AutoRF: Towards an Agentic Framework for Automated RF Hardware Design

Ruichun Ma
Microsoft Research Asia
ruichunma@microsoft.com

Lili Qiu
Microsoft Research Asia, UT Austin
lili@cs.utexas.edu

Wenjun Hu
Yale University
wenjun.hu@gmail.com

Jiazhao Wang
SUTD
jiazhao_wang@mymail.sutd.edu.sg

Yiwen Song
CMU
yiwens2@andrew.cmu.edu

Hao Pan
Shanghai Jiao Tong University
panh09@sjtu.edu.cn

Abstract

RF hardware design is a complicated, time-consuming, and expertise-bound process, which constrains the development and adoption of hardware innovation. Manual workflows do not scale to emerging wireless applications, while existing design generation strategies, e.g., learning-based approaches, lack training efficiency and generalizability across hardware types, frequency bands, operation modes, and substrate materials. In this paper, we present AutoRF, the first agentic framework for automated RF hardware design, supporting metasurfaces and antennas. It allows users to specify their demands and generate corresponding designs. We introduce extensible design abstractions to enable a modular framework. The core of the framework is an efficient and generalizable algorithm for design search and optimization, which utilizes LLMs to drive both circuit model simulator and EM simulator. To boost reliability and performance, we propose custom programming interfaces and a rule reviewer agent as feedback sources to train a specialized LLM. Evaluation demonstrates high success rate and significant optimization speedup; case studies with fabricated metasurface and antennas, ranging from 2.4 GHz to sub-THz, illustrate an ability to derive novel designs for next-generation wireless infrastructure.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**; • **Hardware** → **Wireless devices**; • **Applied computing** → **Computer-aided design**.

Keywords

Wireless Systems, RF Hardware, Metasurfaces, Large Language Models, Agentic AI

ACM Reference Format:

Ruichun Ma, Lili Qiu, Wenjun Hu, Jiazhao Wang, Yiwen Song, and Hao Pan. 2026. AutoRF: Towards an Agentic Framework for Automated RF Hardware Design. In . ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Hardware development is widely regarded as challenging and time-consuming. In particular, electromagnetic (EM) hardware design, such as antennas and metasurfaces, is inherently complex, requiring a deep understanding of physics, mathematics, and the intricate RF design space. Experts acquire design knowledge from textbooks and practical experience, both requiring significant time and learning effort. Even then, any new design project still requires extensive iterations.

The development of emerging RF hardware, particularly metasurfaces, is especially hurdled by this issue. Composed of sub-wavelength 2D surface element arrays, metasurfaces offer novel physical functionalities to control fundamental properties of passing wireless signals, such as amplitude, phase, and polarization. Surfaces have been identified as a strong candidate for 6G [7, 27, 34, 60, 81]. They are critical hardware enablers to boost wireless networking by altering channel conditions [14, 23, 41, 45, 51, 70, 74] and unlock numerous new applications in sensing, powering, security [22, 61, 62, 64, 71, 78] (subsection 2.1). Since hardware primitives are the foundations for system capabilities, prior work often feature new metasurface prototypes, custom-designed for specific applications through laborious manual efforts. This approach is not only resource-intensive but also difficult to scale across diverse scenarios. More critically, it depends on expert knowledge, making it challenging and inaccessible for most researchers, network engineers, or application developers to customize designs.

There have been efforts to automate the design of RF hardware. State-of-the-art approaches leverage deep learning to generate metasurface patterns or achieve RF inverse design [10, 11, 36, 59]. However, these proposals are limited in data efficiency, generalizability, and interpretability. They require experts to first collect massive datasets for model training, which also inherently limits their design capability to the specific design types within the dataset. (subsection 2.3)

Can we approach human-level efficiency and generalizability in RF hardware design automation? This means creating a system that is capable of learning from limited examples, generalizable across different design types, efficient at design search and optimization with the aid of tools, and flexible to accommodate practical constraints.

In this paper, we present AutoRF, the first framework for automated RF hardware design leveraging Large language models (LLMs). This marks a step towards LLM reasoning-based hardware design process. We start with metasurface design, which is a critical

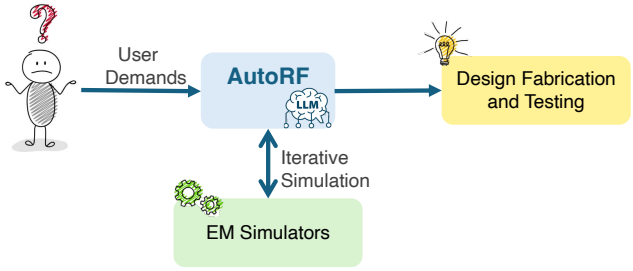


Figure 1: LLM-driven wireless hardware design workflow.

and challenging domain for next-generation wireless networking and sensing. As shown in Figure 1, AutoRF takes in user demands, selects a template and edits the design iteratively using simulators until meeting the target specifications. Users with or without expert knowledge can specify demands and generate the design without manual intervention.

We leverage LLMs' powerful few-shot learning capability [16] and reasoning capability [50], to achieve a generalizable framework. AutoRF supports metasurface designs with operating frequency bands from 900MHz to 160GHz, different substrate materials, one or multiple substrate layers, phase and amplitude control, transmissive and reflective surfaces, programmable (active) and passive surfaces. The operating frequency band affects the application scenarios, substrate materials determine the fabrication process and hardware cost, while the other specifications decide the operation mode of metasurfaces. Supporting all these design types with a few representative templates shows the few-shot generalizability of our approach. AutoRF also facilitates antenna design, highlighting its potential for broader wireless hardware designs. In contrast, existing approaches [36, 79] need over 100K training designs crafted by experts to cover a more limited design space (Table 1).

The design of AutoRF addresses three key challenges (section 3); directly applying LLMs does not achieve our goal (Appendix Figure 13). (i) **Task complexity**: we lack system support to decompose the task and systematically harness the knowledge and reasoning capabilities of LLMs. There is a significant gap between the multi-stage task of RF design and the typical question answering of LLMs. To bridge this gap, we introduce a set of layered abstractions (Figure 2) that capture the design process. Then, we "divide and conquer" the task with LLM-based agentic modules, together with custom tools, to automate the end-to-end process (Figure 3). (ii) **Simulation efficiency**: Brute-force design search and optimization are prohibitively expensive and sometimes infeasible. We present a two-level search and optimization algorithm to combine intelligence with efficiency. We propose a novel LLM-driven circuit model simulator to perform coarse-grained exploration, using lightweight matrix calculation in place of heavyweight EM simulation, to identify promising designs and initialize key parameters. Then, we conduct fine-grained parameter optimization, leveraging LLMs' domain knowledge as optimization priors, to substantially reduce the number of simulation iterations. (iii) **Reliability**: LLMs are probabilistic by nature and may make mistakes or hallucinate, especially when operating the EM simulation. To solve this, we develop task-specific feedback mechanisms, including custom programming interfaces and a design rule reviewer agent, both of which catch and report errors. We utilize these to reject erroneous output, i.e., rejection sampling, and then train a reliable, specialized LLM with filtered trajectories.

We implement AutoRF as a modular agentic framework (Figure 3). We also develop an LLM-based circuit model simulator and an interface to an EM simulator (HFSS [1]), a design library for template designs, and a database for design rules and constraints. The modular architecture of AutoRF enables flexible customization. Users can, for instance, extend the rule database with new requirements, or use particular template designs they prefer, effectively steering the design process while still benefiting from automated search and optimization.

We evaluate AutoRF extensively with both simulated and fabricated RF hardware in Section 5. For simulation, we achieve an end-to-end design success rate of 92% on a diverse testing set of 100 metasurface user demand samples. Similarly, we achieve a success rate of 93% for antenna designs. We further verify the effectiveness of our design search and optimization algorithm and observe over 10× speedup compared to gradient-based variable optimizer of HFSS. For real-world experiments, we fabricated and tested prototypes of 7 metasurface designs and 2 antennas as case studies. Besides a great match between simulated and measured performance, our case studies show that AutoRF can adapt metasurfaces to suit a variety of demands and proposes novel metasurface designs, including a programmable metasurface for continuous reflection phase control and the first paper-based transmissive mmWave metasurface with high power efficiency.

To summarize, we make the following contributions:

- We present the first agentic framework towards RF hardware design generation, with a current focus on metasurfaces and antennas. It pioneers bringing LLM-based reasoning into RF design by introducing extensible abstractions that decompose the end-to-end workflow.
- We present an efficient and generalizable RF design search and optimization algorithm, which utilizes LLM background knowledge as priors to guide optimization and drive tailored simulator tools. It supports various hardware types, across frequency bands, operation modes, substrate materials.
- We propose novel feedback mechanisms, including custom programming interfaces and a design rule reviewer agent, to provide feedback for training a reliable specialized language model.
- We evaluate AutoRF extensively with 150 design tasks via simulation and 9 fabricated designs via real-world experiments, showing high design success rates and the ability to create novel designs. We release code and hardware designs at https://github.com/microsoft/AutoRF.

2 Background and Motivation

2.1 Metasurfaces

Metasurfaces [24, 38, 52, 54] are artificial two-dimensional (2D) materials, constructed as an array of sub-wavelength *elements* or *unit cells*. Each surface element is composed of metallic patterns on substrates, which capture and alter the physical properties of passing electromagnetic waves, such as amplitude, phase, or polarization. With an array of elements, metasurfaces can control the wireless signal propagation [40, 70], such as beam steering or focusing. Also

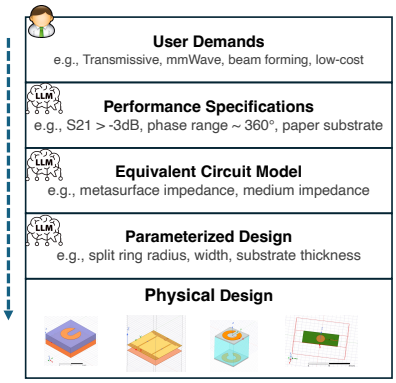


Figure 2: Hardware design abstraction layers. AutoRF converts high-level user demands to detailed designs.

