granular) chunking strategies can be used in the case of bigger problems or when the RAM available per worker is scarce. Our strategy is illustrated in Fig. 2 in which the top and bottom blocks, each of size $N$ pixels, respectively represent the spaces of the two position variables. The direct product of these two spaces then represents all position combinations of the full $G_{dr}(r, r')$ with $\mathcal{O}(N^2)$ elements. The colored region represents a single chunk in our strategy. Namely, the space of $r$ is split into chunks of size $M = m_x m_y n_z$ with $m_x \ll n_x$ and $m_y \ll n_y$. We denote the $k$th chunk of $G_{dr}(r, r')$ as $G_{dr}^{(k)}$. The memory scaling of $G_{dr}^{(k)}$ is then reduced to $\mathcal{O}(MN)$, where $M$ can be $\mathcal{O}(1)$ and, in the extreme case, $M = 1$. Note that we did not split in the out-of-plane direction since we assume $n_z$ is already small. This choice will be particularly convenient for the demonstration in Sec. IV for performing multipixel “pil- lar updates” where each proposed structure modification

---

Figure 1. Computational flow chart summarizing the Green’s function pixel-by-pixel optimization scheme.
Evaluating it at the monitor position \( r \) interpreting quantities with respect to a partition, we have all required inputs contained within the chunked \( V \) to G sets of all \( G \) for determining all resulting \( E \) tions are Eqs. (4)-(9). In particular, the inputs needed partitioning convention, the two main computational phases: the calculation of the Green’s function pixel-by-pixel optimization scheme to select based on the \( \Delta FOM \) values. It emits the chosen modification and the process continues until a single modification \( V^* \) originating from partition \( k^* \) remains. This concludes the first phase of the optimization step.

Next, we turn to the update step after the single region \( V^* \) originating from partition \( k^* \) is chosen and flipped. From the discussion in Sec. II B, we must first compute the updated quantities at the flipped pixel positions: 

\[
G_{0,(r,m',r')}^{new} = \left[ \begin{array}{c} G_{0,(r''',r')}^{new} \\ \vdots \end{array} \right], \quad E_{0,(r')}^{new} = \left[ \begin{array}{c} E_{0,(r''',r')}^{new} \\ \vdots \end{array} \right], 
\]

The information needed to compute these quantities is available only in the partition \( k^* \). Therefore, partition \( k^* \) must communicate these quantities to all other partitions. Its data are fetched through a filter operation and then broadcast. The quantities \( \{G_{dr}^{(k)}, G_m^{(k)}, E_{dr}^{(k)}\} \) in each partition \( k \) can then be independently updated via Eqs. (4) and (10).

To summarize the discussion so far, we have proposed a parallelization scheme for the Green’s function based pixel-by-pixel optimization algorithm in which the Green’s function—the leading source of memory complexity—is partitioned into chunks that effectively scale as \( O(N) \) as opposed to \( O(N^2) \). Under this partitioning convention, the two main computational phases of an optimization step can be executed across all the partitions independently without the need for any inter-partition communication. The stage that does require
IV. DEMONSTRATION ON METALENS OPTIMIZATION

We illustrate the scale of problems that can be tackled with our massively parallel scheme by optimizing a high NA focusing metalens. The configuration is illustrated in Fig. 4. Our design region consists of a square slab with a side length of 9 \( \mu \)m and a thickness of 300nm. We take the initial background system to consist entirely of vacuum \((\varepsilon_0 = 1)\) such that the initial Green’s function is known analytically\(^{15}\). An x-polarized plane wave of wavelength \(\lambda = 1500\)nm is incident normally, and the optimization objective is to maximize the intensity of the diffracted light at a focal point located 4.5\( \mu \)m (NA = 0.7) from the design region by patterning the design region with blocks of silicon \((\varepsilon = 12)\) that extend through the thickness of the lens.

Although our primary objective here is to demonstrate the ability of our scheme to scale to large problem sizes, we note that the problem configuration described above might model an in-fiber focusing metalens in which the facet of an optical fiber is patterned to focus the outgoing light\(^{27-30}\). For this particular problem, the most popular optimization method used in the literature is the phase profile matching approach where a slowly varying phase...
The metalens design region consists of a square region with a side length of \( s = 9\mu m \) and a thickness of \( t = 300\text{nm} \) surrounded by vacuum. The goal of the optimization is to pattern the design region with silicon such that the resulting metalens focuses an \( x \)-polarized plane wave light source with wavelength \( \lambda = 1500\text{nm} \). The targeted focal length is \( f = 4.5\mu m \), corresponding to a NA of 0.7.

