

# Deep-Learning Estimation of Complex Reverberant Wave Fields with a Programmable Metasurface

Benjamin W. Frazier<sup>1,2,3,\*</sup>, Thomas M. Antonsen Jr.<sup>1,2,4</sup>, Steven M. Anlage<sup>2,4,5</sup> and Edward Ott<sup>1,2,4</sup>

<sup>1</sup>*Institute for Research in Electronics and Applied Physics, University of Maryland, College Park, Maryland 20742, USA*

<sup>2</sup>*Department of Electrical and Computer Engineering, University of Maryland, College Park, Maryland 20742, USA*

<sup>3</sup>*Johns Hopkins University Applied Physics Laboratory, Laurel, Maryland 20723, USA*

<sup>4</sup>*Department of Physics, University of Maryland, College Park, Maryland 20742, USA*

<sup>5</sup>*Quantum Materials Center, University of Maryland, College Park, Maryland 20742, USA*



(Received 16 August 2021; revised 10 December 2021; accepted 11 January 2022; published 9 February 2022)

Electromagnetic environments are becoming increasingly complex and congested, creating a growing challenge for systems that rely on electromagnetic waves for communication, sensing, or imaging, particularly in reverberating environments. The use of programmable metasurfaces provides a potential means of directing waves to optimize wireless channels on demand, ensuring reliable operation and protecting sensitive electronic components. Here we introduce a technique that combines a deep-learning network with a binary programmable metasurface to shape waves in complex reverberant electromagnetic environments, in particular ones where there is no direct line of sight. We apply this technique for wavefront reconstruction and control, and accurately determine metasurface configurations based on measured system scattering responses in a chaotic microwave cavity. The state of the metasurface that realizes desired electromagnetic wave field distribution properties was successfully determined even in cases previously unseen by the deep-learning algorithm. Our technique is enabled by the reverberant nature of the cavity, and is effective with a metasurface that covers only approximately 1.5% of the total cavity surface area.

DOI: [10.1103/PhysRevApplied.17.024027](https://doi.org/10.1103/PhysRevApplied.17.024027)

## I. INTRODUCTION

Highly scattering environments scramble electromagnetic waves, producing interference among the multiple paths between source and receiver. The resulting spatiotemporal fluctuations can seriously degrade imaging, sensing, and communication systems at microwave and optical wavelengths, disrupting operation or even damaging sensitive components. Additional emissions in these environments, whether from unintentional coupling between components or from an intentional electromagnetic attack, can have serious consequences. Future smart radio environments are envisioned to handle such dynamic conditions, adapting on the fly to optimize a given wireless channel through a spatial light modulator (SLM) [1–3]. Intelligently controlling wave fields in the presence of multipath reflections is therefore a critical factor for enabling smart radio environments. In addition, an intelligent and self-adaptive approach will benefit applications such as micromanipulation of objects in complex scattering environments [4], and time-reversal mirrors that can

selectively focus a wavefront or enhance communication system performance [5,6]. A necessary step along this path is to identify approaches for wavefront reconstruction (i.e., determining the configuration of the SLM that accurately produces a given scattering response) that work in complicated scattering environments.

In optics, SLMs have been used to control waves under strong scattering conditions for some time. Applications range from focusing through general disordered media [7,8] to sophisticated biomedical imaging instruments that fall under the umbrella of adaptive optics [9,10]. In the last several years, spatial microwave modulators in the form of programmable metasurfaces have also become widely available. Programmable electromagnetic metasurfaces are metamaterial sheets that can modify their local surface impedance over unit cells (meta-atoms) that have a sub-wavelength characteristic size. They have emerged as powerful tools for shaping waves inside complex microwave cavities [11–19].

Metasurfaces are not limited to shaping only electromagnetic waves. In seismology, control over surface acoustic waves has been demonstrated using metasurfaces made of elastic metamaterials for Love waves

\*frazibw1@umd.edu

(horizontally polarized) [20] and Rayleigh waves (containing both longitudinal and transverse motion) [21]. In the case of quantum waves, a metasurface created from an array of trapped neutral atoms was used to manipulate light at the quantum level [22]. While the underlying physics of these metasurfaces is vastly different, the overall operation and process of wave interaction is essentially the same, implying that strategies for wavefront shaping in one domain can be readily adapted to another.

Wavefront shaping techniques with metasurfaces have been well studied; however, control in complex reverberating environments still relies on simple, online brute-force optimization methods. While these approaches work, they require a large number of iterations to reach convergence, are rarely guaranteed to achieve a global minimum, and can produce undesirable scattering configurations through the intermediate steps of the optimization process. Wavefront reconstruction, or estimating the wavefront on the basis of metasurface commands, is therefore a critical capability for enabling practical wavefront control applications.

In this paper we use a binary programmable metasurface to shape radio-frequency electromagnetic waves inside a chaotic microwave cavity, and present a deep-learning network that solves the wavefront reconstruction problem, enabling real-time operation once trained. We show that the deep-learning network achieves an accuracy exceeding 99%. This high success rate is achieved with a limited amount of training data, requiring the collection of far fewer sets than the number of possible combinations of metasurface commands. A conceptual view of our technique is given in Fig. 1. The metasurface is placed in a reverberant scattering environment, with a signal injected at port 1, and the resulting field measured at a specific point of interest (port 2). The environment is defined by irregular walls and inclusions and is probed by waves with wavelengths much smaller than the characteristic dimension of the enclosure.

We emphasize that our method is enabled by the use of a reverberant environment, which allows the metasurface to interact with multiple ray trajectories, often more than once. A reverberant environment provides two major capabilities that are not present in nonreverberant environments. First, the ability to control the distribution of wave fields at arbitrarily chosen locations inside the cavity is enhanced. This allows the use of relatively small metasurfaces; for example, in our configuration, the metasurface covers only approximately 1.5% of the total surface area of the cavity. Second, the requirement on establishing a line-of-sight path between the metasurface and the ports is removed, which allows the location of the metasurface to be arbitrarily chosen, increasing the flexibility and versatility of the approach. We anticipate that realization of this concept will help usher in an alternative era of smart radio environments, as well as allow on-demand creation of

microwave cold spots to protect sensitive electronic components and coherent perfect absorption states for wireless power transfer.

## II. WAVEFRONT CONTROL IN REVERBERANT ENVIRONMENTS

Microwave experiments show that programmable metasurfaces can provide fine control over the scattering parameters of a cavity, with the most recent work demonstrating perfect absorption [16] and coherent perfect absorption [18,19] states inside the cavity. The relationship between metasurface commands and cavity scattering parameter responses is extremely complicated (there are  $10^{18}$  possible configurations of the metasurface in our case). Therefore, optimization of the metasurface is typically handled through brute-force trial and error or stochastic search algorithms [18,23,24]. As discussed in Appendix A, rapid and accurate wavefront reconstruction techniques that solve the inverse problem between measurements and metasurface commands are necessary to realize practical intelligent wavefront shaping systems. Conventional methods fall apart in complex scattering environments with binary metasurfaces; however, the inherent complexity makes it an ideal place to utilize deep learning. Ma *et al.* explore the use of deep-learning networks with wave chaotic systems, demonstrating the ability to successfully distinguish between different types of wave chaotic cavities through the measured  $S$ -parameters [25]. We now tackle a more difficult problem, identifying a set of metasurface commands required to achieve a specific wave scattering condition, even for cases where that set of commands has not been previously encountered.

Deep learning has been successfully used to design metasurfaces for wavefront shaping applications in both the photonic and microwave domains [26–37]. However, most of the publications so far have focused on designing and arranging the individual unit cells of the metasurface for static use cases. Active deep-learning approaches with programmable metasurfaces have been demonstrated for microwave imaging applications [38–43]. Li *et al.* use a two-bit coding metasurface to generate radiation patterns for a machine learning algorithm that detects and classifies human movement [38,39]. Del Hougne *et al.* start with a pair of metasurfaces as a transmitter and receiver to feed a dense neural network that detects and classifies objects in a learned integrated sensing paradigm [40,41]. Further research by this group used a dense neural network to classify the position of a scattering object inside a complex cavity with a metasurface acting as a coded aperture [42]; this work was recently extended to predict a continuous position with sub-wavelength precision [43].