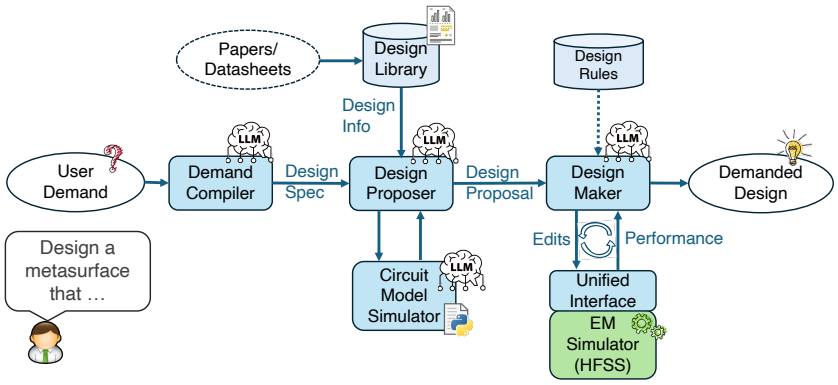


Figure 3: System overview. AutoRF automates the end-to-end workflow from user demand to corresponding design files, with multiple LLM-based modules forming an efficient design search and optimization framework. (Oval components mark the input and output data, while AutoRF modules are marked in blue.)

termed *reconfigurable intelligent surfaces* (RISs) [25, 26], surfaces have drawn substantial industry interest for 6G infrastructure, featuring in white papers [5, 8, 68], standardization efforts [6, 7], and industry testbeds (e.g., [3, 4, 47]). Recent research has explored surfaces to boost wireless connectivity [14, 21, 28, 29, 40–42, 44, 74] and signal-based emerging applications, such as sensing [20, 75, 76, 78], security protection [39, 62], wireless powering/charging [9, 22, 64, 80] and more. Most of these works propose custom surface hardware designs to suit their application scenarios.

Different types of metasurfaces. Metasurfaces can be categorized based on certain attributes of the hardware specifications in Table 2, such as transmissive or reflective surfaces, passive or programmable surfaces. A passive surface has one or multiple layers of metallic patterns with fixed configurations, i.e., one-time programmable before fabrication. A programmable (active) surface has additional circuit components, including varactor or PIN diodes or RF switches, to enable dynamic reconfiguration at the expense of extra cost and complexity [14, 23, 41]. Passive surfaces [45, 51, 57, 64, 71] have the benefits of low-cost, e.g., $1 for 60 thousand elements in AutoMS, and zero power consumption. Together with other specifications, such as frequency band, signal control type and range, substrate materials and corresponding fabrication processes, metasurface hardwares are very diverse and often need customization to suit different applications.

Hardware design scope. The hardware design of surface systems include microscopic element structure design, macroscopic spatial configurations of the element array, and optional (non-RF) peripherals like control circuits. The element structure is the core and basis of metasurface hardware, which decides various fundamental EM properties, such as reflection and transmission coefficients, phase shifts, operating frequency. Currently, optimizing these properties requires iterative parameter tuning and simulations due to the complex interplay between RF structure and electromagnetic waves.

Design workflow. The workflow of designing RF hardware varies among designers, generally following an expertise-guided iterative design process. It requires expertise and simulation tool usage; a designer lacking domain knowledge or a plain LLM can fail easily, e.g., wrong hardware specifications, poor design choices, aimless repetitive simulation. AutoRF formalizes the design workflow into modular steps that encapsulate domain expertise and facilitate both simulator integration and efficient design search (Figure 3). We do not consider iterative improvement based on fabricated prototypes; Full-wave EM simulation results match well with experimental results as shown in our case studies. Debugging fabricated prototype issues is beyond the scope of this paper.

Design file format. The metasurface and antenna design show as the 3D model of the metallic patterns and substrates, such as the structure in Figure 4 (d). The design is often saved as a parameterized project file of EM simulator software like HFSS. For passive surfaces, the design includes a set of metallic patterns on substrate, having both fixed parameters and varying parameters to achieve signal control, such as different ring radius for different phase shifts [45]. For programmable surface, the design includes a fixed metallic pattern and a set of circuit component values, such as different varactor capacitance to control phase shifts in Figure 10. We further export the element design and assemble the element array for fabrication as in section 4.

2.2 The Case for Automated RF Design

Facilitating new system development. For new application and deployment scenarios, researchers explore various surface-based networking and sensing systems. However, such a process is challenging, requiring knowledge of design principles and simulation tools, creating a barrier for people without domain expertise. An automated system would greatly ease the design process, allowing researchers or developers to focus on the high-level system goals and application software, outsourcing hardware expertise to AutoRF.

Adapting to practical scenarios. Practical usage scenarios often need hardware different from the research designs, such as different frequency bands, fabrication processes. Adapting existing designs can be very challenging and laborious. Wireless design specifications are often interdependent; a small change can require a complete

Table 1: Comparison with prior work. We support a much wider design space with minimal data needed, showing data efficiency and generalizability.

	Karahan *et al.* [36]	MetaGen [79]	**Ours**
Data Need	~100K	~100K	Templates (~10)
Freq range	24-100GHz	60-80 GHz	1-160 GHz
Types	Passive	Passive	Passive&Active
Layers	One	One	Multi

redesign of the hardware. Even replicating a design exactly is not straightforward as other people may not have the same design experience as the original designer. Therefore, we need a system to bridge the gap between research designs and various usage scenarios to speed up the adoption of a new technology like metasurfaces.

2.3 Hardware Design Automation

Conventional automation tools. Conventional design automation software uses simulators and tools to support *human* designers [1, 2]. These tools typically provide a GUI for users to manually edit designs and run simulations, and support parameter sweeps and optimizations using conventional algorithms such as gradient descent and genetic algorithms. General-purpose frameworks such as AutoOED [67] further automate the design of experiments (DoE) with optimization algorithms. However, compared to these conventional approaches, our LLM-based framework offers several key advantages: (1) ease of use through natural-language specification of design requirements, (2) the ability to autonomously operate complex simulation tools without manual intervention, and (3) efficient design iteration guided by the domain knowledge embedded in LLMs, akin to an experienced human designer. We quantitatively demonstrate that our approach achieves more efficient optimization in Table 4.

Neural network based generation. Recent work has explored using neural networks to assist the design for electromagnetic passive structure [36] and metasurfaces [10, 11, 79]. MetaGen [79] takes a step towards LLM-driven design, but LLM is only for user interface and focuses on simulation only.

The common approach is to collect a large amount of training data pairs between designs and performance metrics by running EM simulations, then train one or multiple neural networks to learn the mapping as either a forward performance predictor or a backward design generator [35]. Although this approach shows great results and speeds up the design process, it has several limitations. First, it requires experts to collect massive training datasets with simulators, which is time-consuming and labor-intensive, comparable to the traditional design process. And the datasets are typically not open-sourced. Second, the supervised learning approach limits the supported design space to the coverage of the training dataset, for example, the specific frequency band, substrate type and fabrication process. Any deviation would require collecting new training data and retraining a new model. Lastly, this design process lacks explainability and flexibility, as the neural networks are treated as black boxes. Therefore, this approach is not generalizable or scalable enough for our goal to support a wide range of metasurface designs and even other RF hardware types.

Specifically, recent RF design generation works [36, 79] require extensive expert knowledge for dataset preparation, particularly in sampling designs within a fabricable and pruned design space. Moreover, they constrain the design space to single-layer 2D grid passive patterns, where the 2D grid implicitly serves as a fixed template. In contrast, AutoRF explicitly encapsulates such expert knowledge into multiple reusable templates in the design library, enabling straightforward management and easy expansion by adding new templates. As a result, AutoRF supports a substantially larger design space spanning 1–160 GHz, multi-layer elements, and both passive and active metasurfaces (Table 1).

LLM-based automation. Large language models (LLM) have shown great capabilities in automating various tasks, including software development, data analysis, network management [33, 46]. Instead of task-specific training, they utilize the generalizability of LLMs as they are trained on a large corpus of data and have few-shot learning capabilities [16]. Recent works have shown promising results in automating analog circuit design [19, 30, 37], but they are limited to generating circuit topology based on existing python abstractions [56]. Designing RF hardware like metasurfaces is a more complex, multi-stage process involving physics, math reasoning, which lacks system abstractions or interfaces, and relies on EM simulators that have high computational costs. This requires both novel system abstractions to enable end-to-end automation and efficient design search and optimization algorithms to reduce the simulation cost.

3 System Design

AutoRF automates the end-to-end workflow from user demands to design files ready for fabrication.

Hardware design abstractions. To tackle the complexity of hardware design and leverage LLM reasoning capabilities, we introduce a set of hardware design abstractions, as shown in Figure 2. AutoRF translates the high-level design demands into target design specifications, then intermediate circuit models, and eventually specific parameterized designs. Our key insight is that LLMs reason most effectively within a single level of abstraction, i.e., high-level guidance for general questions, specific calculations for detailed questions. Intuitively, this is because LLMs are trained on domain-specific questions rather than end-to-end system-scale problems. By structuring the design process into layers, we break the problem into manageable sub-tasks. This modular approach also simplifies the system design and improves maintainability.

System workflow. AutoRF consists of three main LLM-based modules, *demand compiler*, *design proposer*, *design maker*, along with design tools, a *circuit model simulator* and an *EM simulator (HFSS)*, as shown in Figure 3. First, the demand compiler translates the high-level user demand input into structured hardware specifications, such as operating frequency bands, signal control capability, and substrate materials. Next, the design proposer takes the specifications, searches the design library for the closest template as starting point, and and proposes a design strategy to meet the specifications. To accelerate exploration, a lightweight circuit model simulator helps the proposer navigate the design space by providing insights, such as whether adding an extra surface layer enhances performance. Finally, the design maker decides and calculates the specific edits to

Table 2: Hardware specification primitives to describe the design targets. (Selected representative entries due to space.)

Parameter	Description
1. Basic Characteristics	
`freq_band`	Center operating frequency
`freq_range`	Range of operating frequency band
`polarization`	Linear, circular, or cross-polarized
2. Surface Operation Modes	
`op_mode`	Reflective or transmissive
`control_mode`	Programmable or passive
`control_type`	Amplitude, phase, etc.
`control_range`	e.g., "2-pi phase control range"
3. Deployment Constraints	
`substrate`	FR4, paper, Rogers, air gap, or other. It decides fabrication process.
`max_thickness`	Maximal allowed surface thickness

the design file, guided by the design rules and constraints stored in a database. It modifies the design, iteratively simulates and refines the design until it meets the target specifications or reaching the maximal allowed number of iterations.