Change distribution analytically known to effect focusing behavior is modeled by stitching together large, predefined unit cell structures that have only a small number of degrees of freedom—e.g. nanorods several hundreds of nanometers in length, each configured by an orientation angle. This approach has proven to be extremely efficient in the design of large area metasurfaces\(^3\). However, for metalenses with high NA and therefore fast varying phase changes, the phase profile stitching approach breaks down\(^3\). In contrast, the Green’s function approach is naturally free of such constraints. Furthermore, by optimizing directly in the pixel representation rather than with large unit cells, the Green’s function approach allows for a much larger number of design degrees of freedom per unit area, significantly expanding the landscape of potential designs.

In Fig. 5, we plot the evolution of the FOM over the course of an optimization of the metalens system. The FOM is defined to be the ratio of the achieved electric field intensity at the focal point relative to the initial intensity. How-

\[ \text{FOM} = \frac{I_{\text{focal}}}{I_{\text{initial}}} \]

As expected from a fully greedy search strategy, the FOM monotonically increases throughout the optimization until step 750 (labeled point (3) in Fig. 5) when no blocks that would increase FOM can be found.

Computationally, the partitioning of the design space was implemented by chunking the \( 120 \times 120 \) pixel sized lateral design region into \( K = 60 \times 60 = 3600 \) partitions, each of size \( 2 \times 2 \)—i.e. the size of each partition equals the minimum feature size imposed. The computation of the pipeline was distributed across a fixed pool of 2000 workers (each utilizing at most 1 CPU and 1GiB of RAM). The sum of the sizes of the inputs taken over all partitions in each optimization step was on average 220GiB, and each optimization step took on average 3.7 minutes. We note that our optimization scheme also works at higher resolutions, for example at 50nm per pixel, where the memory requirement for the inputs at each step increases to \( \sim 2\text{TiB} \), in accordance with the \( \mathcal{O}(N^2) \) scaling.

The left panels in Fig. 6 respectively show the metalens design at steps 100, 400, and 750 of the optimization (labeled by the star symbols in Fig. 5). Qualitatively, the optimizer tends to place slabs of silicon material oriented along the \( x \) axis. This aligns with physical intuition as the incident plane wave is polarized along \( x \). Larger currents contributed by the polarization, and therefore larger phase changes, can then be induced when the dielectric patterns are oriented along the polarization direction. Validation simulations of the three designs were performed with the finite-difference time-domain (FDTD) method\(^3\), using the open-source software package Meep\(^3\). The middle and right panels in Fig. 6 show the relative intensity distributions of the focal spot: the middle panels show the \( y-z \) plane cross

![Figure 4](image1.png)  
**Figure 4.** An illustration of the configuration of the high NA focusing metalens optimized using our massively parallel Green’s function based pixel-by-pixel optimization scheme. The metalens design region consists of a square region with a side length of \( s = 9\mu m \) and a thickness of \( t = 300\text{nm} \) surrounded by vacuum. The goal of the optimization is to pattern the design region with silicon such that the resulting metalens focuses an \( x \)-polarized plane wave light source with wavelength \( \lambda = 1500\text{nm} \). The targeted focal length is \( f = 4.5\mu m \), corresponding to a NA of 0.7.

![Figure 5](image2.png)  
**Figure 5.** FOM versus optimization step for an optimization of a focusing metalens as described in the main text. The FOM is defined to be the ratio of the achieved electric field intensity at the focal point relative to the initial intensity. The three points (at steps 100, 400, and 750) labeled by star symbols along the trajectory correspond to three designs studied in detail in Fig. 6.
Figure 6. The designs and FDTD validated relative field intensity distributions for (a) step 100, (b) step 400, and (c) step 750 along the optimization trajectory from Fig. 5 (indicated by the locations of the three star symbols, (1), (2), and (3)). For the designs, white represents the locations of patterned silicon pillars, while black represents the background vacuum. The targeted focal length is 4.5µm, which corresponds to NA = 0.7 for a metalens with side length 9µm. Note that a minimum feature size of 150nm is imposed by considering modified regions that are 2 × 2 pixels large in the lateral design region dimensions. For the relative intensity distribution plots, the middle panels show the y-z plane cross section, while the right panels show the x-y focal plane cross section at z = 4.5µm. The metalens is located at z = 0. As desired, the relative intensity at the focal point increases substantially as the optimization progresses.