These examples demonstrate how a programmable metasurface can enhance the processing power of a deep-learning network for microwave imaging, but they do not

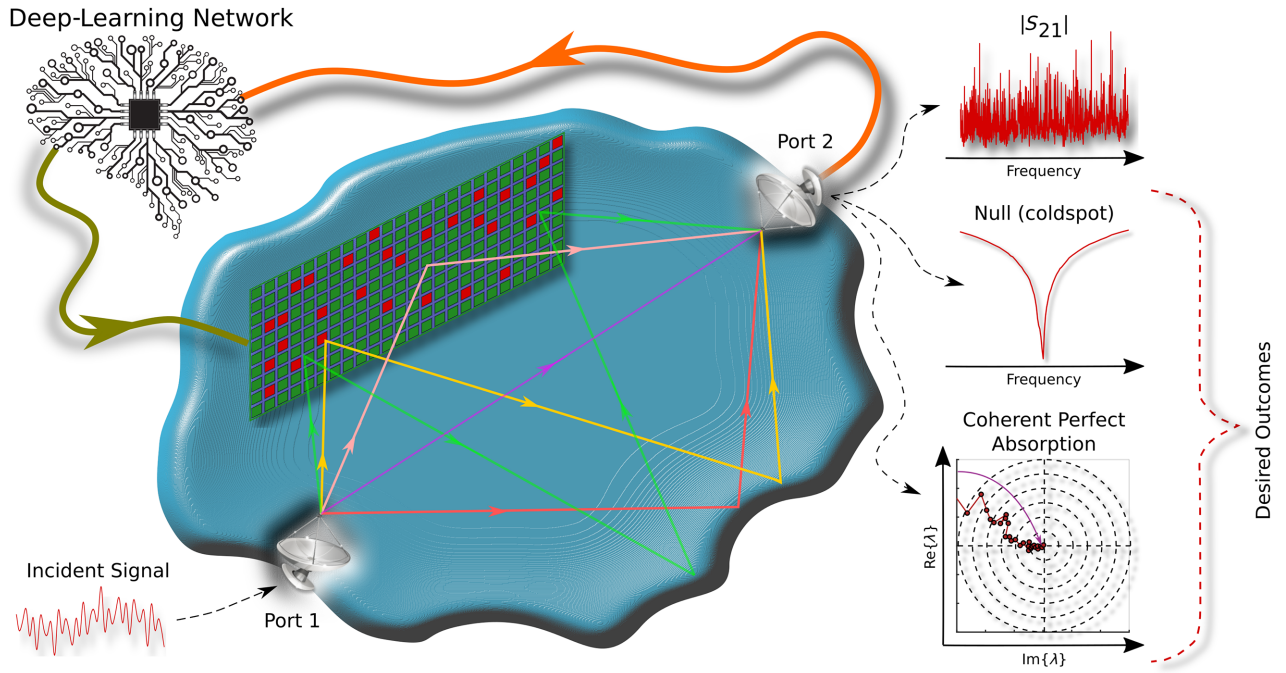


FIG. 1. Conceptual view of a deep-learning-enabled programmable metasurface in a complex electromagnetic environment. Constructive and destructive interference between multiple propagation paths in a reverberating environment induces randomness in the scattering parameters and scrambles electromagnetic waves that are injected at port 1. A reconfigurable metasurface is used to tune the interference to create cold spots for protection of sensitive electronic components, realize coherent perfect absorption states for long-range wireless power transfer, or unscramble the output fields to enable smart radio environments. The metasurface, along with a sensing antenna at port 2, is coupled with a deep-learning network that provides control. Measurements are used as training data, enabling the network to determine the control settings of the metasurface, and allowing the system to adapt to changing environmental conditions on the fly. The metasurface is shown here as large relative to the cavity. In our configuration, however, the metasurface is much smaller, covering only approximately 1.5% of the total surface area of the cavity.

leverage the deep-learning network for wavefront reconstruction. This is a key component of intelligent wavefront shaping, which has so far been an underexplored area of research. Qian *et al.* use a simple dense network to enable cloaking of an object [44], while Shan *et al.* use a two-dimensional (2D) convolutional network to optimize the steering of multiple beams [45]. Both cases utilize an idealized testing environment inside an anechoic chamber, where multipath reflections from the environment are intentionally excluded. In addition, both cases are built around a propagation path with a direct reflection off the metasurface, which means that the metasurface interacts with virtually all ray trajectories from the source to the receiver.

As discussed in Appendix B, a single propagation path eliminates redundancies from persistent short orbits [46,47], reducing the measured correlation between metasurface configurations. These cases can be treated with more traditional system identification techniques or simple neural network models. When the metasurface is placed inside a complex reverberant scattering volume [42,43], determining the relationship between metasurface commands and scattering responses becomes substantially

more difficult due to the presence of multiple scattering paths. A reverberant scattering system is qualitatively different from an open system, and is characterized by extreme sensitivity to initial conditions [48,49]. This means accurate wavefront reconstruction must account for chaotic behavior and be sensitive to small environmental changes, as well as handle nonnegligible large amplitude signal spikes that include phenomena such as rogue waves [50]. This difficulty is further compounded as we wish to optimize the metasurface response over a wide bandwidth or even over multiple separated bandwidths simultaneously.

The reverberating nature of the cavity enables operation with a smaller metasurface than would be possible in a nonreverberating environment. Longer reverberation times (lower cavity losses) mean that the rays will survive longer in the cavity, resulting in more reflections from scattering objects and more ray trajectories that interact with the metasurface, often multiple times. Longer reverberation times then provide the metasurface more flexibility in controlling constructive and destructive interference at the ports, allowing for larger relative changes when toggling metasurface states. We demonstrate this to be the



case and show that the performance of the deep-learning network degrades as the losses in the cavity increase because the metasurface has a smaller relative impact on the  $S$ -parameters. This is another distinction between a reverberant environment and an open one, where environmental losses only impact the signal magnitude through absorption.

We further show that our trained network can successfully determine the metasurface configuration from the measured scattering response in the cavity several days after collection of training data. Measured  $S_{21}$  responses with the same initial conditions inside a chaotic cavity will change over time, a phenomenon known as scattering fidelity decay [51–53]. Scattering fidelity decay is a property of wave chaotic system, and its sensitivity to boundary conditions and scattering environment. This is in contrast to ray chaos and the sensitivity of bouncing ray trajectories to initial conditions in billiards. This decay means that any deep-learning system that learns scattering responses inside a chaotic cavity will require periodic retraining. As discussed in Sec. VI, the fact that we are still able to determine the metasurface configuration accurately after several days means our technique is operationally useful, as it can function at a high level of accuracy for a long period of time before requiring retraining. Our approach is robust and highly accurate in determining metasurface commands from measured cavity  $S_{21}$  spectra, providing an enabling capability for intelligent wavefront shaping applications. In addition, our method is general enough to operate in arbitrary complex scattering systems and does not require a specifically engineered environment.

Our technique is achieved through the development and combination of four major aspects: (1) adaptive configuration of the metasurface unit cells by binning elements together to dynamically alter the relative size of the elements; (2) representation of the complex system  $S$ -parameters in a pseudo-2D “image” to promote extraction of features that are correlated over both local and global frequencies; (3) complex-valued deep-learning layers to exploit both phase and amplitude information, accelerating training and improving the accuracy when applied to complex scattering environments; and (4) introduction of the Terrapin Module to parallelize the deep-learning network, promoting sparse feature representation and improving training robustness.

### III. EXPERIMENTAL CONFIGURATION

The complex ray chaotic cavity used for experimentation is in the same configuration as used in our previous work [18]. It has a volume of approximately  $0.76 \text{ m}^3$  and includes three ports, with ports 1 and 3 used to inject signals and port 2 used for scoring. The cavity configuration and experimental schematic are shown in Fig. 2. Each port is connected to an ultra wideband antenna, and the nominal

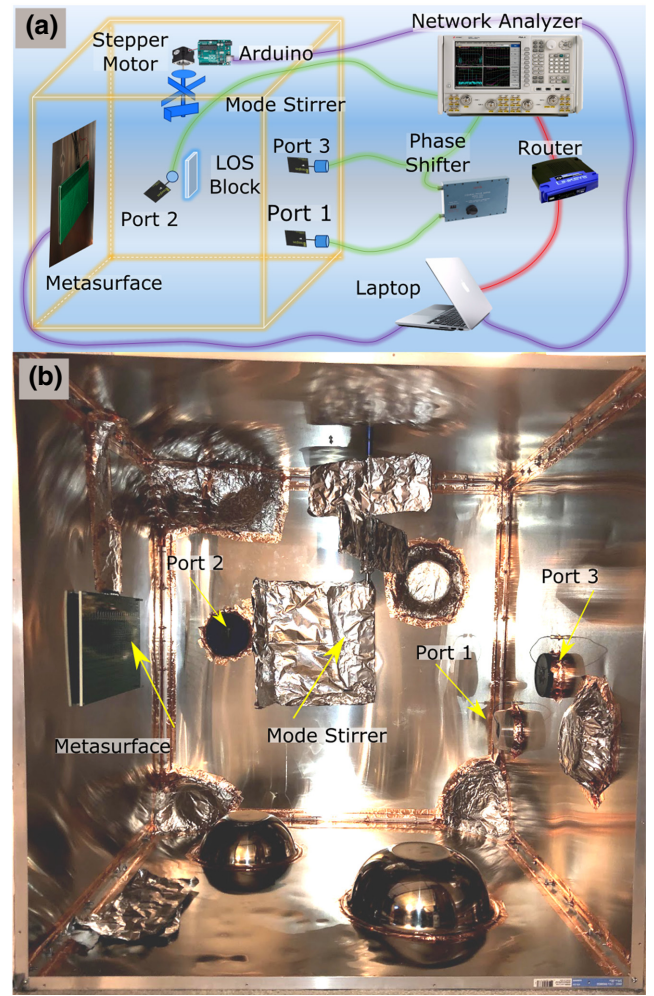


FIG. 2. Cavity configuration. (a) Experimental schematic of the cavity, showing the metasurface installed on the cavity walls, the locations of the three ports, the line-of-sight (LOS) block to prevent direct transmission between port 2 and ports 1 and 3, and the mode stirrer that is controlled by a stepper motor through an Arduino. Also shown are the network analyzer, phase shifter, control laptop, and router. (b) Photograph of the interior of the cavity showing the components from the schematic as well as the irregular scatterers that were installed on the cavity walls.

measurement window is 3–4 GHz. An Arduino controlled mechanical mode stirrer is included to allow collection of an ensemble of cavity realizations. The experimental setup is controlled by a laptop, with an Agilent N5242A network analyzer used to measure cavity  $S$ -parameters. To reduce the cavity symmetry, irregular scattering objects were installed on the walls. Additional details on the cavity are provided in Appendix C.