3.1 Hardware Design Abstractions

Non-expert users often cannot provide detailed and precisely formatted target hardware specifications. Instead, we need to handle user demands expressed without a fixed format or syntax. For example, vague description and specific constraints can appear together. To address this, the demand compiler translates user inputs to predefined hardware specifications (Table 2). This ensures a standardized design description.

Hardware specification primitives. We categorize hardware specifications into three groups. The first category describes the *basic characteristics*, such as the operating frequency and range. The second category describes the metasurface operation *mode*, such as reflective or transmissive, programmable or passive. The last category describes *practical fabrication requirements*, such as substrate material(s), form factor, and hardware cost. Note that the substrate is an important specification as it determines the fabrication process, influencing cost, minimum pattern resolution, and substrate thickness options. Table 2 lists representative specifications, and the framework can be extended for more diverse user demands.

To handle incomplete or vague demands, we define a set of default values for missing fields and also utilize LLM to interpret implicit demands. For example, 'low-cost' can map to low-cost paper substrate, while 'low loss' can lead to high-end Rogers PCB material. We also add a multi-round dialogue to refine the user demands when too much ambiguity exists.

3.2 Design Search and Optimization

3.2.1 Design Search. The goal of the design proposer is to search for a suitable template and propose initial parameters and a design strategy for fine-grained edits.

Building on existing designs. Instead of training with massive datasets, we encode design knowledge with templates. This mimics the human design process and allows LLMs to leverage their training

Algorithm 1 LLM-driven design search and optimization

Input: Target specification $\mathcal{S}$; design library $\mathcal{L}$
Output: Final design $D^\star$ that satisfies $\mathcal{S}$ (or best-effort)

// Step 1: Library-based template search (design proposer)
$T \leftarrow \textsc{RetrieveTemplate}(\mathcal{S}, \mathcal{L})$ ▷ T: template
$C \leftarrow \textsc{ProposeDesign}(\mathcal{S}, T)$ ▷ C: circuit model
// Step 2: Coarse-grained probing via circuit model simulator
$\theta_0 \leftarrow \textsc{CircuitSim}(C, \mathcal{S})$ ▷ θ: design parameters
$D_0 \leftarrow \textsc{Instantiate}(T, \theta_0)$
// Step 3: Fine-grained EM optimization (design maker)
$D \leftarrow D_0, \theta \leftarrow \theta_0, H \leftarrow \emptyset$ ▷ H: optimization history
for $k = 1$ to K **do** ▷ K: maximum iterations
  $m_{\text{HFSS}} \leftarrow \text{HFSS\_SIM}(D)$
  if $\textsc{MeetsSpec}(m_{\text{HFSS}}, \mathcal{S})$ **then**
    return $D^\star \leftarrow D$
  end if
  $(D, \theta) \leftarrow \textsc{ProposeEdits}(\mathcal{S}, m_{\text{HFSS}}, \theta, H)$
  $H \leftarrow H \cup \{(\theta, m_{\text{HFSS}})\}$
end for
// Step 4: Output final or best-effort design
$D^\star \leftarrow \textsc{SelectBest}(H, \mathcal{S})$
return $D^\star$

on related *documents*, such as textbooks and technical documentation. Textbooks [65] or handbooks [15] essentially outline various design templates. By recalling training-stage knowledge, LLMs can choose suitable design templates and strategies. This approach also improve interpretability and allows us to support new hardware types by simply adding a few templates.

Design library. We construct a design library consisting of metasurface and antenna templates, as detailed in section 4. Each entry includes a parameterized HFSS project file with variables capturing the 3D hardware structure. For example, a split-ring resonator metasurface unit is characterized by parameters such as the ring radius, the ring gap width, and the substrate thickness. We leverage LLMs to extract design descriptions from research papers and annotate the templates with their prototype specifications.

Design Space. As each template has tens of variables and each variable can take a continuous range of values, the design space is vast and complex. This brings challenge that we address next, but also opens up opportunities for novel designs. Note that prior work [36, 79] use a discrete 2D grid as the design space, which is a different form of design template.

Template selection and strategy proposal. The design proposer takes the following steps: First, it identifies the top three related templates based on semantic similarity to user demands. If a perfect match exists, we use the corresponding template directly. Otherwise, we select the template with the fewest mismatched attributes and determines the simplest strategy to modify it. Then, we generate a proposal for editing the template design to meet the specifications, including common strategies like updating substrate material or scaling the surface pattern geometry to shift the operating frequency.

Efficient design exploration. For more challenging demands, AutoRF can explore novel metasurface designs, by adding an extra independent surface pattern layer or metal sheet. The theoretical foundation is that cascading multiple metallic patterns allows full

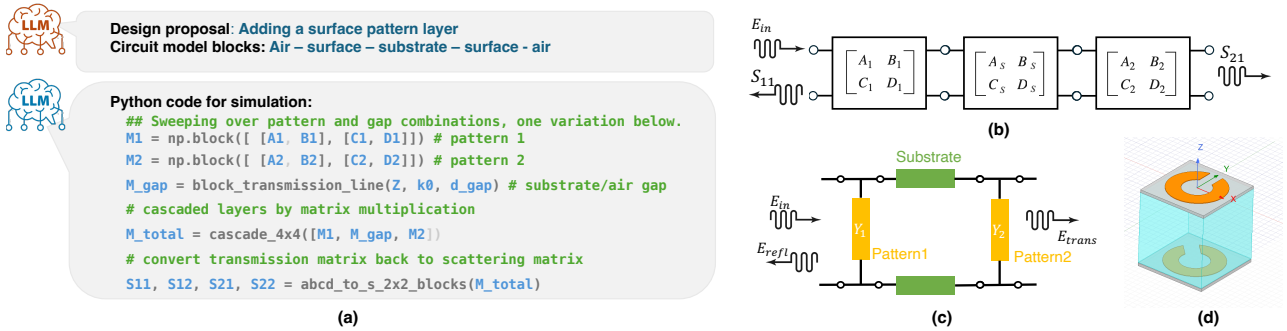


Figure 4: Circuit model simulator. (a) With the design proposal and circuit block description from the design proposer, the circuit model simulator generates the code to sweep the design space and returns the performance metrics. (b) Corresponding cascaded transmission (ABCD) matrix blocks, (c) corresponding circuit model. (d) Final metasurface element design.

control over wave properties by utilizing the resonance among layers [53]. However, such exploratory strategies significantly expand the design space, posing a challenge. Leaving the exploration to Maxwell equation-based EM simulator would be highly inefficient, requiring impractically long computation times. To address this, we draw inspiration from human designers, who use intermediate physics-based abstractions – circuit models, to efficiently reason and search.

Circuit model abstractions. Circuit models enable efficient prediction of hardware behaviors by modeling the metasurface and substrate as circuit components, characterized by their impedance values, which significantly reduces computational costs. The impedance of each common substrate is easily obtained from their dielectric properties, while the per-layer surface impedance can be precomputed via one-time profiling when adding a new template to the design library. With these impedance values, we can compute the transmission matrix (ABCD matrix) for the metasurface and substrate layers. According to microwave network analysis, the transmission matrix of a cascaded circuit is the product of the matrices of all its components, which is a simple matrix multiplication calculation [55]. By converting the result transmission matrix to a scattering matrix (S-matrix), we can efficiently estimate key performance metrics such as transmission power, phase shift, and reflection power. The relevant theoretical background is in Appendix subsection A.1.

Circuit model simulator. Although circuit model is a standard tool, our novel insight is to utilize it as a lightweight RF simulator, unseen in prior work. Combining such physics modeling and coding capability of LLMs, we develop an LLM-based circuit model simulator to explore the design space. When the design proposer module decides to explore a new design, e.g., adding a layer, it would first send a description of circuit block connections, such as *air - surface - substrate - surface - air*, to the circuit model simulator. The simulator generates and executes Python code to perform matrix computations and evaluate performance metrics. Specifically, it sweeps through different parameter variations and identify configurations that meet the target specifications. To facilitate this, we provide the LLM with standard microwave network matrix conversion as library functions. Then, the design proposer evaluates the returned metrics to assess the viability of the design strategy and determine key design parameter values. Figure 4 demonstrates this process with an example two-layer transmissive mmWave metasurface. The demand proposer generates a circuit model description, then the simulator calculates performance metrics based on cascaded matrix blocks (Figure 4(b)). By sweeping over different pattern and substrate thickness combinations, the simulator evaluates feasibility and determines the best achievable metric, such as transmission efficiency (S21).

3.2.2 Design Optimization. The design maker edits design parameters and objects by interacting with the full-wave EM simulator. It operates in two stages: (1) a reasoning-based planning stage that determines overall design strategy and large-scale structural edits, and (2) an iterative fine-grained tuning stage that adjusts parameters to precisely meet the target specifications. Even for relatively simple requests such as shifting the operating frequency, multiple iterations of tuning and optimization are required, as mechanical or rule-based parameter adjustments are ineffective due to the intricate coupling among EM structures. For more challenging design requests, AutoRF can yield genuine discoveries, as demonstrated in the case study (Figure 11).

Scaffolding instructions for reasoning. We want to leverage LLM's reasoning capability to derive the specific edits based on the template information and design strategy proposal from the design proposer. However, LLMs can provide incorrect or incomplete results for our task, primarily due to mixing overall planning with detailed calculation and practical constraints. To address this, we guide the reasoning process to align with system design reasoning steps–from high-level planning to details. We draw inspiration from a pedagogical strategy called scaffolding instruction [17, 66, 69], which uses questions to construct learners' understanding step by step. Similarly, we pose successive questions to LLMs to guide their reasoning process. Then, LLMs trained for reasoning, such as O3 and DeepSeek-r1, closely follow sub-questions and provide detailed, structured responses. Specifically, we use the following prompt structure as listed in Figure 5 (a). We first provide the task goal and context information. Then, the prompt asks a sequence of guiding questions, starting with high-level planning (what edits are needed), followed by knowledge recall (relevant equations or design principles), and specific parameter calculations.

