

Proposal for AI-Assisted Simulation of a Metasurface Imaging System

1 Introduction and Objectives

Metasurfaces are ultrathin, planar optical elements composed of sub-wavelength structures (“meta-atoms”) that can manipulate light’s phase, amplitude, and polarization with high precision. By arranging meta-atoms across a surface, one can create flat optical devices – such as **metalenses** and **meta-holograms** – capable of focusing light or forming images in ways traditionally achieved by bulky lenses or diffractive optics [mdpi.com](https://www.mdpi.com). In a metasurface imaging system, each meta-atom imparts a designed phase shift or polarization change to incident light, enabling complex wavefront shaping for high-resolution imaging, beam focusing, or holography.

Objective: This project aims to **design and simulate a metasurface-based imaging system** (for example, a metalens or a meta-hologram) using a combination of multi-physics simulation tools and AI-driven techniques. The goal is to achieve an optimized metasurface that produces the desired imaging (focusing or image projection) with high efficiency and accuracy across the target spectrum. The outcome will include a validated simulation pipeline (potentially released as software) and results suitable for academic publication. Key objectives include:

- Simulating the **spectral response** (transmission, reflection, phase profile) of candidate metasurface designs to ensure they meet imaging requirements (e.g. focusing at a certain wavelength, or multi-wavelength operation).
- Evaluating the **imaging performance** (e.g. point spread function, image formation quality) of the metasurface by modeling how the structured wavefront produces an image.
- **Optimizing the metasurface design** (geometry, arrangement of meta-atoms) for the desired imaging metrics (such as resolution, efficiency, field of view), using AI-assisted exploration to navigate the large design space more efficiently than brute-force parameter sweeps.
- Integrating the workflow into a unified **Python-based pipeline** for automation. This pipeline will leverage existing simulation engines (Gaussian, LAMMPS, Multiwfn, S4, etc. as appropriate) and utilize AI coding assistants (e.g. OpenAI Codex) to streamline code generation, data analysis, and possibly inverse design.

By combining physics-based simulations with AI tools, we aim to demonstrate a modern “AI for Science” approach: using machine learning and automation to accelerate metasurface design, which is a cutting-edge paradigm highlighted by recent research [mdpi.com](https://www.mdpi.com) [mdpi.com](https://www.mdpi.com).

2 Background and Significance

Metasurface Imaging: Traditional refractive optics are constrained by curvature and thickness, whereas metasurfaces achieve similar (or more complex) functionality on a flat surface. For instance,

a *metallens* uses thousands of nanostructures to focus light, and a *metasurface hologram* can project arbitrary images. These devices have applications in miniaturized imaging systems, AR/VR displays, microscopy, and sensing. However, designing an effective metasurface is challenging because the behavior of each nano-structure and their collective interference must be precisely tuned.

Simulation Challenges: Conventional design of metasurfaces often relies on *parameter sweeps* (iterating over geometric parameters of meta-atoms) or simplified *analytical models*. Parameter sweeps are **computationally intensive** and become impractical for high-dimensional design spaces [mdpi.com](#) [mdpi.com](#). Analytical models (based on e.g. phase approximation or effective medium theory) can provide insight but often **lack accuracy** for complex structures and must be re-derived for each new scenario [mdpi.com](#). Consequently, researchers have begun adopting AI and advanced simulation to handle the complexity.

AI for Metasurface Design: Modern approaches integrate machine learning to *accelerate the design cycle*. Deep learning can learn the relationship between metasurface structure and its optical response, enabling **inverse design** (finding structures that yield a desired output) much faster than brute force search [mdpi.com](#). The integration of AI makes metasurface design **more rapid and precise**, simplifying the previously complex trial-and-error process [mdpi.com](#). For example, neural networks have been used to generate meta-atom geometries for a target phase profile, and evolutionary algorithms have optimized arrangements for holography [mdpi.com](#) [mdpi.com](#).