Experiments were carried out over several months, with the cavity placed in a heated and air-conditioned basement. While there were many commercial devices in the vicinity that emitted signals within the measurement window, we found the cavity to be well isolated. When not actively in



use, such as in between experimental runs, the metasurface was powered off.

The metasurface installed on an interior wall was fabricated by the Johns Hopkins University Applied Physics Laboratory. It is designed to operate in the frequency range of 3–4 GHz, and contains 240 binary meta-atoms ( $LC$  resonators) arranged in a rectangular grid of  $10 \times 24$  elements. Each element has a characteristic length of approximately  $\lambda/6$  and is switched by a GaAs transistor amplifier to one of two states (0 or 1), changing the phase of the reflection coefficient by approximately  $180^\circ$  [54]. The metasurface covers a small region of the interior surface area of the cavity (1.5%), and intercepts only a limited number of rays.

The goal of our deep-learning network is to enable wavefront reconstruction inside a complex cavity. The network will accept given  $S_{21}$  spectra from 3–4 GHz and accurately determine the metasurface commands necessary to closely realize that specific scattering response. The relatively small size of the metasurface and its unit cells leads to high correlation between system scattering responses with minor changes in metasurface commands, which means the inverse problem is ill-posed. Deng *et al.* recently introduced a neural-adjoint approach for solving the ill-posed inverse problem of designing unit cell geometries to match specified absorption spectra [55]. In this case, a fully connected deep-learning network was used to model the forward problem, acting as a Green's function to predict the spectrum from a given design. The inverse problem was then solved iteratively, driving the design along an estimated gradient towards the optimal solution. As discussed in Appendix A, gradient methods work best for a continuous or near continuous solution space rather than a binary one such as ours; however, the adjoint method from [55] can be adapted into a reinforcement learning approach [56]. In addition, inside a chaotic reverberating environment, the spectra will have more structure, resulting in higher-frequency oscillations or local features that must also be learned. Therefore, we require a different deep-learning approach.

#### IV. DEEP-LEARNING NETWORK DESIGN

The metasurface has  $2^{240}$  possible combinations of commands. To reduce the dimensionality, we introduced the concept of binning neighboring unit cells as discussed in Appendix D. Groupings consisting of  $2 \times 2$ ,  $3 \times 3$ , and  $5 \times 4$  elements were chosen. Binning is an important capability that allows us to adapt the size of an effective element to the underlying scattering system. This is one of the major contributions of our work.

The primary limitation of our approach is that we are not guaranteed to be able to generate any arbitrary  $S_{21}$  response, as a configuration of the metasurface that produces that response does not necessarily exist. The small

size of the metasurface relative to the cavity limits its ability to interact with all possible ray trajectories, emphasizing the importance of a binning capability to adapt the effective pixel size to the environment. This limitation is therefore a function of the system configuration, and not the deep-learning network. The small relative size of the metasurface does represent a realistic configuration for practical smart radio environments, however.

An important step for deep learning is preparation of the measured data. The goal here is to represent the data in a basis set that can be ingested by the neural network architecture. The raw data consist of  $M$  sets of complex two-port  $S$ -parameter values, each containing 32 001 points measured over a 3–4 GHz window. We are interested in the relationship between metasurface commands and transmission between the ports, so we select  $S_{21}$  as the primary variable of interest. The measured data contain local and global correlations, both of which must be captured by the deep-learning network. We can exploit the local correlations with one-dimensional (1D) convolutional neural network (CNN) layers, but would like the individual windows to cover a smaller bandwidth. Our previous work showed diminishing returns for optimization over bandwidths greater than 10 MHz [18], so 10 MHz provides a reasonable limit for the local window size. We therefore extract the complex  $S_{21}$  in 10 MHz frequency windows at 100 distributed center frequencies to provide 100 feature vectors containing 321 points each. A representative data set is shown in Fig. 3(a), with only 50 feature vectors used for illustration. The data are organized into a three-dimensional structure of  $M$  sets of data  $\times F$  local frequencies  $\times N$  features, or  $10\,000 \times 321 \times 100$  for the  $2 \times 2$  binning configuration. Each data set takes on a pseudo-2D format with a  $321 \times 100$  pixel “image” as shown in Fig. 3(b). Local features in the 10 MHz frequency windows (over the  $F$  dimension) will be extracted by 1D CNN layers and global features (over the  $N$  dimension) will be extracted by the overall deep-learning network architecture. The overall architecture then acts as a dense or fully connected layer from the perspective of the global features. The pseudo-2D format and its ability to capture both long-range and short-range correlations in frequency provides the second major aspect of our approach. Additional details on data collection are provided in Appendix E.

The output values of the deep-learning network (equal in number to the number of binned metasurface elements) are floating-point numbers rather than binary numbers and can be interpreted as the probability that a given element in the metasurface is active (set to 1). The determined commands are then found by rounding the outputs to either a 1 or a 0. Inspection of the raw (unrounded) outputs allows us to assess how correct the deep-learning network was, or how confident the network was in the result. A discussion of the different types of neural network layers used is provided in Appendices F and G.

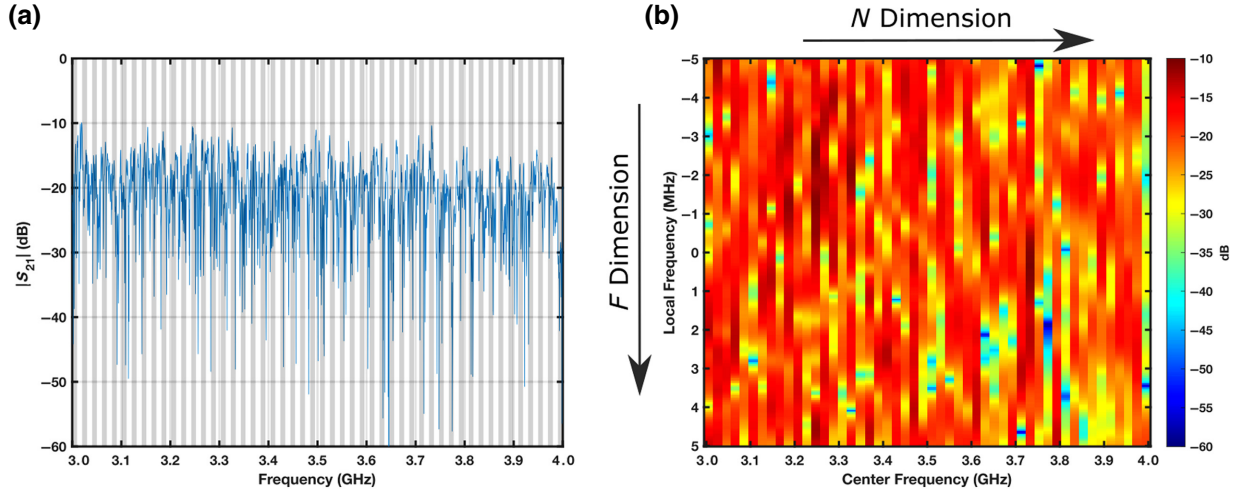


FIG. 3. Data preparation example. The deep-learning networks use complex amplitudes; however, only the magnitude is shown here for illustrative purposes. (a) Raw  $|S_{21}|$  data vs frequency showing 50 local windows highlighted in gray. The actual data preparation uses 100 local windows; only 50 are shown here for clarity. The data were collected over a 1 GHz measurement window with 32 001 points and each local window (highlighted in gray) has a bandwidth of 10 MHz or 321 points. (b) Extracted  $|S_{21}|$  data (in log scale) in a pseudo-2D format as a  $321 \times 50$  pixel “image.” The data are represented as center frequency (over the full 1 GHz measurement window or  $N$  dimension) vs local frequency (over the 10 MHz local window or  $F$  dimension). The deep-learning network will use 1D convolutions to extract features in the 10 MHz local frequency windows ( $y$  axis) and use the relationships between convolutional filters to capture global correlations over the full 1 GHz measurement window ( $x$  axis).

As described in Appendix H, the training data sets are randomly shuffled and then split into 75% training data and 25% validation data. At each step (epoch), validation of the trained model is performed by testing the model with a new set of data not present in the training set.

Complex-valued multiplicative layers have been used to invert propagation through multimode fibers [57,58], but have not previously been used for wavefront reconstruction. Unfortunately, as discussed in Appendix I, there are no officially supported complex-valued modules in any of the major deep-learning frameworks. Multiplicative layers are straightforward to implement, but more complicated modules, such as convolutional layers, are not. For the research described here, we leverage the open source complexPyTorch library [59] as the basis for our complex-valued network layers.

The added complexity resulting from placing the metasurface inside a chaotic cavity requires a correspondingly complicated deep-learning network to extract the relevant features. Rather than only designing progressively more intricate network topologies, we can also introduce complex-valued layers [60], which serve as the third major aspect of our approach. The wave scattering phenomenon is fundamentally complex-valued, so using complex-valued layers allows the network to exploit both phase and amplitude information and better match the underlying physical system.

To demonstrate the impact of utilizing complex-valued network layers, we performed an experiment comparing three different architectures: (1) a complex-valued network

that processes the complex-valued  $S$ -parameters and converts the result to a magnitude at the end of the network [Fig. 4(b)]; (2) a hybrid network that processes the real and imaginary components of the  $S$ -parameters independently and then combines them through a root-sum-square operation at the end of the network [Fig. 4(c)]; and (3) a real-valued network that operates on the magnitude of the  $S$ -parameters [Fig. 4(d)].