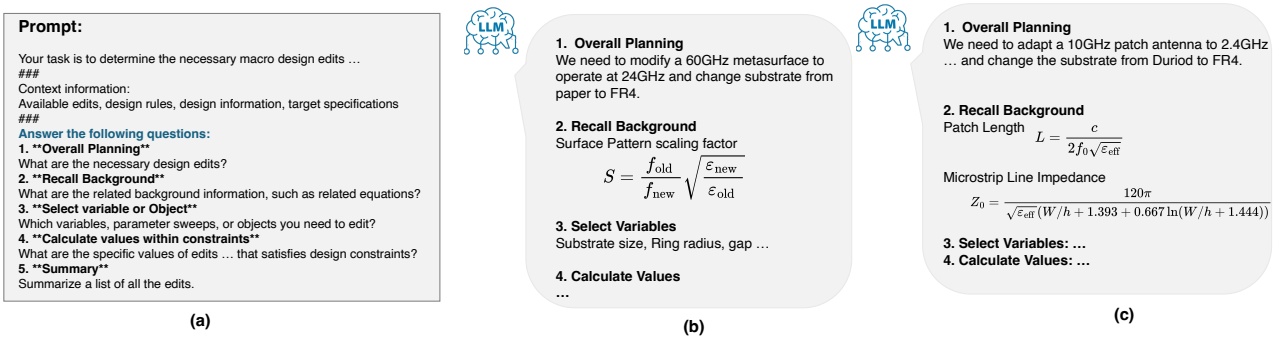


Figure 5: Scaffolding prompt and example response snippets of design maker. (a) We provide the task goal and context, then ask successive questions to guide LLM's reasoning. LLM answers the questions progressively and recall related equations to perform calculations for the specific edits, as shown in LLM responses of (b) metasurface design and (c) antenna design.

Recalling knowledge as optimization priors. A key part of this reasoning process is knowledge or background recall. The equations and design rules recalled by LLM serve as priors to effectively guide the design optimization. As shown in Figure 5 (b-c), LLMs can recall and apply domain-specific equations. For metasurface operating frequency tuning, the LLM calculates a scaling factor based on the target and the original frequencies, and then applies the scaling factor to related design variables. For antenna design, the LLM calculates the impedance of the microstrip line based on the substrate material to determine the correct dimension for impedance matching. Note that we do not hard-code the equations in the prompt and rely on LLM reasoning. This is more efficient than conventional gradient based optimization, which lacks priors about the search space. On the other hand, pure equation-based design alteration can be inaccurate due to complex EM coupling effects. By combining equation-based reasoning with iterative simulation, our approach achieves both speed and accuracy, as demonstrated in Table 4.

Fine-grained design tuning. The fine-grained tuning stage performs iterative variable adjustments to precisely meet target specifications. While the reasoning stage provides a calculation, it cannot guarantee perfect accuracy due to the complexity of EM resonance and coupling, especially in metasurfaces. Based on simulation results, we use a set of Python functions to generate reports summarizing key performance metrics based on the hardware type—e.g., S11 for antennas, reflection/transmission efficiency and phase shift range for metasurfaces. The design maker then evaluates whether all design specifications are met. If not, it determines how to update design variables. The simulation and tuning iterations continue until all the specifications are met or reaching a predefined iteration limit. We also store history records as memory to make informed tuning decisions [63, 73].

Array assembly. Our framework focuses on element structure design and assembles them to generate the full surface as final output. Specifically, we use uniform element array for programmable surfaces following common practice [14, 29, 41]. For passive surfaces, we prepare a set of composable patterns, such as split rings with different sizes for phase shifts, and assemble elements based on external configurations as in prior work [45, 58]. By default, we generate beam steering surfaces for evaluation. Note that our design simulation already takes account of coupling between elements. Upstream application systems can set array configurations and sizes easily with python scripts, which are external to our framework.

Extending to Antenna design. Our abstractions and system design are applicable to other RF hardware types. To demonstrate this, we extend to antenna design by adding design templates and evaluation metrics. Antenna hardware specifications align with those of metasurfaces, except they lack surface operation and signal control parameters. The key objective in antenna design is achieving impedance matching at the target frequency. We use the reflection coefficient S11 as the primary evaluation metric, as it is standard for assessing impedance matching. Following the same workflow, our system effectively adapts antenna designs as in subsection 5.3.

3.3 Training LLM with Feedback

To further improve LLM capability for design optimization, we propose novel task-specific feedback mechanism, including design rule reviewer agent and custom EM simulator APIs. Built on these, we perform rejection sampling fine-tuning.

Design rules and constraints. We maintain a database of design rules and constraints from human experts, e.g., system maintainers of AutoRF, and add them to LLM prompt as extra context. Design rules encompass high-level *guidelines* derived from textbook knowledge and engineering experience, for instance, 'metasurface per-element substrate size is half of the wavelength or smaller.'. Such guidance helps LLM to recall related knowledge or equations and use reasoning to make design edits as shown in Figure 5. Design *constraints*, on the other hand, impose practical limitations and user preferences, such as 'must choose from the following substrate materials: FR4, Rogers, paper, PVC plastic, ... the available FR4 substrate thickness is 0.4, 1mm, 1.6mm', 'the minimum resolution for surface pattern is 0.05mm', 'Round all your output to 3 significant digits.'

Rule reviewer. Due to the probabilistic nature of LLMs, they may overlook design rules, constraints, or make calculation errors. To mitigate this, we introduce a separate LLM agent as a design reviewer to verify the edits. The reviewer checks the output against design rules and constraints, identifying any violations. If a violation is found, it rejects the output and requests a revision based on the identified issues. This process repeats until the edits pass the review

or the maximum number of iterations is reached. This approach resembles a multi-agent debate system [18].

EM simulator interface. Although HFSS provides a Python interface [13], it differs significantly from common Python packages and LLM often mix different versions of APIs, causing errors. We introduce an abstraction layer that wraps HFSS APIs, which provides clearly named, semantically meaningful design editing functions: (1) setting the EM simulation frequency range, (2) editing design variable values, (3) editing design variable variations that corresponds to different signal control states, for example, a set of pattern lengths can control phase shift values, (4) adding, refining or deleting substrate layers, (5) adding extra surface layers, any interdependent pattern layer or metal sheet. Underlying the APIs, we craft robust error handling mechanisms to catch and report errors.

Rejection sampling fine-tuning. We fine-tune our LLM with filtered trajectories based on system feedback, inspired by training reasoning models with rejection sampling [32]. We first sample a dataset of LLM generated traces for various design optimization tasks. Using the rejection of the design reviewer and errors of simulator as selection criteria, we filter out low-quality output samples, i.e., rejecting the ones caused errors, and fine-tune the LLM on the high-quality subset. This allows the LLM to internalize the design rules imposed by the reviewer and improve its ability on using EM simulator programming interfaces. We formally describe the process here. Our filtering criteria can be written as follow.

$$v(x, y) = \mathbf{1}\Big[\text{EM\_pass}(x, y) \ \wedge \ \text{Rule\_pass}(x, y)\Big],$$

where x is the input of design optimization context, y is the LLM output, EM_pass$(\cdot)$ means the EM simulator finishes without error, and Rule_pass$(\cdot)$ means the reviewer agent finds no rule violation. We sample a large amount of data $\mathcal{D} = \{(x, y)\}$ using the base LLM. Then, we apply the validity predicate to reject erroneous samples and obtain the dataset

$$\mathcal{D}_{\text{RS}} = \big\{(x, y) \ \big| \ v(x, y) = 1\big\},$$

for fine-tuning. Finally, we train the LLM with loss function

$$\mathcal{L}_{\text{RS}}(\theta) = -\frac{1}{|\mathcal{D}_{\text{RS}}|} \sum_{(x,y)\in\mathcal{D}_{\text{RS}}} \log \pi_\theta(y \mid x),$$

where $\pi_\theta(y \mid x)$ is the language model parameterized by θ.

4 Implementation

LLM API and other software. We implement AutoRF in Python, following a modular architecture. Our default LLM is ChatGPT o3 [49] invoked via API calls for all framework modules. We fine-tune a GPT-OSS-20B model [48], with a A100 GPU, to operate the EM simulator at design maker module. We also explore other LLMs, such as o3-mini and DeepSeek-r1 [32]. Note that we depend on advanced third-party LLM API for best performance, but achieve comparable performance with different LLMs as in Figure 7. We use `numpy` to perform numerical matrix calculations in the circuit model simulator and use HFSS [1] via pyAEDT [13] to run EM simulations.

Design templates. We collect templates for the design library from prior metasurface research work, through open source access and manual replication [14, 31, 41, 45, 61, 77], as detailed in Table 3.

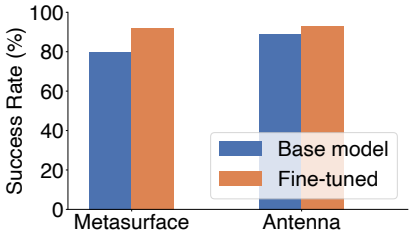


Figure 6: Success rate of design tasks.

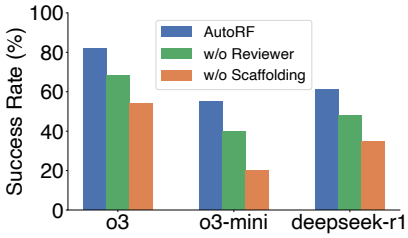


Figure 7: Success rate with different LLMs.

These represent distinct types of metasurfaces, allowing AutoRF to generate a variety of designs spanning common use cases. We also extract 10 antenna templates from HFSS toolkit [12], covering different types, including dipole, patch, bowtie and inverted-F antennas.

Metasurface fabrication. We use Python scripts to extract the metallic patterns of surface element designs from HFSS and convert them to fabrication files. Specifically, we first extract DXF files for each element, then assemble all elements into a surface element array to have measurable signal control effects. We further convert the array to a Gerber file for standard PCB fabrication or a PDF file for paper-based hot stamping fabrication as in prior work [31, 45, 61]. [43, 45].

5 Evaluation

We aim to evaluate (1) correctness and success probability of AutoRF through large-scale dataset based simulation testing; and (2) practicality and accuracy of generated designs through fabricate hardware prototypes and real-world experiments.

5.1 Experiment Setup

Evaluation approach and metrics. We use the *success rate* as the metric, which is defined as the ratio of successful designs to the total number of designs requested. A design is considered successful if it meets all specifications in simulation, first verified by the design maker using an LLM and then double-checked by human experts against the original user demand. Since our task involves objective, quantifiable performance specifications, the evaluation is not influenced by different users' subjective opinions. Nevertheless, we survey several potential users and confirm the applicability of AutoRF and the generated designs.