V. DISCUSSIONS AND CONCLUSIONS

We have already enumerated several inherent advantages of pixel-by-pixel topology optimization, namely, the circumvention of full-wave simulations, binarization by construction of the scheme, and the ease of enforcing minimum feature size constraints. We highlight another feature of the pixel-by-pixel paradigm: it allows for the reframing of the nanophotonic topology optimization problem as a graph traversal, where the optimizer can be considered as an agent selecting among a discrete set of actions (i.e., which pixels to flip) at each optimization step. In other words, the topology optimization can now be naturally reinterpreted as an RL problem.

The policy of the agent utilized in the results presented in Section IV is the simple greedy policy where the maximum positive ΔFOM inducing action is always selected. However, completely greedy policies do not, in general, lead to the best results, especially for loss landscapes as complicated as those in photonic inverse design problems. In the language of RL, effective agents are those that strike the right balance between exploration and exploitation.

To showcase this concept, we performed optimizations of a small scale 21 × 21 pixel sized focusing metalens at a resolution of 75nm per pixel targeting a NA of 0.7, using the ϵ-greedy policy. The ϵ-greedy policy is a sim-
We expect this advantage to be amplified for larger sized problems involving longer optimization trajectories. For $\epsilon = 0.3$, the performance degrades to approximately the level of the greedy strategy, which can be ascribed to an overemphasis on exploration. Our simple example illustrating the effects of including non-greediness hints at the potential of applying techniques from the vast and topical field of RL to the field of nanophotonic inverse design—a connection that becomes all the more apparent and ripe for exploration through the lens of the Green’s function based pixel-by-pixel optimization paradigm.

In conclusion, in this paper we have demonstrated a massively parallel implementation of the Green’s function based pixel-by-pixel optimization scheme—a scheme that would otherwise be severely limited in the scope of applications by the $\mathcal{O}(N^2)$ memory scaling for the storage of the two-point Green’s function. Researchers and designers in industry and academia alike now have access to high performance computing clusters as well as commercial cloud computing platforms. Thus, the method we have proposed elevates the Green’s function based optimization paradigm to a new level of scale and practicality comparable to that of adjoint variable, gradient-based approaches. Furthermore, by reinterpreting the photonic inverse design problem as a graph traversal, immediate connections to the field of RL can be made. In combination with our method for scaling the Green’s function scheme, we believe that opportunities await for tackling large scale photonic inverse design problems with RL inspired techniques. Future work will focus on further developing this connection, as well as extending the Green’s function scheme to optimize for more realistic configurations of the metalens (e.g. inclusion of a substrate layer, imposition of spatial symmetries) and beyond (e.g. integrated photonic devices).

REFERENCES

1. S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A. W. Rodriguez, Nature Photonics 12, 659 (2018).
2. J. Wang, Y. Shi, T. Hughes, Z. Zhao, and S. Fan, Opt. Express 26, 3236 (2018), URL http://www.osapublishing.org/oe/abstract.cfm?URI=oe-26-3-3236.
3. G. Veronis, R. W. Dutton, and S. Fan, Optics letters 29, 2288 (2004).
4. A. M. Hammond, A. Oskooi, S. G. Johnson, and S. E. Ralph, Opt. Express 29, 23916 (2021), URL http://www.osapublishing.org/oe/abstract.cfm?URI=oe-29-15-23916.
5. D. Sell, J. Yang, S. Doshay, R. Yang, and J. A. Fan, Nano letters 17, 3752 (2017).
6. A. Y. Piggott, J. Lu, K. G. Lagoudakis, J. Petykiewicz, T. M. Babinec, and J. Vucković, Nature Photonics 9, 374 (2015).
7. D. Vercruyssse, N. V. Sapra, L. Su, R. Trivedi, and J. Vucković, Scientific reports 9, 8999 (2019).
8. A. Y. Piggott, J. Petykiewicz, L. Su, and J. Vucković, Scientific reports 7, 1 (2017).
9. M. F. Schubert, A. K. C. Cheung, I. A. D. Williamson, A. Spyra, and D. H. Alexander, Inverse design of photonic devices with strict finding fabrication constraints (2022), 2201.12965.
10. D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Luncot, L. Sifre, D. Kumaran, T. Graepel, et al., Science 362, 1140 (2018).
A. Mirhoseini, A. Goldie, M. Yazgan, J. W. Jiang, E. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, A. Nazi, et al., Nature 594, 207 (2021).