Multi-physics Considerations: Metasurface performance can also depend on material properties (refractive index dispersion, loss) and fabrication constraints. In some cases, *multi-scale simulation* is beneficial – for instance, using **quantum chemistry** tools to model novel optical materials or coatings on meta-atoms (e.g. molecular films, 2D materials), or using **molecular dynamics** to simulate mechanical stability (e.g. if the metasurface is flexible or uses colloidal nanoparticle assembly). This motivates a workflow that can incorporate **Gaussian** (for electronic structure and material optical properties), **LAMMPS** (for atomistic/mesoscale structural simulation), and **Multiwfn** (for wavefunction analysis), in addition to dedicated electromagnetic solvers.

Simulation Tools: The core optical simulation in this project will use a Maxwell solver suitable for nano-structured surfaces. One powerful option is **S4 (Stanford Stratified Structure Solver)** – an RCWA (Rigorous Coupled-Wave Analysis) code that computes electromagnetic fields in layered periodic structures [web.stanford.edu](#). S4 can efficiently yield **transmission, reflection, and absorption spectra** of a metasurface, as well as near-field distributions [web.stanford.edu](#). Notably, S4’s frequency-domain approach produces very smooth spectra, which is advantageous for gradient-based or global optimization of designs [web.stanford.edu](#). We will integrate S4’s Python API to automate simulations. Other tools will be used as needed: for example, **Meep (FDTD)** could validate broadband or aperiodic effects, and **COMSOL Multiphysics** (or Lumerical FDTD) might be considered for full-device simulations, though we aim to rely on open-source or in-house code when possible.

By combining these tools in a single workflow and harnessing AI for code generation and optimization, we align with emerging best-practices in computational optics. In fact, recent work has demonstrated that an entire metasurface design process – from meta-atom library construction to full device analysis – can be performed in a Python environment with automated scripts [github.com](#). We will follow a similar strategy, ensuring our approach is **efficient, flexible, and reproducible** within a scriptable framework [github.com](#).

3 Methodology Overview

To achieve the project goals, we propose the following integrated methodology:

- **3.1 Design Specification:** Define the target function of the metasurface imaging system. For example, *focusing* a collimated beam to a focal spot at a given distance (metalens behavior), or *projecting an image* at a certain plane (hologram), or perhaps a *spectral imaging element* (dispersive lens). The specifications include operating wavelength(s) or band, polarization state, aperture size, desired resolution, and efficiency. A clear specification guides the simulation setup (e.g. periodic cell size, number of meta-atoms, required phase profile).
- **3.2 Initial Analytical Design:** Begin with known design formulas to get a starting point. For a lens, this could mean calculating the ideal phase profile (e.g. hyperboloidal phase for focusing). For a hologram or complex image, use algorithms like Gerchberg–Saxton to obtain a phase mask that produces the desired image github.com. This gives us a target phase/amplitude map that the metasurface should implement. At this stage we decide the *type of meta-atom* (e.g. pillars, cross nanoantennas, dielectric vs metallic) and *substrate configuration*, informed by literature (for instance, dielectric TiO₂ nanopillars are common for visible meta-optics github.com).
- **3.3 Unit Cell Electromagnetic Simulation (using S4 or FDTD):** With a chosen meta-atom geometry parameterization (e.g. pillar diameter, height, periodicity), perform simulations to create a “**meta-atom library**”. This entails sweeping geometry parameters and computing each structure’s transmission amplitude and phase at the operating frequency github.com. Using S4’s RCWA, we can simulate a single unit cell with periodic boundary (assuming the metasurface locally periodic) and extract its phase delay and efficiency. The result is a set of design curves – for example, pillar radius vs. phase shift – from which discrete candidates (e.g. radii that yield phase shifts covering 0 to 2π) are chosen. *Note:* S4’s Python extension will be used to script these simulations; an AI coding assistant can help write these scripts by generating the Lua/Python code needed for each run. For instance, we will prompt Codex to generate a parameter-sweep loop that calls S4’s simulation object for varying geometry, and parse the output to record phase response. This automates what would otherwise be a tedious manual process.
- **3.4 Full Metasurface Construction:** Using the library of meta-atoms (with known phase responses), we map them onto the aperture according to the target phase profile. For a lens, this means placing meta-atoms that impart the correct phase delay at each radial position. For a hologram, the meta-atoms are placed to match the hologram phase mask. We will create a layout file (e.g. GDSII) that can represent the design – this step can leverage an existing Python tool or our own script (the GitHub repository’s *layout* module demonstrates generating GDS files of metasurface patterns github.com github.com). At this stage, we have a *design file* and can proceed to simulate its performance.
- **3.5 Imaging Simulation:** Simulating the *entire metasurface* directly with full-wave FDTD or RCWA can be computationally heavy, especially if the device is large and non-periodic. We will employ two complementary approaches:
 - (a) **Fourier Optics Simulation:** Using the calculated phase/amplitude distribution of the metasurface, we perform a wave propagation (e.g. Fresnel or angular spectrum method) to compute the field at the image plane. This will yield the expected focal spot size or reconstructed image. We can implement this in Python (NumPy/Scipy for FFTs) – essentially treating the metasurface as a phase mask. This approach efficiently predicts imaging performance (point spread function, focal length achieved, etc.) without simulating every nanostructure in full detail.