Simulation-to-real gap. Our simulation accuracy is underpinned by HFSS, an industry-standard 3D EM simulation tool that achieves high fidelity through finite element methods. In our experiments,

Table 3: Metasurface design templates. Replicated designs may differ from research work slightly. Transmissive (T) and reflective (R).

Metasurface	Freq Band	Signal Control	Mode	Control Type	Substrate
RFocus [14]	2.4 GHz	Amplitude	T & R	Programmable	FR4
RF-Mediator [41]	2.4 GHz	Impedance	T & R	Programmable	FPC
Split Ring [31, 61]	60 GHz	Phase	T	Passive	Paper
AutoMS [45]	60 GHz	Phase	R	Passive	Paper
Hydra [77]	3.5 GHz	Phase	R	Programmable	FR4

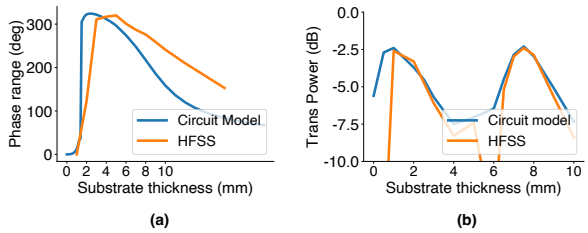


Figure 8: Circuit model accuracy. The estimated values match very well with HFSS simulations, while the speed is much faster (HFSS takes hours, circuit model takes < 0.1s).

simulation results match well with measurements on fabricated prototypes, though they do not guarantee identical real-world performance. We selected representative designs for fabrication as case studies (Section 5.4), given the cost and time constraints of prototyping. Note that residual discrepancies may arise from fabrication imperfections, which are beyond our control. Employing high-end fabrication processes (e.g., RF PCB fabrication with high-frequency specialized substrates) can minimize such gaps by reducing tolerances and material variations.

Test dataset. To create a large-scale and comprehensive test dataset, we randomly sample 100 design specification samples, covering frequencies from 0.9GHz to 160GHz, substrate materials like FR4, Rogers, and printer paper, programmable and passive surfaces, transmissive and reflective metasurfaces, phase control and amplitude control metasurfaces. We exclude programmable transmissive phase-shifting metasurface from the samples due to the lack of a suitable template design, while we expect the extension is straightforward by adding a corresponding template in the future. Similarly, we create 50 antenna design demands, covering frequencies from 0.9GHz to 10GHz and different substrate materials.

Baseline methods. As discussed in subsection 2.3, there is no direct baseline for AutoRF, as we aim for generalizable few-shot design generation. For antenna design automation, a direct baseline is the antenna toolkit from HFSS [12]. It synthesizes antenna designs based on equations, according to user's selection of template design, operating frequency and substrate material as input. We compare AutoRF with the HFSS toolkit in Table 4.

Experimental setup. We use SNA5032A vector network analyzer (VNA) to measure hardware operating below 26GHz and another high-frequency VNA to measure hardware operating above 26GHz. With standard horn antennas or Vivaldi antennas connected to VNA ports and pointing towards the surface, we measure power and phase of signals passing through or being reflected by the surface.

Table 4: Comparison with other design automation methods. AutoRF achieves the best accuracy for operating frequency with a short time.

	Freq Error	Time (iterations)	Metasurface
HFSS Toolkit	8.6%	0	No
Toolkit + Gradient Opt	2.1%	79.5	No
Toolkit + Genetic Algo	4.2%	9.53	No
Ours	0.95%	2.1	Yes

5.2 Metasurface Design Automation

Success rate. We achieve a success rate of 92% for 100 metasurface design tasks (Figure 6). Considering the complexity and diversity of surface hardware we cover, this is a very high success rate. Without the fine-tuned model of design maker module, the success rate drops to 80%, which shows the effectiveness of our fine-tuning process. The common failure cases are: (1) the design does not meet the target specifications after reaching maximal tuning iterations, 5 for our setting. We set this number low to save time. (2) LLM of the design maker skips one of the design constraints or design rules, leading to a wrong design. The design reviewer can help catch some of these mistakes, but not all. Our rejection sampling fine-tuning process further helps reduce such mistakes.

Impact of different LLMs. We evaluate AutoRF using different LLMs, including OpenAI o3, o3-mini and DeepSeek-r1. Despite showing similar performance for various benchmarks, we find that o3 outperforms others, while o3-mini performs the worst with a 55% success rate. We find that o3-mini more often skips design constraints and rules, leading to incorrect designs, such as an unrealistic substrate thickness. o3-mini also tends to reply tersely, which can lead to incomplete design edits. Moreover, DeepSeek-r1 provides a good balance between the LLM API cost and performance, achieving 62% success rate. The primary issue for DeepSeek-r1 is that it often fails to follow desired output format. Overall, AutoRF works with different LLMs with varying but good performance.

Ablation study. In subsubsection 3.2.2, we present scaffolding instructions and design reviewer as two key methods. We evaluate their impact on the performance by removing them and observe the performance loss; the higher loss, the more helpful the method is. As shown in Figure 7, the success rate of AutoRF without design reviewer is 66%, 40%, and 48% for o3, o3-mini and DeepSeek-r1 respectively, which is around 10% lower because mistakes made by design maker are not caught. If we further remove the scaffolding instructions by only prompting with the overall context and one output question, which leads to unstructured reasoning process

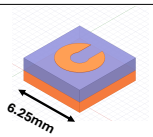
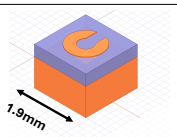
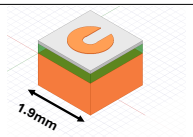
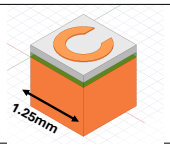
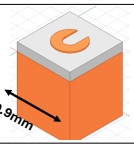
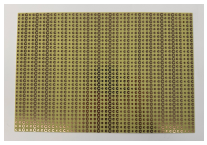
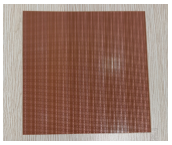
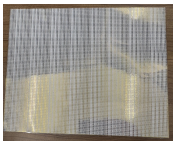
Demands	"I want a metasurface for 24GHz cellular coverage enhancement. It should use PCB fabrication process."	"Design a phase control metasurface for 79GHz radar sensing using FR4 substrate."	"Design a low-cost surface for 79GHz mmWave signal reflection control."	"Design a reflective surface for beamforming at 120GHz band."	"Design a metasurface for 160GHz signal control. The surface should be low cost."
Specifications	24GHz, Reflective, Phase control, FR4 …	79GHz, Reflective, Phase control, FR4 …	79GHz, Reflective, Phase control, Paper…	120GHz, Reflective, Phase control, Paper …	160GHz, Reflective, Phase control, Paper …
Surface element design					
Fabricated Surfaces					

Figure 9: Case Study: mmWave and sub-THz metasurface generalization across frequencies and substrate materials.

and response, the success rate drops further by around 10% for all LLMs. The results show that both methods are important for the performance of AutoRF.

Circuit model simulator accuracy and speedup. Here, we evaluate the accuracy of the circuit model simulator by comparing with HFSS simulation results for the second and third cases in subsection 5.4. As we can not simulate the entire design space with HFSS due to high computation load, we focus on exploring the best substrate thickness for two surfaces. As shown in Figure 8, the circuit model simulator can accurately predict (a) the phase range of programmable phase-shifting surface and (b) transmissive power for the transmissive mmWave metasurface. Meanwhile, the time taken by the circuit model simulator is less than 0.1 second, while HFSS takes hours to simulate the same designs.

Token cost and design time. We measure the token usage of AutoRF across 40 design tasks. On average, each design consumes approximately 92K tokens (71K input, 21K output), costing less than $1 per design at current frontier LLM API pricing. The average end-to-end design time is approximately 2 hours, dominated by the computationally intensive EM simulations. The wall-clock time per iteration varies with available resources and configurations (e.g., CPU cores, parallelism, simulation accuracy settings); accordingly, we focus on minimizing the number of iterations required (table 4). In contrast, human designers typically require days to master a specific metasurface design task and tools like HFSS, highlighting the substantial cost and time advantages of our system.

5.3 Antenna Design Automation

Success rate. For antenna design automation, we achieve a success rate of 93% for 50 testing design tasks (Figure 6). The success rate is higher than metasurfaces because the antenna specification range we set is smaller, while metasurface designs cover a wider frequency range and different signal control capabilities. The cause of failure cases is similar to metasurface design automation. The results show that AutoRF can generalize to antenna design with high success rate.

Comparison with conventional tools. We compare AutoRF against the HFSS antenna toolkit [12] and conventional optimization algorithms on 15 antenna design tasks. The toolkit synthesizes antenna designs via equation-based calculations given a user-selected template, operating frequency, and substrate material. However, this approach cannot accurately predict the resulting operating frequency, particularly when both frequency and substrate material are changed simultaneously. We quantify design accuracy as the relative center-frequency error (%). AutoRF achieves less than 1% error, whereas the HFSS toolkit yields 8.6% error (Table 4). Applying the HFSS gradient-based optimizer on top of the toolkit reduces the error to 2.1%, but requires 79.5 iterations on average to converge—significantly more than the 2.1 iterations needed by AutoRF, which leverages domain knowledge (e.g., scaling antenna dimensions proportionally to shift frequency). Moreover, neither the toolkit nor the gradient optimizer supports metasurface design. Although gradient optimization is general-purpose, its high iteration cost restricts it to tuning only a few variables over limited ranges, making it suitable only as an aid to expert designers. We additionally compare with a genetic algorithm, widely used for design optimization (e.g., in AutoOED [67]); it achieves 4.2% error with 9.53 iterations on average, which is both less accurate and slower than our system.