Ten separate training runs were performed with each architecture on the same set of input data. Panels (e)–(g) of Fig. 4 show the evolution of the accuracy of network over the validation data set and demonstrate that the complex-valued network layers are ultimately able to achieve a higher accuracy than the hybrid or real-valued network layers and exhibit more stable training behavior. Training instability manifests as wide variations in the accuracy response, resulting in certain runs not effectively learning until much later (several cases for the real-valued network are still learning after 500 epochs). It indicates that the hybrid and real-valued networks are much more sensitive to initial weights or ordering of the data sets after shuffling than the complex-valued networks.

As described in Appendices J and K, the  $5 \times 4$  binning case performed extremely well using a straightforward sequential CNN architecture. After training, we were able to accurately realize 100% of the target responses over both the training and validation sets. Unfortunately, the purely sequential architecture of the network did not work as well for the  $3 \times 3$  binning configuration (see Appendix M). The increased complexity implies that we need a more

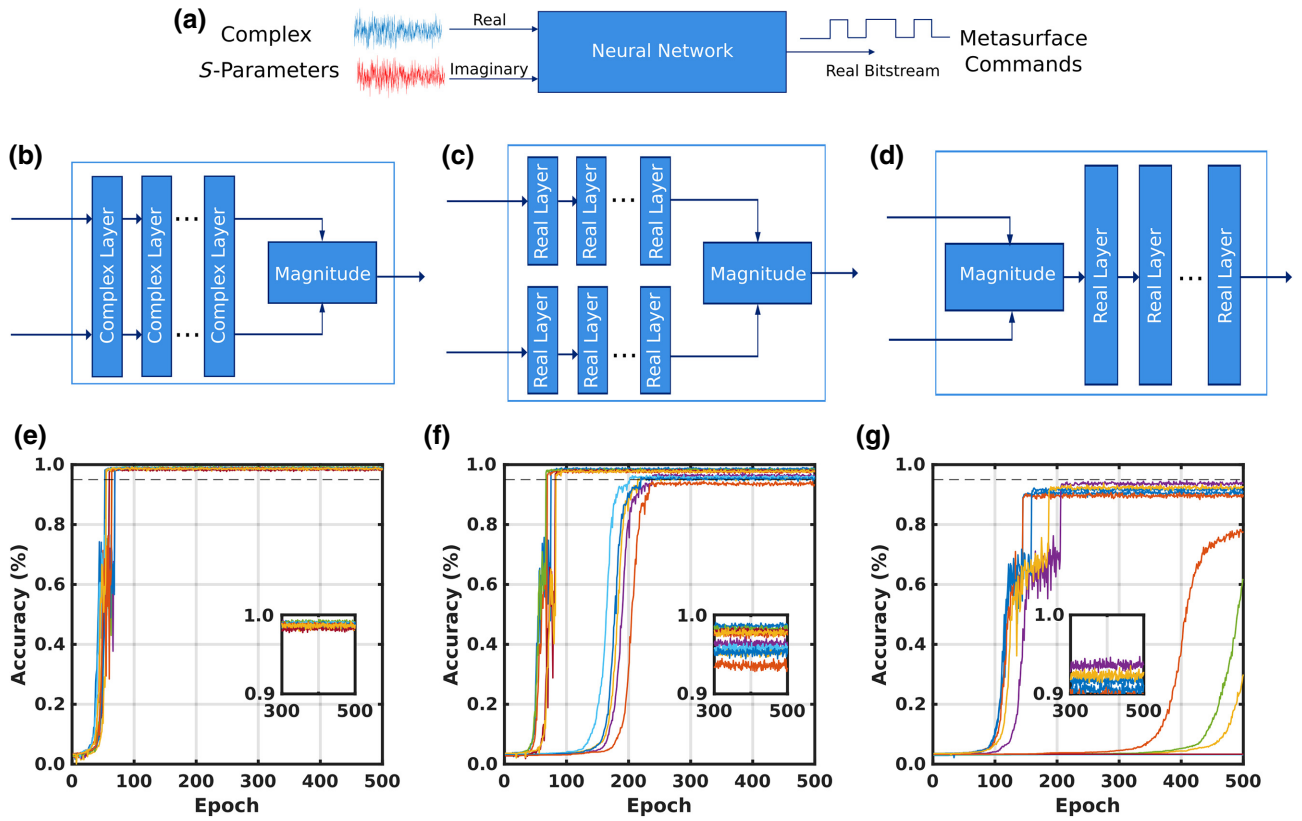


FIG. 4. Training performance for different architectures. (a) Generic network architecture. The real and imaginary components of the complex  $S$ -parameters are ingested by the network, with a real-valued bitstream provided at the output for metasurface commands. (b) Schematic architecture for the complex-valued network showing that the magnitude operation only occurs at the end of the network. (c) Schematic architecture for the hybrid network showing that the real and imaginary components are processed independently and combined at the end. (d) Schematic architecture for the real-valued network showing that the magnitude operation occurs at the beginning of the network. (e)–(g) Evolution of the accuracy over the validation data set for the various architectures with a mean accuracy of 99.2% for the complex valued network (e), 97.4% for the hybrid network (f), and 91.4% for the real-valued network (g) cases that successfully trained. The dashed line shows 95% accuracy, and the insets show closeups of the accuracy evolution from epochs 300–500 in the range of 90%–100%. This figure demonstrates that the complex-valued network achieves better overall accuracy with more stable training performance than the other architectures.

complex network, so we turned to approaches successfully used in modern image classification, specifically inception modules [61,62]. As discussed in Appendix L, we modified the general architecture to perform 1D convolutions over the 10 MHz local frequency windows. The 1D convolutional filters then extract local features over the 10 MHz windows, while the relationship between the filters acts as a dense or fully connected layer, extracting global features over the full 1 GHz measurement window. The final version, which we refer to as a “Terrapin Module,” is shown schematically in Fig. 13 and provides the fourth and final major technical contribution of our approach.

With a deep-learning network containing four Terrapin Modules in series, we were able to get excellent performance for the  $3 \times 3$  binning configuration with only 4000 sets of training data, as discussed in Appendix M. The  $2 \times 2$  configuration required 10 000 sets of data for a similar level of performance (see Appendix N). The smaller

effective elements in this configuration produce responses with a larger degree of correlation. Thus, the network require more data in order to learn and distinguish the more subtle relationships between metasurface command configurations and scattering matrix responses,  $S_{21}(f)$ .

Training is performed in a parallelized fashion over all the training data at once, for example, for our computational resources, taking approximately 4 h to collect a sufficient set of data and train the deep-learning network. Testing, however, is performed on single shot measurements and takes less than 1 s to measure the  $S_{21}$  response and make a determination of the metasurface commands, enabling real-time operation. In contrast, the iterative approach in our earlier work [18] required approximately 300 measurements to converge to a desired configuration, taking approximately 10 min to reach the answer for each configuration. Online iterative optimization does not require training, but may produce



undesirable configurations due to the randomly applied perturbations. When time is available for offline training, the deep-learning approach is preferred.

## V. RESULTS

The primary objective of this work is to demonstrate that deep wavefront shaping is a viable technique for wavefront reconstruction inside complex scattering environments, enabling intelligent wavefront shaping in a chaotic cavity. In this section, we shown how our deep-learning approach accurately determines metasurface commands from measured cavity scattering parameters.

Training results for the  $5 \times 4$  and  $3 \times 3$  binning configurations are provided in Appendices K and M, while training results for the  $2 \times 2$  binning configuration are provided in Appendix N and shown in Fig. 5. The training data consist of 10 000 random realizations of metasurface commands, representing an extremely small fraction of the  $1.2 \times 10^{18}$  possible configurations. The data were split into 7500 sets for training and 2500 sets for validation to ensure the validation process is unbiased. The evolution of the loss function is shown in Fig. 5(a) and the evolution of the accuracy is shown in Fig. 5(b). The loss function was chosen as the mean absolute error, or the  $\mathcal{L}_1$  norm, between determined and actual metasurface commands. Accuracy is the fraction of sets that was determined without error and provides a more conservative estimate of performance. The variation around epoch 45 is due to choosing an aggressive initial learning rate, and the impact of reducing the learning rate on a plateau can be seen at epoch 55. These panels demonstrate that we were able to obtain high accuracy and a small loss function for both the training and validation data sets.

The trained model did not have perfect accuracy but was able to determine 2443 out of 2500 sets in the validation data without error, for an accuracy of 97.7%. One set had 2 errors and 56 sets had a single error, as shown in Fig. 5(c). A comparison of the determined and true commands is shown in Figs. 5(d), 5(e), 5(g), and 5(h). These panels show that for the worst case set with two errors, the network was not highly confident in the results as the erroneous determined command probabilities were 0.41 and 0.73. Finally, example scattering responses are shown in Figs. 5(f) and 5(i), which demonstrate both the complexity of the  $S_{21}$  responses and the fact that the difference between the measured and predicted responses is generally at least 20 dB lower than the signal magnitudes themselves.