- (b) **Full-Wave Validation:** For more accuracy, we can simulate a portion of the metasurface or use periodic approximations. Using S4, we might simulate the central unit cells with an appropriate phase gradient to estimate the focus (though S4 inherently assumes periodicity – one trick is to use a supercell approximation for a portion of the lens). Alternatively, a small-scale FDTD (using Meep or Lumerical) can model a subset of the metasurface to validate phenomena like coupling between elements or aberrations. We will use such full-wave simulations sparingly (to manage computation time), focusing on critical regions or worst-case scenarios.
- **3.6 Material and Multi-Scale Modeling (optional):** If the metasurface involves novel materials or if the design hinges on material dispersion, we will incorporate additional simulations:
 - **DFT/Quantum Calculations (Gaussian/Quantum Espresso):** To obtain optical constants or molecular resonance data, we can simulate materials at the atomic scale. For instance, if exploring a metasurface made of a new **nonlinear optical polymer or molecular film**, Gaussian 16 (with TD-DFT) could compute its refractive index spectrum or polarizability. Multiwfn can analyze the resulting electron density or transition dipoles frontiersin.org to relate molecular structure to optical absorption. These properties inform our electromagnetic model (e.g. by providing input to S4 for refractive index dispersion files github.com).
 - **Molecular Dynamics (LAMMPS):** If our metasurface design involves **mechanical flexibility or thermal effects** (for example, a reconfigurable metasurface or a soft polymer substrate), we will use MD simulations to test structural stability. LAMMPS can simulate bending of a metasurface film or the self-assembly of colloidal meta-atoms on a surface. This helps ensure the design can be fabricated and operated under real conditions (e.g. does not collapse, can sustain thermal loads). Such multi-physics coupling (as also suggested in recent research mdpi.com) extends the realism of our simulations.
- **3.7 AI-Assisted Optimization:** Designing a high-performance metasurface often requires fine-tuning many parameters (geometry sizes, orientations, spacing arrangements). Rather than relying solely on manual or grid search (which, as noted, is impractical for large spaces mdpi.com mdpi.com), we will integrate **AI-driven optimization**:
 - **Surrogate Modeling:** We plan to train a simple ML model (e.g. a neural network or Gaussian Process) on the simulation data (from unit-cell sweeps and a few full-device simulations). The model will learn the mapping from design parameters to performance metrics (efficiency, focal error, etc.) mdpi.com. This surrogate can then be queried or used in an optimizer to propose improved designs in a fraction of the time of full simulations.
 - **Inverse Design via Neural Networks:** For more complex objectives (like hologram image fidelity), we may employ deep learning as in recent literature – e.g. using a generative model or an evolutionary strategy guided by a neural network to evolve the metasurface structure. The deep learning’s ability to capture hidden relationships can significantly speed up finding a good solution mdpi.com. (We will be mindful of ensuring the physical realism of any AI-proposed designs.)
 - **AI Coding Assistant for Automation:** Throughout this process, we will use an AI coding tool (such as OpenAI Codex or ChatGPT) to automate scripting and data processing. For example, Codex can be prompted to write a Python function that reads S4 output and calculates merit functions (like imaging resolution from a simulated focal