5.4 Case Studies

Next, we conduct four design case studies to demonstrate the generalizability and capability of proposing new designs.

mmWave metasurface design generalization. We show the generalizability of AutoRF by designing mmWave metasurfaces across a wide frequency range from 24GHz to 160GHz and different substrate materials. In Figure 9, we list 5 different user demands and the corresponding design specifications, output designs, and the fabricated surfaces. AutoRF accurately captures the design specifications, selects AutoMS [45] as the closest template, and generates the corresponding designs that achieve required hardware specifications, such as 2π phase control range. Note that simply scaling all design variables for different frequencies would not work. First, besides

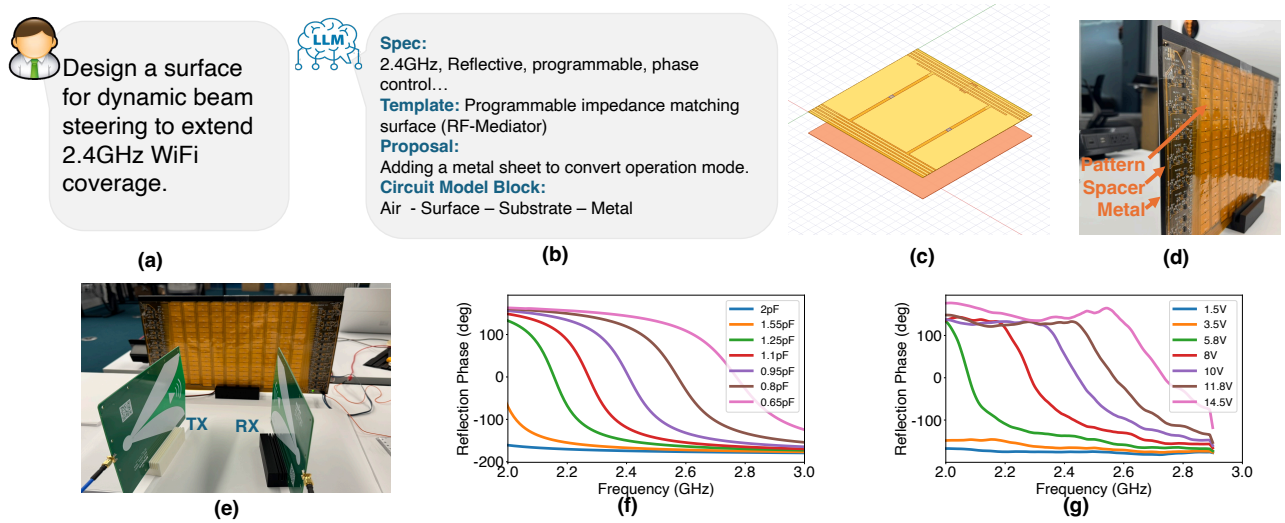


Figure 10: Case study: Reflective programmable phase-shifting metasurface. (a) User demand (b) Summarized LLM outputs. (c) Generated surface design (d-e) Fabricated surface and measurement setup. (f) Simulated reflection phase control range with different capacitance. (g) Measured reflection phase versus different biasing voltages applied to the surface.

updating the parameters, our system edits substrate intelligently, i.e., replacing material or removing PVC layer to suit the design need. Second, our output designs must allow practical fabrications by following design constraints, such as paper substrate can only be 0.15 mm. Lastly, the intricate EM coupling requires careful iterative tuning.

We fabricate four surfaces with phase configurations set to steer the beam towards 45 degrees and test the beam steering performance. We see over 20 dB power gain towards the target directions, which shows that the generated surfaces provide the desired phase control capabilities at the target frequencies.

Programmable phase-shifting metasurface. Next, we show the capability of AutoRF to create a new design by adding a metal sheet layer (Figure 10). AutoRF first interprets the user demand to get the specifications – a programmable phase-shifting metasurface at 2.4 GHz, with reflective being the default mode. Then it selects the RF-Mediator surface as the template design, which is a programmable impedance matching surface. By adding a metal sheet, we can convert the design to a programmable phase-shifting metasurface by reflection resonance. The design maker automatically generates the design proposal and circuit model block description. Then, the circuit model simulator fetches impedance values of the template from the design library, verifies the viability of the design and finds the best substrate thickness to be around 5 mm. The design maker then generates the final design by adding a metal sheet design object and tune related design parameters. The final design shows a high reflection power and near-2-pi phase control range as shown in Figure 10 (f). We fabricate the design and measure the reflection phase control range as shown in Figure 10 (d-e). By varying different voltages applied to the varactor diodes on the surface, we can control the capacitance of the diodes and subsequently the reflection phase of the surface. The measured phase shifts from 2 GHz to 3 GHz match well with the simulation, which verifies the design correctness.

Novel low-loss transmissive mmWave metasurface. We show a case study of designing a novel transmissive low-loss mmWave metasurface with two paper-based pattern layers in Figure 11. AutoRF chooses transmissive split ring resonator [31, 61] as the template design. Despite having a large phase control range for cross-polarization signals, the template design has a low cross-polarization power efficiency, which does not meet the user demand. The design maker then proposes adding a second layer of metallic pattern to improve the power efficiency and uses the circuit model simulator (cross-polarization is also supported by circuit model) to verify and suggest suitable surface pattern combinations and substrate thickness. Then, the design maker edits the design in HFSS and fine-tune the design parameters.

This leads to the final design, the first high-efficiency phase-control transmissive mmWave metasurface that allows low-cost hot-stamping fabrication. For mmWave transmissive surfaces, we often either sacrifice power efficiency or use very expensive low-loss PCB substrate. This novel design brings new opportunities for low-cost metasurface based mmWave sensing and communication systems. We fabricate the design and measure the transmissive power and phase control range as shown in Figure 11 (d-h). We use a 3D-printed plastic frame to create an air gap between the two hot-stamped paper-based metallic pattern layers and assemble them, with a size of 20 cm by 25 cm. The fabricated surface shows a high transmitted through power (S21) of -2.5 dB at 24Ghz, which is over 4dB higher than the original one-layer template design. We have a 2-π phase control range measured with different patterns, which is further verified with a beam steering surface towards 45 degrees in Figure 11 (h).

Antenna design adaptation. We show two antenna design examples in Figure 12. AutoRF automatically adapts the designs to 2.4GHz and 5.8GHz by tuning the design parameters, such as feedline width,

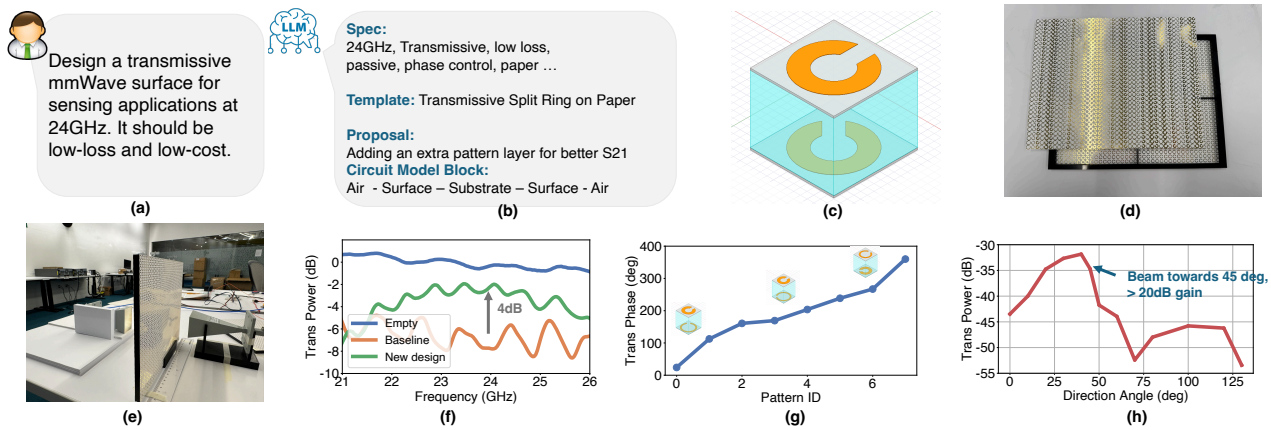


Figure 11: Case Study: Transmissive low-loss paper-based mmWave metasurface. (a) User demand (b) Summarized LLM outputs (c) Generated surface design (d-e) Fabricated surface and measurement setup. (f) Measured transmission power efficiency (normalized). (g) Measured phase shifts versus different patterns. (h) Measured beam steering performance towards 45 degrees.

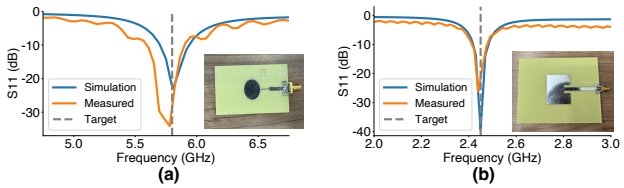


Figure 12: Case study: Antenna design automation. We show results for (a) a 5.8GHz elliptical patch antenna and (b) a 2.4GHz rectangular patch antenna.

patch length. As shown in Figure 12, we fabricate the designs from AutoRF and measure the port reflection power (S11). Low S11 means signal power is transmitted away from antenna to air; the range with S11 below -10dB shows the operating frequency range, which matches with targets. These two designs are nontrivial, representative examples because they each include 3 structures: a feedline, a patch and an impedance matching structure, tuning these three structures rely on different equations and reasoning.

6 Conclusion

We presented AutoRF, the first agentic framework for automated RF hardware design, bringing LLM-based reasoning to RF hardware design. AutoRF efficiently searches and optimizes designs, with circuit model simulator for coarse exploration followed by fine-grained EM simulator-based optimization. We also tailored feedback mechanisms to train a reliable and performant language model. AutoRF achieves around 90% end-to-end success rate for metasurfaces and antennas, while delivering over 10X optimization time speedups. Fabricated case studies match simulations well, demonstrating generalizability and capability for novel designs. We believe this work takes a concrete step toward a faster, more scalable, and more accessible RF design process for next-generation wireless systems.

References

[1] Ansys HFSS 3D High Frequency Simulation Software. https://www.ansys.com/products/electronics/ansys-hfss.

[2] AWR Design Environment Platform. https://www.cadence.com/en_US/home/tools/system-analysis/rf-microwave-design/awr-design-environment-platform.html.

[3] China Mobile and ZTE successfully complete application verification of 5G-A Reconfigurable Intelligent Surface for Asian Games. https://www.zte.com.cn/global/about/news/china-mobile-and-zte-successfully-complete-application-verification-of-5g-a-reconfigurable-intelligent-surface-ris-for-asian-games.html.

[4] DOCOMO Conducts World's First Trial of Transmissive Metasurface on Window to Deliver Indoor Radio Waves to Outdoor Foot of Building. https://www.docomo.ne.jp/english/info/media_center/pr/2023/0130_02.html,.

[5] EMPOWER: Final technology roadmap for advanced wireless. https://www.advancedwireless.eu/wp-content/uploads/2022/02/Del2-5-Final-technology-roadmap.pdf,.