To further validate the trained deep-learning network, we adopted the online, closed loop configuration as shown in Fig. 6(a). Commands were applied to the metasurface and the  $S_{21}$  response was measured and then passed through the trained deep-learning model to verify accuracy. This provides a third set of data not seen during training (or the initial validation). When the

deep-learning-determined commands had errors, the determined commands were applied to the metasurface so that the difference in  $S_{21}$  responses could be computed. We define the difference,  $\Delta S_{21}$ , between two measured  $S_{21}$  responses,  $S_{21}^a$  and  $S_{21}^b$  through the  $\mathcal{L}_2$  (Euclidean) norm,  $\|S_{21}(f)\|_2 = \sqrt{\sum_f |S_{21}(f)|^2}$ . The summation is taken over the full measured frequency range (3–4 GHz) and  $\Delta S_{21}$  is defined as

$$\Delta S_{21} = 2 \frac{\|S_{21}^a(f) - S_{21}^b(f)\|_2}{\|S_{21}^0(f)\|_2 + \|S_{21}^1(f)\|_2}. \quad (1)$$

The normalization factor here is determined by the average of the  $\mathcal{L}_2$  norms of the active commands (all 1s),  $S_{21}^1$ , and the inactive commands (all 0s),  $S_{21}^0$ . To understand how  $\Delta S_{21}$  depends on the difference between commands, we first identified the minimum Hamming distance for each of the 10 000 sets in the training data. The Hamming distance is simply the number of elements that are different between two sets of commands. It is a useful metric for comparing command sets, but does not include scaling or correlation based on position; in some cases, the impact of toggling an element in the center may be significantly different than the impact of toggling an element on the edge of the metasurface.

The smallest Hamming distance between the training data sets ranged from a single element to 19 elements (out of 60). A series of whisker box plots showing  $\Delta S_{21}$  for the various Hamming distances is shown in Fig. 6(b). The general trend shows an increase in  $\Delta S_{21}$  with an increase in the Hamming distance. While the relationship is non-linear, the dynamic range in  $\Delta S_{21}$  for Hamming distances up to one-third of the total number of elements is large, approximately an order of magnitude.

Validation of the deep-learning network in the configuration shown in Fig. 6(a) was performed periodically after collecting the training data, and the results are shown in Fig. 6(c) at 2 h, (d) at 36 h (1.5 days), and (e) at 72 h (3 days), with the metasurface powered off between each validation test. Over time the scattering environment is expected to “age” and systematic changes to the scattering environment will occur. The blue diamonds show cases where there was a single error, and the black dots show cases where there were two errors. Each online validation experiment showed approximately 95% accuracy and the resulting  $\Delta S_{21}$  for errors was small compared to the observed statistical extent of  $\Delta S_{21}$  for single-element Hamming distances. This suggests that even when the deep-learning network is unable to determine the commands completely accurately, the resulting difference in  $S_{21}$  is very small. As shown in Appendix O, the accuracy was still greater than 85% after 120 h (5 days), but dropped to approximately 65% after 9 days.

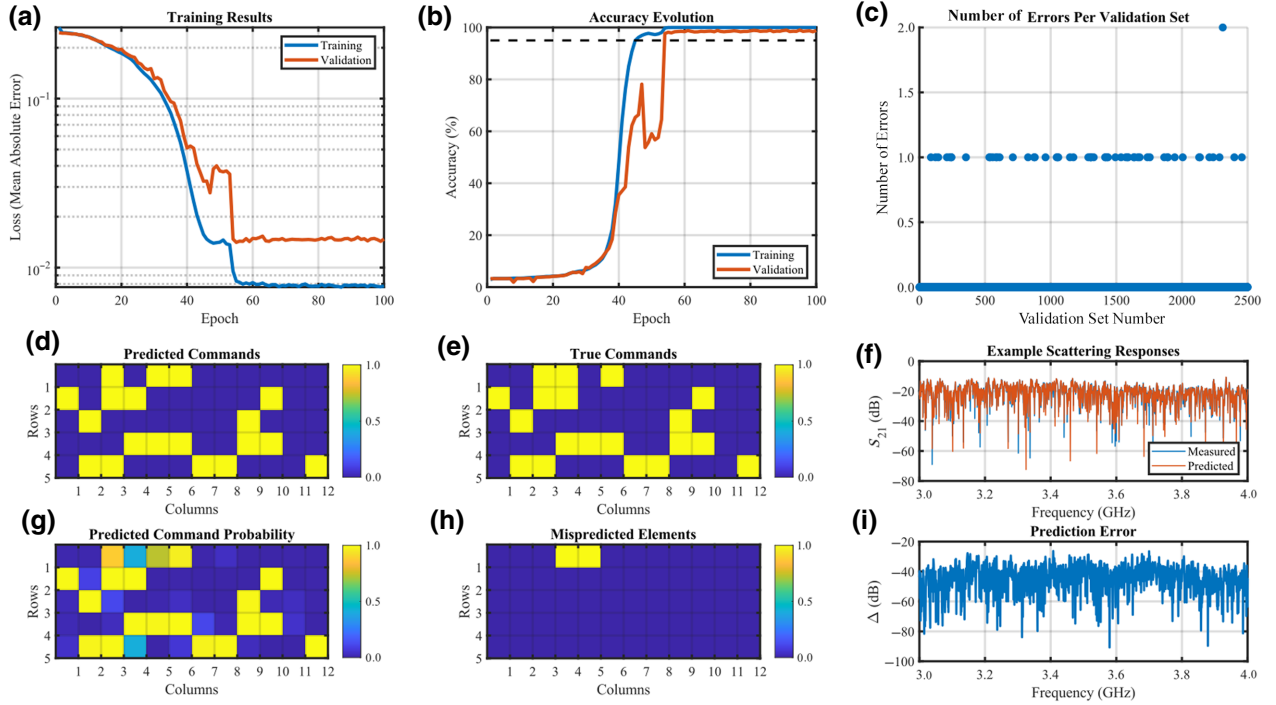


FIG. 5. Deep-learning performance with complex-valued layers for  $2 \times 2$  binning. (a) Evolution of the loss function for the training and validation sets over 100 epochs. The learning rate is reduced at epoch 55, inducing an additional reduction in the loss function. The initial learning rate was aggressive, resulting in a large variation in the validation loss function between epochs 45 and 50. (b) Evolution of the accuracy for the training and validation sets over 100 epochs; the dashed black line identifies 95% accuracy. (c) Number of errors over the validation set for the trained model, showing a total of 58 errors and 97.7% accuracy. The maximum number of errors in a single set was 2 (out of 60 elements), which occurred once. (d) Determined commands for validation set No. 2311 showing the output from the deep-learning network. (e) True commands for validation set No. 2311 showing what was actually applied to the metasurface. (f) Example scattering responses for online validation showing the measured and predicted  $S_{21}$  responses. (g) Determined command probability for validation set No. 2311, showing the raw outputs from the deep-learning network prior to rounding. This panel shows that the two elements determined incorrectly have command probabilities of 0.41 and 0.73, meaning the network was not highly confident in the result. (h) Errors or incorrectly determined commands for validation set No. 2311, showing the two elements that were determined incorrectly. (i) Prediction errors or difference between the measured and predicted  $S_{21}$  responses.

The number of errors and number of cases with more than one error increase with time, showing the “aging” effect of the cavity, which can be quantified through the concept of scattering fidelity. Scattering fidelity is the normalized correlation as a function of time between two cavity responses with the same initial conditions [63]. Because a chaotic cavity is sensitive to small changes in the boundary conditions, such as volume perturbations, the scattering fidelity will decay over time [51–53]. Loss of scattering fidelity means that the accuracy of any trained deep-learning network has a finite lifetime, so we must periodically retrain the network on new training data to maintain accuracy. In our case, we have demonstrated that the deep-learning network can determine metasurface commands with high accuracy (over 95%) for at least 72 h (3 days) after the initial training data collection, and with reasonable accuracy (over 85%) up to 120 h (5 days) after the initial training data collection. Large variations in environmental conditions, such as temperature or humidity, will introduce larger

perturbations and more rapidly degrade the scattering fidelity.

An additional set of experiments showing the performance of the deep-learning network with different cavity reverberation times is provided in Appendix P and Fig. 17.

## VI. SUMMARY AND DISCUSSION

In this paper we demonstrated the use of a deep-learning network for wavefront reconstruction, enabling intelligent wavefront shaping in complex reverberant environments. Major aspects include complex-valued deep-learning layers that exploit both phase and amplitude information, binning of the metasurface elements, a pseudo-2D data format that allows features to be extracted over both narrow and wide bandwidths, and a Terrapin Module that enhances the receptive field, providing width and depth to the network. While our configuration leverages a binary (1-bit) tunable metasurface, it can be adapted for a continuous device, such as a varactor, by replacing the output sigmoid