spot). It can also assist in writing optimization loops or even in setting up neural network training code (using frameworks like TensorFlow/PyTorch). This not only accelerates development but ensures our workflow is reproducible – the AI can help document and standardize the code. Importantly, by keeping the human in the loop (we will verify all AI-generated code), we maintain scientific validity while benefiting from accelerated coding.

- **3.8 Validation and Analysis:** Once an optimized design is obtained, we will perform thorough validation:
 - **Broadband and Off-axis Performance:** Simulate the metasurface at wavelengths across the desired band to ensure acceptable performance (e.g. focus quality) over the spectrum. For imaging systems, also test varying incident angles or polarization to evaluate robustness.
 - **Comparison with Diffraction Limit or Theoretical Models:** Analyze the achieved imaging resolution vs. the theoretical diffraction limit for the aperture, and quantify any aberrations. If the metasurface is a lens, compute its focal length and compare with design; if a hologram, compute the fidelity (e.g. structural similarity index) between the reconstructed and target images.
 - **Efficiency and Tolerance:** Calculate the optical efficiency (how much of the incident light is utilized for imaging) and examine the effect of small perturbations (manufacturing tolerances like dimension errors of meta-atoms). This step is crucial for a publication, demonstrating the design’s viability. Tools like S4 can compute these by adjusting parameters, and an AI script can sweep tolerance scenarios automatically.
- **3.9 Iteration:** Based on validation results, iterate on the design if needed. The pipeline allows adjusting the meta-atom library or the arrangement algorithm and quickly re-simulating. This iterative loop continues until the design meets all targets (or we reach the practical limits where trade-offs must be accepted).
- **3.10 Implementation Plan and Timeline:** We anticipate the project to proceed in phases:
 - **Phase 1: Literature Review & Problem Definition (Week 1–2).** Gather state-of-art on metasurface imaging designs (e.g. latest metalenses, holograms) and finalize specifications. Set up software environment (install S4, Gaussian, etc., and ensure Python integration works). *Deliverable:* A review summary and chosen design targets.
 - **Phase 2: Initial Design & Simulation Setup (Week 3–5).** Develop the target phase profile and initial meta-atom geometry concept. Use AI assistance to script unit-cell simulations in S4; run parameter sweeps to create the meta-atom library. Also, if needed, run Gaussian for material properties. *Deliverable:* Meta-atom performance curves; selection of candidates covering $0-2\pi$ phase.
 - **Phase 3: Assemble Metasurface & Preliminary Imaging Test (Week 6–7).** Map meta-atoms to form the full metasurface (write code for layout). Perform Fourier-optics simulation to check focusing or image formation. If issues (e.g. phase quantization errors causing aberration), adjust design (increase library density or refine phase profile). *Deliverable:* Simulated image or focal spot from the initial design, and analysis of aberrations/efficiency.
 - **Phase 4: Optimization Loop (Week 8–10).** Integrate an optimization approach. Use surrogate modeling or gradient-free optimization (like differential evolution) with

our simulation-in-the-loop. Leverage AI to speed up coding this optimization. Iterate to improve performance metrics (maximize Strehl ratio for a lens, or image contrast for a hologram, etc.). *Deliverable:* An optimized set of design parameters and corresponding simulation results.