[6] ETSI first report for RIS standardization. https://www.etsi.org/deliver/etsi_gr/RIS/001_099/001/01.01.01_60/gr_RIS001v010101p.pdf,.

[7] ITU International Mobile Telecommunications (IMT) for 2030 and beyond. https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2160-0-202311-I!!PDF-E.pdf,.

[8] Next G Alliance Report: Roadmap to 6G. https://nextgalliance.org/wp-content/uploads/2022/02/NextGA-Roadmap.pdf,.

[9] Devansh R Agrawal, Yuji Tanabe, Desen Weng, Andrew Ma, Stephanie Hsu, Song-Yan Liao, Zhe Zhen, Zi-Yi Zhu, Chuanbowen Sun, Zhenya Dong, et al. Conformal phased surfaces for wireless powering of bioelectronic microdevices. *Nature biomedical engineering*, 1(3):0043, 2017.

[10] Sensong An, Clayton Fowler, Bowen Zheng, Mikhail Y Shalaginov, Hong Tang, Hang Li, Li Zhou, Jun Ding, Anuradha Murthy Agarwal, Clara Rivero-Baleine, et al. A deep learning approach for objective-driven all-dielectric metasurface design. *Acs Photonics*, 6(12):3196–3207, 2019.

[11] Sensong An, Bowen Zheng, Hong Tang, Mikhail Y Shalaginov, Li Zhou, Hang Li, Myungkoo Kang, Kathleen A Richardson, Tian Gu, Juejun Hu, et al. Multifunctional metasurface design with a generative adversarial network. *Advanced Optical Materials*, 9(5):2001433, 2021.

[12] Ansys. AEDT Antenna Toolkit documentation. https://aedt.antenna.toolkit.docs.pyansys.com/,.

[13] Ansys. PyAEDT: Python Library for Ansys Electronics Desktop. https://github.com/ansys/pyaedt,.

[14] Venkat Arun and Hari Balakrishnan. RFocus: Practical Beamforming for Small Devices. In *Symposium on Networked Systems Design and Implementation (NSDI)*, pages 1047–1061. USENIX, 2020.

[15] Constantine A Balanis. *Antenna theory: analysis and design*. John wiley & sons, 2015.

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[17] Jerome Seymour Bruner. *Toward a theory of instruction*. Harvard university press, 1974.

[18] Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023.

[19] Chen-Chia Chang, Yikang Shen, Shaoze Fan, Jing Li, Shun Zhang, Ningyuan Cao, Yiran Chen, and Xin Zhang. Lamagic: Language-model-based topology generation for analog integrated circuits. *arXiv preprint arXiv:2407.18269*, 2024.

[20] Baicheng Chen, John Nolan, and Xinyu Zhang. MetaBioLiq: A Wearable Passive Metasurface Aided mmWave Sensing Platform for BioFluids. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '24, New York, NY, USA, 2024. ACM.

[21] Lili Chen, Wenjun Hu, Kyle Jamieson, Xiaojiang Chen, Dingyi Fang, and Jeremy Gummeson. Pushing the physical limits of IoT devices with programmable metasurfaces. In *Symposium on Networked Systems Design and Implementation (NSDI)*, pages 425–438. USENIX, 2021.

[22] Lili Chen, Bozhong Yu, Yongjian Fu, Ju Ren, Jeremy Gummeson, and Yaoxue Zhang. Pushing Wireless Charging from Station to Travel. ACM MobiCom '24, New York, NY, USA, 2024. Association for Computing Machinery.

[23] Kun Woo Cho, Mohammad H Mazaheri, Jeremy Gummeson, Omid Abari, and Kyle Jamieson. mmWall: A steerable, transflective metamaterial surface for NextG mmWave networks. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 1647–1665, 2023.

[24] Tie Jun Cui, Mei Qing Qi, Xiang Wan, Jie Zhao, and Qiang Cheng. Coding metamaterials, digital metamaterials and programmable metamaterials. *Light: science & applications*, 3(10):e218–e218, 2014.

[25] Marco Di Renzo, Fadil H Danufane, and Sergei Tretyakov. Communication models for reconfigurable intelligent surfaces: From surface electromagnetics to wireless networks optimization. *Proceedings of the IEEE*, 110(9):1164–1209, 2022.

[26] Marco Di Renzo and Sergei Tretyakov. Reconfigurable intelligent surfaces [scanning the issue]. *Proceedings of the IEEE*, 110(9):1159–1163, 2022.

[27] NTT Docomo. 5G Evolution and 6G Whitepaper. https://www.docomo.ne.jp/english/binary/pdf/corporate/technology/whitepaper_6g/DOCOMO_6G_White_PaperEN_v5.0.pdf,.

[28] Manideep Dunna, Chi Zhang, Daniel Sievenpiper, and Dinesh Bharadia. ScatterMIMO: enabling virtual MIMO with smart surfaces. In *Proceedings of International Conference on Mobile Computing and Networking (Mobicom)*, pages 1–14. ACM, 2020.

[29] Chao Feng, Xinyi Li, Yangfan Zhang, Xiaojing Wang, Liqiong Chang, Fuwei Wang, Xinyu Zhang, and Xiaojiang Chen. RFlens: metasurface-enabled beamforming for IoT communication and sensing. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, MobiCom '21, pages 587–600, New York, NY, USA, October 2021. Association for Computing Machinery.

[30] Jian Gao, Weidong Cao, Junyi Yang, and Xuan Zhang. Analoggenie: A generative engine for automatic discovery of analog circuit topologies. *arXiv preprint arXiv:2503.00205*, 2025.

[31] Hichem Guerboukha, Yasith Amarasinghe, Rabi Shrestha, Angela Pizzuto, and Daniel M. Mittleman. High-volume rapid prototyping technique for terahertz metallic metasurfaces. *Optics Express*, 29(9):13806–13814, April 2021.

[32] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[33] Zhiyuan He, Aashish Gottipati, Lili Qiu, Xufang Luo, Kenuo Xu, Yuqing Yang, and Francis Y Yan. Designing network algorithms via large language models. In *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks*, pages 205–212, 2024.

[34] Huawei. 6G: The Next Horizon. https://www.huawei.com/en/huaweitech/future-technologies/6g-white-paper,.

[35] Wenye Ji, Jin Chang, He-Xiu Xu, Jian Rong Gao, Simon Gröblacher, H Paul Urbach, and Aurèle JL Adam. Recent advances in metasurface design and quantum optics applications with machine learning, physics-informed neural networks, and topology optimization methods. *Light: Science & Applications*, 12(1):169, 2023.

[36] Emir Ali Karahan, Zheng Liu, Aggraj Gupta, Zijian Shao, Jonathan Zhou, Uday Khankhoje, and Kaushik Sengupta. Deep-learning enabled generalized inverse design of multi-port radio-frequency and sub-terahertz passives and integrated circuits. *Nature Communications*, 15(1):10734, 2024.

[37] Yao Lai, Sungyoung Lee, Guojin Chen, Souradip Poddar, Mengkang Hu, David Z Pan, and Ping Luo. Analogcoder: Analog circuit design via training-free code generation. *arXiv preprint arXiv:2405.14918*, 2024.

[38] Geoffroy Lerosey and Mathias Fink. Wavefront shaping for wireless communications in complex media: From time reversal to reconfigurable intelligent surfaces. *Proceedings of the IEEE*, 110(9):1210–1226, 2022.

[39] Xinyi Li, Chao Feng, Fengyi Song, Chenghan Jiang, Yangfan Zhang, Ke Li, Xinyu Zhang, and Xiaojiang Chen. Protego: securing wireless communication via programmable metasurface. In *Proceedings of ACM MobiCom*, pages 55–68, 2022.

[40] Zhuqi Li, Yaxiong Xie, Longfei Shangguan, Rotman Ivan Zelaya, Jeremy Gummeson, Wenjun Hu, and Kyle Jamieson. Towards Programming the Radio Environment with Large Arrays of Inexpensive Antennas. In *Symposium on Networked Systems Design and Implementation (NSDI)*, pages 285–300. USENIX, 2019.

[41] Ruichun Ma and Wenjun Hu. RF-Mediator: Tuning Medium Interfaces with Flexible Metasurfaces. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 155–169, 2024.

[42] Ruichun Ma and Wenjun Hu. Cross-medium networking with transflective flexible metasurfaces. *IEEE Transactions on Networking*, 2025.

[43] Ruichun Ma, Lili Qiu, and Wenjun Hu. SurfOS: Towards an Operating System for Programmable Radio Environments. In *Proceedings of the Twenty-Third ACM Workshop on Hot Topics in Networks (HotNets)*, 2024.

[44] Ruichun Ma, R Ivan Zelaya, and Wenjun Hu. Softly, deftly, scrolls unfurl their splendor: Rolling flexible surfaces for wideband wireless. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2023.

[45] Ruichun Ma, Shicheng Zheng, Hao Pan, Lili Qiu, Xingyu Chen, Liangyu Liu, Yihong Liu, Wenjun Hu, and Ju Ren. AutoMS: Automated Service for mmWave Coverage Optimization using Low-cost Metasurfaces. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, MobiCom '24, pages 62–76, New York, NY, USA, 2024. ACM.

[46] Sathiya Kumaran Mani, Yajie Zhou, Kevin Hsieh, Santiago Segarra, Trevor Eberl, Eliran Azulai, Ido Frizler, Ranveer Chandra, and Srikanth Kandula. Enhancing network management using code generated by large language models. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, pages 196–204, 2023.

[47] RCR Wireless News. Qualcomm focused on making 5G mmWave more cost effective. https://www.rcrwireless.com/20240325/spectrum/qualcomm-focused-on-making-5g-mmwave-more-cost-effective,.

[48] OpenAI. Introducing gpt-oss. https://openai.com/index/introducing-gpt-oss/,.

[49] OpenAI. Introducing OpenAI o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/,.

[50] OpenAI. Learning to reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/,.

[51] Hao Pan, Lili Qiu, Bei Ouyang, Shicheng Zheng, Yongzhao Zhang, Yi-Chao Chen, and Guangtao Xue. PMSat: Optimizing Passive Metasurface for Low Earth Orbit Satellite Communication. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2023.

[52] John B Pendry, David Schurig, and David R Smith. Controlling electromagnetic fields. *science*, 312(5781):1780–1782, 2006.