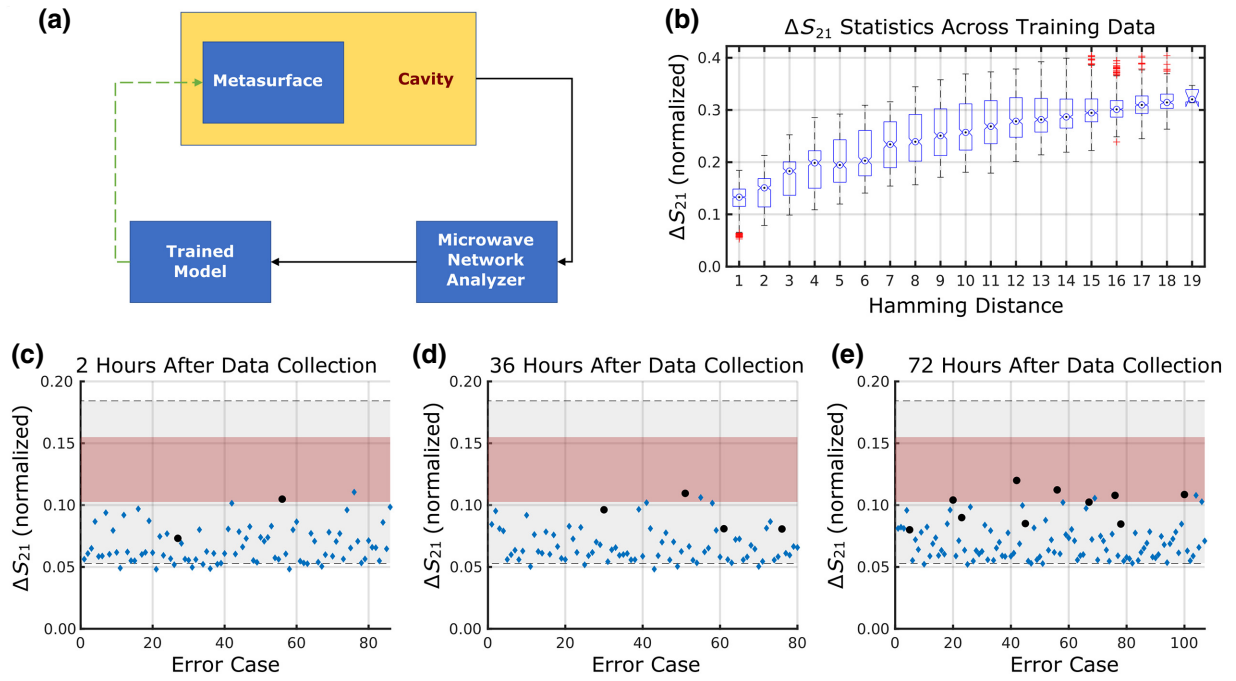


FIG. 6. Online performance verification. (a) Closed loop validation configuration. Commands were applied to the metasurface inside the cavity, the corresponding  $S_{21}$  response was measured on the PNA, and the results were passed through the trained deep-learning network. Errors for the trained model were then measured to determine the difference in  $S_{21}$ ,  $\Delta S_{21}$ , between the two command sets. (b)  $\Delta S_{21}$  statistics for the minimum Hamming distance across the 10 000 sets from the training data. Whisker plots are given for the smallest Hamming distance for each case, and show the mean value, 25th and 75th percentiles, and maximum and minimum values. (c) through (e)  $\Delta S_{21}$  for online validation sets taken a specified time after the training data were collected. The shaded regions show the extent of the single-element Hamming distance whisker box plot from panel (b). The grey region shows the full range from maximum to minimum, and the red region shows the 25th and 75th percentiles. The blue diamonds indicate cases with a single error, while the black circles indicate cases with two errors. These panels show that the  $\Delta S_{21}$  for errors is very small, and in the lower region of the statistics covered by observed cases with single-element Hamming distances. (c) Validation 2 h after collecting training data; 2000 sets of commands were tested, with 86 errors for an accuracy of 95.7%. (d) Validation 36 h after collecting training data; 2000 sets of commands were tested, with 80 errors for an accuracy of 96%. (e) Validation 72 h after collecting training data; 2000 sets of commands were tested, with 107 errors for an accuracy of 94.7%.

activation function with a rectified linear unit activation function and then discretizing the signal to the desired resolution.

One of the primary limitations of traditional deep learning is the amount of data required to train the networks. This is especially concerning in light of the fact that loss of scattering fidelity requires periodically collecting new training data. We have demonstrated the ability to train highly accurate networks with a limited amount of training data, requiring far fewer sets than the number of possible combinations of commands. We have also demonstrated that the accuracy can be maintained for a period of at least several days, and is successful with varying amounts of loss in the cavity. This indicates that successful training on a reduced amount of data is possible, provided it is sufficiently diverse. Diversity in both the metasurface commands and the measured responses is then a key aspect in setting up any potential autonomous system.

Several concerns must be addressed to enable practical fielded hardware systems. First, the sensing component

must be reduced in cost and size. The availability of software defined radio (SDR) architectures presents an ideal path here, with many inexpensive platforms readily available. Compact devices such as the bladeRF [64] can replace the bulky network analyzer. SDRs have limited instantaneous bandwidth, typically 10–20 MHz, so modifications would be required to the pseudo-2D data representation. Second, processing large deep-learning models on power-hungry graphical processing units (GPUs) may exceed the allowable footprint in terms of both cost and power consumption. Deep-learning models can be compressed by pruning and quantization [65], and the explosion of edge intelligence for connected devices in the Internet of Things is leading to more efficient embedded deep-learning systems. An example is the Jetson series of embedded GPUs from NVIDIA; the currently available TX2 series can provide up to 1.26 trillion floating-point operations per second on a 256-core GPU while consuming only 10–20 W of power [66].



In closing, we have shown that deep-learning-enabled wavefront reconstruction provides an important step towards realizing intelligent reconfigurable metasurfaces for smart radio environments. Potential applications in the domain of electromagnetics include wireless power transfer, protection of sensitive electronic components, optimization of wireless networks, micromanipulation of objects, and nonlinear time reversal. Our technique is applicable to general wave chaotic scattering systems and is not strictly limited to electromagnetic waves. Adopting this technique to control the system scattering response with metasurfaces that interact with seismic waves [20,21] or quantum waves [22] will unlock many innovative applications for wave chaotic systems.

### ACKNOWLEDGMENTS

We thank Zerotti Woods of JHU/APL for supporting many fruitful discussions on applied deep learning. Funding for this work was provided through AFOSR COE Grant No. FA9550-15-1-0171 and ONR Grant No. N000141912481.

### APPENDIX A: WAVEFRONT RECONSTRUCTION AND CONTROL

Wavefront control has a rich history and has been well studied from the perspective of adaptive optics. Conceived by Horace Babcock in 1953 [67], and realized in the 1970s [68], adaptive optics provides a method of correcting wavefront aberrations induced by propagation through random media. It has been successfully used in a diverse array of applications including astronomical imaging [69], biomedical imaging [70], high-energy laser propagation [71], free-space optical communications [72], quantum networking with satellites [73], and laser processing of materials [74].

Adaptive optics systems typically employ reflective deformable mirrors that provide mechanical phase compensation [75,76]. Conventional deformable mirrors are built with ferroelectric actuators [77] with spacing as small as 5 mm, though monolithic deformable mirrors manufactured from a block of lead magnesium niobate material can have 1 mm spacing between actuators [78]. Micro-electrical-mechanical-system mirrors using electrostatic actuation have made great strides over the past decade [79,80] and are very competitive with conventional deformable mirrors, particularly where a large actuator density is desired. Refractive liquid crystal devices have also been proposed and developed [81,82], but tend to be slow and are uncommon outside of microscopy [10] or laser processing [74].

The availability of inexpensive reconfigurable metasurfaces has driven research into a field known as wavefront shaping [7,8,12,83–85]. While there is not a strict convention or definition, adaptive optics is generally associated

with controlling distorted wavefronts for a single propagation path while wavefront shaping is generally associated with controlling (or shaping) a combination of multiple scattered wavefronts. We will adopt this convention here and refer to adaptive optics and wavefront shaping in general as wavefront control. While many applications are built around scattering systems possessing time-reversal symmetry (TRS), the presence of TRS is not a requirement for all wavefront control systems.

Conventional adaptive optics systems measure the wavefront directly and use an operator, called the reconstructor, to solve the inverse problem between measurements and control signals. Wavefront reconstruction is at the heart of any wavefront control system. The process is generally indirect, as the reconstructor evaluates the wavefront in the basis of command signals, rather than explicitly in phase. Wavefront reconstruction is a specialized area of system identification [86], and relies heavily on methods for solving inverse problems.

For a linear system, or one that can be linearized, the standard reconstructor,  $\mathbf{R}$ , is a regularized optimal Wiener filter given by [87]

$$\mathbf{R} = [\mathbf{B}^T \mathbf{C}_n^{-1} \mathbf{B} + \mathbf{W} + \mathbf{B}^T \mathbf{C}_\varphi^{-1} \mathbf{B}]^{-1} \mathbf{B}^T \mathbf{C}_n^{-1}. \quad (\text{A1})$$

Here,  $\mathbf{C}_n$  is the covariance matrix of the measurement noise,  $\mathbf{C}_n = \langle \mathbf{n}^T \mathbf{n} \rangle$ ,  $\mathbf{C}_\varphi$  is the covariance matrix of the environmental disturbance,  $\mathbf{C}_\varphi = \langle \varphi^T \varphi \rangle$ ,  $\mathbf{W}$  is a weighting/regularization matrix, and  $\mathbf{B}$  is a system configuration (geometry) matrix that relates control signals to wavefront measurements. Both  $\mathbf{C}_n$  and  $\mathbf{C}_\varphi$  are defined in sensor space, while  $\mathbf{W}$  is defined in command space.

The inverse problem is often ill-posed due to singularities or the presence of highly correlated responses with different commands [88], which effectively means we do not have enough information to solve the problem. We can add the necessary information through a process known as regularization. The weighting matrix,  $\mathbf{W}$ , in Eq. (A1) implicitly provides Tikhonov regularization, which acts as a spectral filter on the singular values [89,90].  $\mathbf{W}$  can be the identity matrix to raise all the singular values, or a projection matrix to suppress specific modes as they may either induce singularities or expend control energy in ways we wish to avoid. Using the inverse of the environmental covariance matrix,  $\mathbf{C}_\varphi^{-1}$ , preconditions the solution towards expected spatial modes with the appropriate spatial statistics.