- **Phase 5: Detailed Multi-Physics Analysis (Week 11–12).** Use MD to simulate any mechanical aspects if relevant (or skip if purely optical and solid-state). Perhaps simulate thermal effects if the device might heat under high power. Use these to ensure the design is robust. *Deliverable:* Data on mechanical/thermal stability (or a statement of not needed if design is purely passive dielectric).
- **Phase 6: Final Validation & Documentation (Week 13–14).** Run final full-spectrum simulations, compile all results. Prepare figures (spectral plots, field distributions, imaging results). Document the methods, and package the simulation code into a pipeline (with instructions or notebooks) for reproducibility. *Deliverable:* Draft of a research paper and a functional software pipeline (likely as Jupyter notebooks or Python scripts) that can be shared.

This timeline is tentative and can be adjusted as needed. The overall plan is comprehensive but each step is manageable with the assistance of automation and existing frameworks. By the end, we expect to have a **working metasurface design** verified in simulation, an **AI-augmented design workflow**, and extensive data to support a publication in a photonics or computational physics journal.

4 Technical Approach and Tool Integration

To ensure clarity, here we detail **how specific tools will be used and integrated** in the workflow:

- **Python as the Orchestrator:** All tasks will be orchestrated through Python, using it as a “glue” to connect various simulation softwares. This approach has been proven effective in prior metasurface design work github.com. We will write (and partly auto-generate) Python scripts/notebooks that call external tools, manage data flow, and perform analyses. For example, Python can call S4 via its Python API, invoke Gaussian via subprocess (or use PySCF as a Python alternative for quantum calc), and run LAMMPS via input scripts. The use of Jupyter notebooks will facilitate mixing code, results, and documentation for iterative development.
- **Electromagnetic Solver – S4:** We will install S4 (which offers a Python extension) web.stanford.edu. In Python, we’ll use S4’s `Simulation` object to define layer stacks, material properties, and Fourier orders. For a given meta-atom geometry, the script will set up the structure (e.g. a cylindrical post of certain diameter and height in a unit cell, on a substrate), specify source illumination (plane wave, relevant polarization, at target wavelength), and then compute the S-parameters. From these, transmission amplitude and phase are obtained web.stanford.edu. This process will run in a loop for different geometry parameters. We will ensure that the simulation settings (e.g. Fourier harmonics count, mesh, convergence criteria) are sufficient for accuracy. *Note:* If S4’s assumptions (periodicity, uniform infinite array) do not hold for some aspect, we will cross-check with FDTD (using **Meep**, an open-source FDTD, which can also be scripted in Python). Meep is particularly useful if we want to simulate a finite array or an aperiodic arrangement; however, it is slower than S4 for periodic structures, so we will use it selectively.