[53] Carl Pfeiffer and Anthony Grbic. Cascaded metasurfaces for complete phase and polarization control. *Applied Physics Letters*, 102(23):231116, June 2013.

[54] Carl Pfeiffer and Anthony Grbic. Metamaterial Huygens' Surfaces: Tailoring Wave Fronts with Reflectionless Sheets. *Physical Review Letters*, 110(19):197401, May 2013.

[55] David M Pozar. *Microwave engineering*. John wiley & sons, 2011.

[56] PySpice. PySpice: Simulate electronic circuit using Python and the Ngspice / Xyce simulators. https://github.com/PySpice-org/PySpice,.

[57] Kun Qian, Lulu Yao, Xinyu Zhang, and Tse Nga Ng. MilliMirror: 3D printed reflecting surface for millimeter-wave coverage expansion. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pages 15–28, 2022.

[58] Kun Qian and Xinyu Zhang. Millimirror: 3d printed reflecting surface for millimeter-wave coverage expansion. In *Proceedings of ACM MobiCom*, 2022.

[59] Tianshuo Qiu, Xin Shi, Jiafu Wang, Yongfeng Li, Shaobo Qu, Qiang Cheng, Tiejun Cui, and Sai Sui. Deep learning: a rapid and efficient route to automatic metasurface design. *Advanced Science*, 6(12):1900128, 2019.

[60] Qualcomm. Vision, market drivers, and research directions on the path to 6G. https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/Qualcomm-Whitepaper-Vision-market-drivers-and-research-directions-on-the-path-to-6G.pdf,.

[61] Zhambyl Shaikhanov, Fahid Hassan, Hichem Guerboukha, Daniel Mittleman, and Edward Knightly. Metasurface-in-the-Middle Attack: From Theory to Experiment. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec '22, pages 257–267, New York, NY, USA, May 2022. Association for Computing Machinery.

[62] Jayanth Shenoy, Zikun Liu, Bill Tao, Zachary Kabelac, and Deepak Vasisht. RF-protect: privacy against device-free human tracking. In *Proceedings of the ACM SIGCOMM 2022 Conference*, pages 588–600, 2022.

[63] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

[64] Yiwen Song, Hao Pan, Longyuan Ge, Lili Qiu, Swarun Kumar, and Yi-Chao Chen. Microsurf: Guiding energy distribution inside microwave oven with metasurfaces. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '24, New York, NY, USA, 2024. ACM.

[65] Warren L Stutzman and Gary A Thiele. *Antenna theory and design*. John Wiley & Sons, 2012.

[66] Yuan Tian, Nan Xu, and Wenji Mao. A theory guided scaffolding instruction framework for llm-enabled metaphor reasoning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7731–7748, 2024.

[67] Yunsheng Tian, Mina Konaković Luković, Timothy Erps, Michael Foshey, and Wojciech Matusik. Autooed: Automated optimal experiment design platform. *arXiv preprint arXiv:2104.05959*, 2021.

[68] Mikko A Uusitalo, Patrik Rugeland, Mauro Renato Boldi, Emilio Calvanese Strinati, Panagiotis Demestichas, Mårten Ericson, Gerhard P Fettweis, Miltiadis C Filippou, Azeddine Gati, Marie-Helene Hamon, et al. 6g vision, value, use cases and technologies from european 6g flagship project hexa-x. *IEEE Access*, 9:160004–160020, 2021.

[69] Lev Semenovich Vygotsky and Michael Cole. *Mind in society: Development of higher psychological processes*. Harvard university press, 1978.

[70] Allen Welkie, Longfei Shangguan, Jeremy Gummeson, Wenjun Hu, and Kyle Jamieson. Programmable Radio Environments for Smart Spaces. In *Workshop on Hot Topics in Networks (HotNets)*. ACM, 2017.

[71] Timothy Woodford, Kun Qian, and Xinyu Zhang. Metasight: High-resolution nlos radar sensing through efficient metasurface encoding. In *Proceedings of the 21th ACM Conference on Embedded Networked Sensor Systems*, 2023.

[72] Zhanni Wu, Younes Ra'di, and Anthony Grbic. Tunable Metasurfaces: A Polarization Rotator Design. *Physical Review X*, 9(1):011036, February 2019.

[73] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[74] R Ivan Zelaya, William Sussman, Jeremy Gummeson, Kyle Jamieson, and Wenjun Hu. LAVA: fine-grained 3D indoor wireless coverage for small IoT devices. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*, pages 123–136. ACM, 2021.

[75] Ganlin Zhang, Dongheng Zhang, Hongyu Deng, Yun Wu, Fengquan Zhan, and Yan Chen. Practical passive indoor localization with intelligent reflecting surface. *IEEE Transactions on Mobile Computing*, 2024.

[76] Hongliang Zhang, Boya Di, Kaigui Bian, Zhu Han, H Vincent Poor, and Lingyang Song. Toward ubiquitous sensing and localization with reconfigurable intelligent surfaces. *Proceedings of the IEEE*, 110(9):1401–1422, 2022.

[77] Yangfan Zhang, Zhihao Hui, Hao Jia, Xiaojiang Chen, and Yaxiong Xie. Hydra: Attacking OFDM-base Communication System via Metasurfaces Generated Frequency Harmonics. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '24, New York, NY, USA, 2024. ACM.

[78] Yongzhao Zhang, Yezhou Wang, Lanqing Yang, Mei Wang, Yi-Chao Chen, Lili Qiu, Yihong Liu, Guangtao Xue, and Jiadi Yu. Acoustic sensing and communication using metasurface. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 1359–1374, 2023.

[79] Yizhe Zhao, Lili Chen, Xinyi Li, Yuejiang Dong, Ju Ren, and Yaoxue Zhang. Metagen: Llm-driven generative framework for intelligent metasurface element. In *Proceedings of The 23rd Annual International Conference on Mobile Systems, Applications and Services*, MobiSys '25, 2025.

[80] Jiafeng Zhou, Pei Zhang, Jiaqi Han, Long Li, and Yi Huang. Metamaterials and metasurfaces for wireless power transfer and energy harvesting. *Proceedings of the IEEE*, 110(1):31–55, 2021.

[81] ZTE. B5G Technology White Paper. https://www.zte.com.cn/content/dam/zte-site/res-www-zte-com-cn/white_paper/B5G_Technology_Whitepaper_7.pdf,.

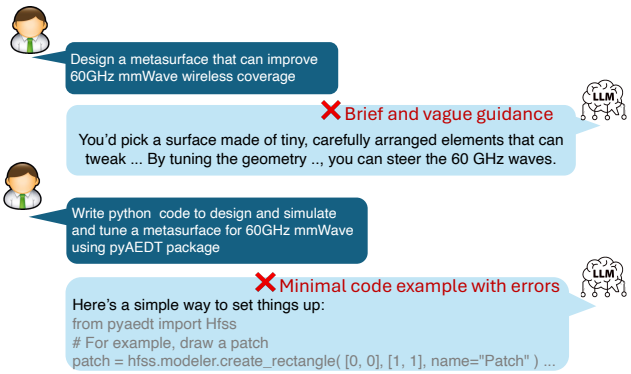


Figure 13: Example LLM responses without AutoRF. Without our system design, ChatGPT does not provide helpful answers for metasurface design.

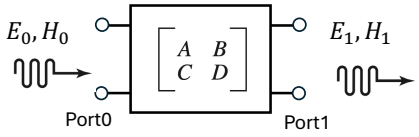


Figure 14: Definition of transmission (ABCD) matrix.

A Appendix

A.1 Circuit Model Theory

We consider the circuit model as a cascade of two-port microwave circuit networks, following microwave network analysis theory [55]. Each block, as a two-port network, can be characterized with a ABCD matrix, which is defined by the ratios of total electronic and magnetic fields at the two ports (Figure 14):

$$\begin{bmatrix} E_0 \\ H_0 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E_1 \\ H_1 \end{bmatrix} \tag{1}$$

We take Figure 4 (b) as an example, which includes 3 two-port networks, each block with ports marked as circles. We model the metasurface pattern as a shunt circuit component, and model the substrate or air gap as a transmission line, as shown in Figure 4 (c), following common practice for circuit modeling of metasurfaces [53, 72].

The ABCD matrices for the shunt circuit component and transmission line are well-understood [55]. Therefore, we have the ABCD matrix for each layer of surface pattern:

$$\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ Y_1 & 1 \end{bmatrix} \tag{2}$$

where Y_1 is the admittance of the shunt circuit component, the inverse of the surface impedance. We can characterize the specific value based on HFSS simulations for this single-layer metallic pattern.

The ABCD matrix for the substrate or air gap is

$$\begin{bmatrix} A_s & B_s \\ C_s & D_s \end{bmatrix} = \begin{bmatrix} \cos\beta_k l & jZ_s \sin\beta l \\ j\sin\beta l/Z_s & \cos\beta l \end{bmatrix} \tag{3}$$

where Z_s is the characteristic impedance of the substrate, l is the "length" (thickness) of the substrate, and β is the phase constant of the substrate decided by the dielectric constant ϵ_r and the frequency f.

The key benefit of using the ABCD matrix is that we can easily compute the transmission matrix for a cascaded circuit model by multiplying the individual matrices. This helps us model the complicated interactions among the surface patterns and the substrate or air gap. For our example here, the ABCD matrix for the entire circuit model is

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} \begin{bmatrix} A_s & B_s \\ C_s & D_s \end{bmatrix} \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix} \tag{4}$$

where A_1, B_1, C_1, D_1 are the ABCD matrices for the first layer of surface pattern, A_s, B_s, C_s, D_s are the ABCD matrices for the substrate or air gap, and A_2, B_2, C_2, D_2 are the ABCD matrices for the second layer of surface pattern.

Finally, by converting the result transmission matrix to a scattering matrix (S-matrix), we can efficiently estimate key performance metrics, such as the transmission coefficient (S21) and reflection coefficient (S11), as

$$S_{11} = \frac{A + \frac{B}{Z_0} - C\,Z_0 - D}{A + \frac{B}{Z_0} + C\,Z_0 + D},$$

$$S_{21} = \frac{2}{A + \frac{B}{Z_0} + C\,Z_0 + D},$$

which includes both power and phase information.

We also support modeling the polarization information, by extending the theory above. Specifically, both the ABCD matrix and S-matrix are extended to 4x4 matrices, where the first two rows/columns are for the horizontal polarization and the last two rows/columns are for the vertical polarization.