B. Shen, P. Wang, R. Polson, and R. Menon, Nature Photonics 9, 378 (2015).

S. Boutami and S. Fan, J. Opt. Soc. Am. B 36, 2378 (2019), URL http://www.osapublishing.org/josab/abstract.cfm?URI=josab-36-9-2378.

O. J. F. Martin, A. Dereux, and C. Girard, J. Opt. Soc. Am. A 11, 1073 (1994), URL http://www.osapublishing.org/josaa/abstract.cfm?URI=josaa-11-3-1073.

O. J. F. Martin and N. B. Piller, Phys. Rev. E 58, 3909 (1998), URL https://link.aps.org/doi/10.1103/PhysRevE.58.3909.

S. Boutami and S. Fan, J. Opt. Soc. Am. B 36, 2387 (2019), URL http://www.osapublishing.org/josab/abstract.cfm?URI=josab-36-9-2387.

S. Boutami, K. Hassan, C. Dupré, L. Baud, and S. Fan, Applied Physics Letters 117, 071104 (2020), https://doi.org/10.1063/5.0013558, URL https://doi.org/10.1063/5.0013558.

J. Dean and S. Ghemawat, in OSDI ’04: Sixth Symposium on Operating System Design and Implementation (San Francisco, CA, 2004), pp. 137–150.

T. Akidau, R. Bradshaw, C. Chambers, S. Chernyak, R. J. Fernández-Moctezuma, R. Lax, S. McVeety, D. Mills, F. Perry, E. Schmidt, et al., Proceedings of the VLDB Endowment 8, 1792 (2015).

A. B. Community, Apache Beam (2021), URL https://github.com/apache/beam.

T. Akidau, R. Bradshaw, C. Chambers, S. Chernyak, R. J. Fernández-Moctezuma, R. Lax, S. McVeety, D. Mills, F. Perry, E. Schmidt, et al., Proceedings of the VLDB Endowment 8, 1792 (2015).

M. Paulus, P. Gay-Balmaz, and O. J. Martin, Physical Review E 62, 5797 (2000).

P. Cheben, R. Halir, J. H. Schmid, H. A. Atwater, and D. R. Smith, Nature 560, 565 (2018).

M. Khorasaninejad, W. T. Chen, R. C. Devlin, J. Oh, A. Y. Zhu, and F. Capasso, Science 352, 1190 (2016).

H. Chang and O. D. Miller, Optics express 28, 6945 (2020).

G. Cloud, Dataflow Google Cloud, URL https://cloud.google.com/dataflow.

J. Yang, I. Ghimire, P. C. Wu, S. Gurung, C. Arndt, D. P. Tsai, and H. W. H. Lee, Nanophotonics 8, 443 (2019), URL https://doi.org/10.1515/nanoph-2018-0204.

M. Kim and S. Kim, Scientific Reports 10, 1 (2020).

M. Principe, M. Consales, A. Micco, A. Crescitelli, G. Castaldi, E. Esposito, V. La Ferrara, A. Cutole, V. Galdi, and A. Cusano, Light: Science & Applications 6, e16226 (2017).

A. Asadollahbaik, S. Thiele, K. Weber, A. Kumar, J. Drozella, F. Sterl, A. M. Herkommer, H. Giessen, and J. Fick, ACS Photonics 7, 88 (2020), https://doi.org/10.1021/acsphotonics.9b01024, URL https://doi.org/10.1021/acsphotonics.9b01024.

Z. Lin and S. G. Johnson, Optics Express 27, 32445 (2019).

A. Taflove, S. C. Hagness, and M. Picket-May, The Electrical Engineering Handbook 3 (2005).

A. F. Oskooi, D. Roundy, M. Ibanescu, P. Bernal, J. D. Joannopoulos, and S. G. Johnson, Computer Physics Communications 181, 687 (2010).