- **Quantum Chemistry – Gaussian & Multiwfn:** If our design requires material properties that are not readily available, we will use **Gaussian 16/16** for DFT calculations. For instance, if exploring a metasurface made of a new **nonlinear optical polymer or molecular film**, Gaussian can compute the molecular refractive index or absorption spectrum (via time-dependent DFT for excited states). We might calculate the dielectric function at the wavelength of interest. **Multiwfn** will assist in analyzing Gaussian outputs (e.g. obtaining refractive index from the polarizability tensor, or visualizing electron density to ensure the material’s bandgap is appropriate for low loss). These results feed back into the EM simulation (we can input a custom refractive index dispersion into S4 or Meep). This quantum step is optional and dependent on the materials in our metasurface; if we stick to well-characterized materials (like TiO₂, silicon, gold, etc.), we will rely on literature data (e.g. from RefractiveIndex.info database, as the GitHub repository did github.com).
- **Molecular Dynamics – LAMMPS:** For any structural simulation, we will set up LAMMPS with appropriate force fields. One potential use-case: simulate a **self-assembly process** of nano-particles into a metasurface pattern (if we pursue a bottom-up fabrication concept). Another use-case: test the **mechanical integrity** of a nanopillar array under strain – e.g. apply force in LAMMPS to see if pillars buckle at a certain aspect ratio. LAMMPS will be run with input scripts (we can generate these with Python or AI assistance). The outputs (trajectories, energies) will be analyzed to deduce if any mechanical constraints must be applied to our design (for example, maximum height of a pillar for stability). Additionally, if the metasurface will operate with power that causes heating, we might simulate a nano-unit cell in LAMMPS to see how heat distributes – or use COMSOL’s multi-physics for a continuum thermal analysis (the MDPI review highlights such multi-physics co-design for thermal effects [mdpi.com](https://www.mdpi.com)).
- **AI Integration:** The AI’s role will primarily be to *accelerate coding tasks* and assist with optimization:
 - *Code Generation:* We will use Codex/ChatGPT to write repetitive pieces of code, such as the S4 simulation loop, data parsing functions, or even LaTeX code for our paper. For example, after manually doing one S4 simulation, we can prompt the AI to generate a generalized script for all geometries. This reduces human error and speeds up development. According to the user’s request, we ensure all generated code is feasible and testable; we won’t rely on AI for decisions without verification.
 - *Data Analysis:* AI can also help in making sense of large data. If we have thousands of simulation results, we might use it to quickly draft visualization code (matplotlib plots of phase vs frequency, etc.) and even to suggest patterns (“the AI might observe that increasing pillar diameter yields a nearly linear phase shift until resonance, which matches theory”). While we will do rigorous analysis ourselves, such insights can be useful for writing the discussion in the paper.
 - *Design Suggestions:* As a creative tool, we could query an AI model for design ideas (“How might one reduce chromatic aberration in a metalens?”) to ensure we’re not missing known techniques. This is more speculative, but given the broad knowledge base of models like ChatGPT, it could point us to relevant strategies (like dispersion engineering or multi-layer metasurfaces). Any suggestions would be cross-checked with literature.
- **Validation & Results Processing:** We will maintain a structured approach to store simulation results (possibly using Pandas dataframes or NumPy arrays saved to file). This

makes it easier to apply analysis scripts across different design iterations. We’ll produce plots for: the phase coverage of our meta-atom library, the efficiency spectrum of the metasurface, the point spread function at focus, and possibly a comparison image for hologram target vs reconstructed. For publishing, we will ensure to cite comparison with other works: for instance, if our metalens focal efficiency is 80% at 532 nm, we’ll note how it stands relative to recent metalenses in literature. Our pipeline will also allow *reproducing* these results by running the notebooks – an advantage of the fully scripted approach github.com.

5 Expected Outcomes and Impact

By the completion of this project, we anticipate the following outcomes:

- **Validated Metasurface Design:** A specific metasurface (with defined geometry and material) that achieves the desired imaging function (e.g., a flat lens focusing at a certain distance, or a hologram projecting a given pattern) with high performance. We expect to demonstrate quantitatively the imaging quality (e.g., diffraction-limited focal spot, or high-fidelity holographic reconstruction) and efficiency of the device. This will be backed by simulation data and could be used as a blueprint for experimental fabrication in future work.
- **Comprehensive Simulation Pipeline:** A **Python-based software pipeline** that integrates electromagnetic simulation, data analysis, and optimization. The pipeline will be well-documented, possibly open-sourced, so that other researchers can reproduce or build upon it. The use of Python notebooks and AI-assisted code ensures the workflow is transparent and modular. For example, one could swap out S4 for another solver, or plug in a different optimization algorithm, with minimal changes. This pipeline will be a valuable tool for continued research in metasurface design and can be extended to other applications (e.g., meta-antennas for antennas, or photonic crystal design), aligning with the idea of AI-accelerated scientific computing.
- **Publication and Reports:** The results will be compiled into a research paper or report, detailing the methodology and findings. We will include: the background and need for AI in metasurfaces (citing works that highlight how AI improves design efficiency), the technical approach (with any novel integration of tools described), and the performance of our designed metasurface. Given the novelty of combining *quantum material simulation*, *electromagnetic design*, and *AI optimization* in one study, this work could be suitable for a photonics or computational physics journal. We will also emphasize the “AI for Science” angle, noting that our approach exemplifies how recent AI advancements (like large language model-based coding assistants) can tangibly accelerate research – a timely insight following the recognition of AI’s role in science (e.g. the mention of 2024 Nobel prizes related to AI for Science in the prompt context).
- **Potential Discoveries:** During optimization, the AI might discover non-intuitive design solutions (for instance, a pattern of meta-atom arrangement that yields an improved off-axis performance). Documenting these could provide new understanding or design principles for metasurfaces. At minimum, we’ll analyze the optimized design to extract human-understandable insights (e.g., “the algorithm favored slightly elliptically-shaped pillars at the lens periphery to correct astigmatism”), turning black-box optimization results into domain knowledge.