For nonlinear systems, we can apply iterative methods to handle the reconstruction process. These are typically Krylov subspace methods such as the conjugate gradient method [89,91,92]. Regularization can be applied through Landweber iteration, where the gradient is allowed to decay with a relaxation parameter [89,91]. In this case, the iterations are performed “offline,” meaning that each iteration is evaluated numerically on the measurements.

The convergence rate is therefore only limited by the computational power we can throw at the problem.

Wavefront reconstruction is viable for some metasurface applications, but depends on being able to solve the inverse problem. Complex environments include uncertainties in determining the system configuration ( $\mathbf{B}$  matrix), and multiple reflection paths create intricate interference patterns at the antennas, producing chaotic fluctuations [93,94]. In addition, short orbits that manifest as persistent features in the ensemble [46] are not removed. This leads to a complicated relationship between metasurface commands and cavity scattering parameters. For a metasurface that is small relative to the cavity, the effective strength of the metasurface commands on the cavity scattering parameters is reduced. This results in high correlation between measurements taken with different sets of commands and creates problems for uniqueness, as many potential solutions are extremely similar. In addition, the scattering process is linear, but the relationship between metasurface commands and measured scattering parameters is not necessarily so, particularly for measurements in the temporal domain. In these extreme scattering environments, we are limited to partial information and may not be able to define the system, let alone determine the inverse. This leads to model-free control approaches that do not require knowledge of the system configuration, and bypass an explicit wavefront reconstruction step altogether.

Early metasurface control approaches used brute-force trial and error, toggling every element or combination of elements [23,24]. This guarantees that a global minimum is reached, but becomes infeasible with large numbers of elements. The simplest practical approach is an extension of iterative multidither techniques in adaptive optics [95], where, at the  $k$ th iteration, the algorithm updates a trial command vector,  $\mathbf{a}^*$ , with a small perturbation,  $\delta\mathbf{a}$ , so that

$$\mathbf{a}_{k+1}^* = \mathbf{a}_k + \delta\mathbf{a}_k. \quad (\text{A2})$$

The impact on performance is evaluated through a metric or cost function,  $J$ , that is positive and real-valued, and dependent on both the command vector and the environment,  $\mathcal{E}$ . If the cost function,  $J(\mathbf{a}_{k+1}^*, \mathcal{E})$ , is improved, the trial command vector becomes the new command vector,  $\mathbf{a}_{k+1} = \mathbf{a}_{k+1}^*$ . Otherwise, the trial command vector is rejected and a new trial command vector is generated. The iterative process continues until either a specified number of iterations,  $T$ , are performed without improving the metric, or the cost function reaches a predetermined value, at which point we claim convergence. While simple to implement and not reliant on knowledge of the system configuration, the dithering approach is by no means optimal.

Gradient-based approaches have proven extremely successful for general stochastic optimization problems [96]. In a stochastic gradient descent (SGD) optimization, the

descent is performed by taking steps along the gradient of the cost function with respect to the element command vector. The step size,  $\gamma$ , which may or may not depend on the iteration, determines how quickly the algorithm descends, and is sometimes referred to as the “learning rate.” Tuning the step size is an important aspect of SGD methods. If  $\gamma$  is too small, the algorithm will take a long time to converge and may not be able to escape a local minimum. On the other hand, if  $\gamma$  is too large, the algorithm may become unstable. The basic SGD is implemented as

$$\mathbf{a}_{k+1} = \mathbf{a}_k - \gamma_k \nabla_{\mathbf{a}_k} J(\mathbf{a}_k, \mathcal{E}). \quad (\text{A3})$$

In most cases, it is not possible to evaluate the gradient,  $\nabla_{\mathbf{a}} J$ , directly, so it must be approximated. The general approach is to apply a small perturbation to the current command vector and estimate the gradient from a one-sided or two-sided finite difference. The perturbation is applied to all elements of the command vector simultaneously (in parallel) to increase the convergence rate.

A specialty of wavefront control, known as wavefront sensorless or target-in-the-loop adaptive optics [97], leverages stochastic optimization in a sensor-agnostic manner, indirectly evaluating the wavefront through a scalar cost function,  $J$ . In target-in-the-loop approaches, the iterations are performed “online,” meaning that each iteration requires applying commands and measuring the result. The convergence rate is therefore limited by the sampling rate of the system.

Target-in-the-loop methods are not as easily analyzed through modern multivariable control theory as conventional methods, but they are highly applicable to the problem of controlling metasurfaces in complex scattering environments. In particular, stochastic parallel gradient descent (SPGD) [98,99] has enjoyed great success in target-in-the-loop adaptive optics systems. For SPGD, the gradient is estimated from a one-sided finite difference,

$$\nabla_{\mathbf{a}} J(\mathbf{a}, \mathcal{E}) \approx [J(\mathbf{a} + \delta\mathbf{a}, \mathcal{E}) - J(\mathbf{a}, \mathcal{E})] \delta\mathbf{a}^{-1}. \quad (\text{A4})$$

The cost function is arbitrarily defined, allowing SGD methods to be applied based on the specific need. It can be an image quality metric such as Strehl ratio [98] for imaging systems, signal strength for free-space optical communications [100], transmission coefficient for cold spot generation, or scattering matrix eigenvalue magnitudes for coherent perfect absorption (CPA) state realization. Specific to the problem of controlling metasurfaces in microwave wavebands, energy efficiency in terms of bits per joule is an attractive metric for wireless networks [101]. Energy efficiency optimization using a reconfigurable metasurface has been proposed and simulated using both SGD and sequential programming in an open scattering environment [102].

SGD methods work well in principle for controlling a metasurface. However, they begin to fail with coarse quantization, which limits the ability to tune both the size of the applied perturbation and the size of the step taken along the gradient. In the extreme case of a binary (1-bit) metasurface, applying a perturbation boils down to simply toggling or not toggling each element, so that for the  $n$ th element,  $\delta a_n = \{0, 1\}$ . This leads to singularities in estimating the gradient [Eq. (A4)], as well as approximation errors with driving the solution along the gradient [Eq. (A3)], since the resulting command must also be quantized to either 0 or 1. While metasurfaces can be manufactured with more bits of resolution for phase control, this increases complexity, cost, and power consumption considerably, making them less attractive for wide scale use. The capability of binary metasurfaces has been demonstrated many times; these devices can be expected to be utilized whenever power and cost are drivers for implementation.

Since gradient-based approaches are problematic with coarse quantization, dithering methods have dominated for wavefront control applications with binary tunable metasurfaces. We can modify the dithering technique in Eq. (A2) to use shaped or intelligent perturbations. When the algorithm is initialized, we do not know where the optimal commands are located with respect to the solution space. We would like to apply “larger” effective changes that induce highly diverse responses with large-scale global changes. As the algorithm proceeds, effectively moving along the gradient, we want the changes to become “smaller” and more localized. In this manner, we are able to continue the optimization process without wasting trials on global changes that are less likely to improve the specific metric of interest. Finally, once the algorithm has converged, we would like to be able to make sure we are not stuck in a local minimum.

This shaped perturbation approach was demonstrated to successfully enable generating cold spots and realizing CPA states for a binary metasurface with 240 elements [18]. In this case, the perturbations were “shaped” by changing the number of elements that were toggled (perturbed) each time the algorithm converged. The algorithm cycled through perturbations that toggled 120, 48, 24, 12, and then 6 elements, with a convergence criterion of  $T = 30$  trials. This can also be thought of as a simple policy-iteration method of reinforcement learning [56]. To ensure the solution was not stuck in a local minimum, the algorithm then entered a “single-element” phase where three trial command vectors were generated at each iteration that toggled the individual element, the nearest neighbors of that element, and the diagonal neighbors of that element. Cold spots were generated with this technique and provided 4–40 dB of suppression over a 1 GHz frequency range and CPA states were realized and verified with power absorption ratios approximately  $10^6$  [18].

While dithering allows us to provide wavefront control in some capacity, a true wavefront reconstruction method is desired. Deep learning may provide a viable approach, as it has already been successfully demonstrated for general ill-posed inverse problems [55].

## APPENDIX B: COMPLEX SCATTERING ENVIRONMENTS

Large enclosed spaces, such as offices or compartments on ships or aircraft can act as “chaotic” reverberating chambers for short-wavelength electromagnetic waves [103]. These complex scattering environments contain universal fluctuations with statistics governed by random matrix theory (RMT) [49], as well as deterministic behavior from the system-specific configuration of the ports and short orbits (i.e., prompt, direct paths) between the ports [46,104,105]. We often characterize complex scattering environments by their scattering matrix, or  $S$ -matrix, which is a frequency-dependent transfer function matrix containing the complex-valued reflection and transmission coefficients between the ports. While useful for describing the overall behavior, separating the universal and deterministic features is often difficult when working with the  $S$ -matrix.

An analytical approach known as the random coupling model (RCM), has been shown to accurately predict fluctuation statistics and allows separation of the universal and deterministic contributions in a simple additive manner [106,107]. Like RMT, the RCM leverages the random plane wave hypothesis, which asserts that the chaotic wave field is statistically equivalent to a random superposition of plane waves [108]. The RCM is characterized by a single parameter,  $\alpha$ , that describes the losses in the system, and is supported by wealth of experimental validation data with chaotic microwave cavities [47,93,94,109,110]. The behavior in large, three-dimensional enclosures has also been studied to understand these statistics and the potential impact of high-power microwave attacks [103,111].