Impact: Successfully completing this project will demonstrate a cutting-edge approach to photonic design, combining physics-based modeling with AI. It will provide a template for tackling other

complex simulation-driven design problems (not just in optics, but in materials science, nanotechnology, etc.) by leveraging automation and machine learning. Moreover, the specific metasurface imaging system designed could have practical implications in imaging technology – for example, a high-efficiency flat lens for microscopy or a compact holographic projector for augmented reality. By publishing our data and code, we contribute to the open science movement, allowing others to verify and utilize our results github.com.

Finally, this project aligns with the broader trend of **AI-augmented scientific research**. As noted in recent reviews, deep learning and AI can drastically reduce the cost and time of exploring complex design spaces in photonics mdpi.com. Our work will stand as a case study of this paradigm shift, potentially inspiring further adoption of AI tools (like Codex) in simulation workflows across different scientific disciplines.

6 Conclusion

In summary, this proposal outlines a comprehensive plan to design and simulate a metasurface imaging system by integrating advanced simulation tools (Gaussian, LAMMPS, Multiwfn, S4) within a Python-driven, AI-assisted framework. We will tackle the multi-faceted challenge – spanning electromagnetic wave propagation, materials properties, and optimization – by leveraging each tool’s strengths: S4 for efficient nanophotonic simulation web.stanford.edu, quantum calculations for material insights, and AI for intelligent automation. The structured workflow, from initial analytical design to final optimization and validation, ensures that every aspect of the metasurface’s performance is accounted for. Moreover, by using AI assistants for coding and learning from data, we greatly accelerate the research process, overcoming the limitations of brute-force methods mdpi.com mdpi.com.

The deliverables will include both a high-performance metasurface design (ready for potential experimental realization) and a re-usable pipeline for future **AI-for-Science simulation projects**. This project not only aims to produce a specific scientific result (a novel metasurface design) but also to demonstrate a methodology that can be generalized to other problems where simulation and AI intersect. The convergence of computational physics tools and AI reflects the new paradigm in scientific research, and this work will place us at the forefront of that movement.

By executing this plan, we expect to contribute valuable knowledge and resources to the photonics and materials community, potentially enabling more rapid innovation in metasurface-based imaging devices. We look forward to carrying out this project and disseminating the findings through both publication and open-source software, thereby adding to the collective progress in **AI-empowered scientific discovery**.

Sources Cited

- Liu, V. et al. “S4: Stanford Stratified Structure Solver” – Documentation on the RCWA tool S4 and its capabilities web.stanford.edu web.stanford.edu.
- Hzzg0727 (GitHub user). “Metasurface-Design Workflow (Python/Lumerical FDTD)” – Demonstrates a Python-based metasurface design process and mentions alternatives like Meep and S4 github.com github.com.
- Hou, Z. et al. “Advances in Deep Learning-Driven Metasurface Design and Holographic Imaging”, Photonics 2023 – Review article describing traditional vs. AI-based metasurface

design approaches [mdpi.com](#) [mdpi.com](#) and the role of deep learning in accelerating metasurface hologram design [mdpi.com](#).

- Jiang, J. et al. “Metasurface holographic imaging” (Nat. Photonics 2021) – Explains how metasurfaces control amplitude, phase, polarization for imaging applications [mdpi.com](#). (Additional references within text)