The RCM works in the impedance domain to separate the universal contributions; conversion between impedance and scattering is handled through standard bilinear transformations [112]. In the RCM, the fluctuating impedance,  $\bar{\mathbf{Z}}$ , is defined as

$$\bar{\mathbf{Z}} = j \text{Im}\{\mathbf{Z}_r\} + \text{Re}\{\mathbf{Z}_r\}^{1/2} \xi \text{Re}\{\mathbf{Z}_r\}^{1/2}. \quad (\text{B1})$$

Here,  $\mathbf{Z}_r$  is the radiation or free-space impedance of the ports and  $\xi$  is the fluctuating or universal component, which is described by RMT [106]. For lossless systems,  $\xi$  is a Lorentzian distributed random variable. With loss, the distribution becomes much more complicated, but it is well suited to Monte Carlo simulations [113].

The RCM has been shown to apply to fading statistics in open systems, such as wireless communication paths,



as well as closed systems, such as microwave cavities [114,115]. A chaotic system is characterized by extreme sensitivity to initial conditions and is qualitatively different than an open one. In open systems, fading statistics are often modeled with empirically fit distributions [116]: the Rayleigh distribution when no line-of-sight path is present, the Rician distribution when a strong line-of-sight path is present, or the K distribution for propagation over the ocean [117]. The limiting cases of Rayleigh and Rician distributions are captured by the RCM with the  $\sigma$  parameter related to the loss parameter, as  $\alpha = (8\pi\sigma^2)^{-1}$ , and the  $\nu$  parameter equal to the magnitude of the short orbits [114,115].

Operating in a rich scattering environment presents an additional set of challenges in comparison to an open environment. In addition to the fundamental difference in the character of the fluctuations, in the semiclassical case or short wavelength limit, we can look at the behavior of ray trajectories. Specifically, we are interested in the change in ray trajectories in response to a change in the metasurface configuration. In an open system, there is a single ray (or ray bundle) that is observed by the sensor, with at most a single reflection off the metasurface. In a chaotic system, that single ray will reflect off multiple walls and obstacles in the cavity and possibly off the metasurface itself multiple times before reaching the sensor. This creates a cascading effect, so that the wavefront at the sensor is a combination of constructive and destructive interference of the multiple rays. The effect of these multiple interference paths is highly dependent on the configuration of the cavity. For an open system, a wavefront reconstruction approach is only dependent on the geometry between the sensor and the correcting device and is invariant to environmental changes (provided the disturbances remain within the dynamic range of the sensor and corrector). Wavefront reconstruction is therefore very robust for an open system. For a chaotic system, however, small environmental changes can cause a wavefront reconstruction technique that was previously successful to no longer be viable, so that being “close” is not good enough.

To build environmental models for simulations, we often need to make assumptions or approximations for simplicity or computational tractability. We also need to ensure that these assumptions are valid for the environments that are being modeled. Otherwise, the models may neglect potentially significant effects. We will outline some of the most problematic simplifying assumptions here.

The first simplifying assumption often made is that the channels are perfectly known by the transmitter (base station), so the only uncertainty in the environment is random thermal noise at the receiver(s). In complex scattering environments, there is always uncertainty, which can be significant. In addition, inside chaotic cavities, measured scattering responses with the same initial conditions will diverge over time, a phenomenon known

as scattering fidelity decay [51–53], which means that perfect knowledge of a complex scattering environment has a finite lifetime. This lifetime can be several days in controlled conditions, but is sensitive to temperature and humidity and will be reduced in scenarios such as dense urban environments. Having only partial knowledge of the system limits deterministic control approaches and encourages learning algorithms. Scattering fidelity has an impact here as well; as the scattering responses start to diverge, any machine learning algorithm will require periodic retraining.

The second simplifying assumption often made is that the equivalent channel matrix is invertible, so the inverse problem is well-posed. Complex scattering environments generally contain short orbits, or prompt direct paths, that are persistent across measurements [46,47]. These short orbits induce correlations that are difficult for simple machine learning approaches to unwrap and typically lead to ill-posed inverse problems. Excluding multipath reflections and short orbits can overestimate the performance of a given algorithm.

The third simplifying assumption often made is that all the propagation paths are assumed to have a single reflection off the metasurface, so direct line-of-sight and multipath trajectories are not included. Channel fading is then modeled with Rayleigh amplitude statistics. As stated previously, in real-world systems, strong direct line-of-sight paths induce Rician statistics and the presence of multipath reflections drives statistics that are governed by RMT [114,115]. Neglecting these statistics can lead to algorithms that are not properly tuned. The longer tails in the distributions lead to large amplitude signal spikes that can degrade imaging performance or disrupt signal processing algorithms.

Testing and verification are often done in anechoic chambers to remove the environment and capture only the impact of the reconfigurable metasurface. Anechoic chambers are very good at covering up emissions problems, which become immediately apparent in reverberation chambers [118]. A complex scattering system is reverberant in nature, so mismatches that seem negligible in anechoic chambers may be significant in real-world environments.

Finally, the metasurface is often assumed to have infinite phase resolution, so quantization effects are not included. Most commercially available metasurfaces have a single bit of control, though custom devices with 2 bits of control are becoming available [119–122], which means quantization effects are important and likely significant. In addition, as discussed in Appendix A, coarse quantization can cause gradient-based controllers to fail, so neglecting quantization effects may lead to poorly performing real-world controllers. The metasurfaces are also idealized, with identical responses across all elements. In real devices, manufacturing defects produce nonuniformities

between the phase at each element, and the metasurface may also include uncontrollable losses or gain.

Caution should be exercised when applying any of these simplifying assumptions. Otherwise, they can overestimate the performance or instill a false sense of confidence in a particular approach.

A final characteristic of complex scattering environments to describe is coherent perfect absorption (CPA). CPA is a special state of the scattering matrix where an eigenvalue is driven to 0 and the electromagnetic energy for the corresponding eigenvector is completely absorbed inside the scattering system [123,124]. While practical applications are still being developed, research has demonstrated that a high fraction of the power was absorbed by the target in a CPA demonstration using a tuned absorber embedded in a lossy environment [124]. This work also showed that the target absorbed virtually nothing in the “anti-CPA” state, demonstrating a high degree of control over absorption by a specific target in a CPA scenario. An interesting future application is to utilize a generalized Wigner-Smith operator [125] to apply a high absorption fraction to a target with a modulated impedance or loss.

Full coherent multichannel CPA is more complicated than single channel perfect absorption. However, one advantage is the increase in delivered power by a factor of  $N$ , where  $N$  is the number of channels. This is a significant gain and worth the difficulty of additional phase and amplitude control.

### APPENDIX C: CAVITY AND EXPERIMENTAL CONFIGURATION

For the cavity discussed in the main paper, a line-of-sight block is used to obstruct the direct transmission path from port 2 to both ports 1 and 3. Also, ports 1 and 3 can be driven either independently or collectively with a relative phase shift provided by a NARDA phase shifter.

In the frequency range of interest, 3–4 GHz, the Weyl formula [126] predicts approximately 8524 resonant modes of the complex enclosure and the measured quality factor of the cavity is roughly  $5.5 \times 10^3$  [18]. A resonance mode width is then around five times greater than the mean mode spacing, which means there is some local overlap between modes.

### APPENDIX D: METASURFACE BINNING

The complexity of the cavity scattering responses, combined with the enormous number of possible metasurface command configurations ( $2^{240}$ ), means the direct development of a deep-learning network for the full space of 240 elements is overly ambitious. To simplify the problem, we reduced the number of degrees of freedom of the metasurface by binning together neighboring pixel elements, or grouping them together so that each element in a group is always commanded with the same value.

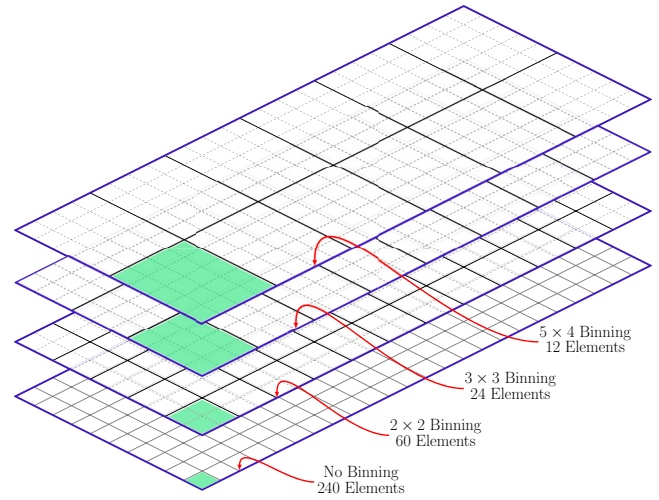


FIG. 7. Metasurface binning configurations. Binning configurations showing the relationship between the various options. The shaded green region identifies a single effective element for the specified configuration and the thin gray lines show the layout of the unbinned elements. With no binning there are 240 elements, binning into groups of  $2 \times 2$  yields 60 elements, binning into groups of  $3 \times 3$  yields 24 elements, and binning into groups of  $5 \times 4$  yields 12 elements. For the  $3 \times 3$  binning case, the bottom row of elements consists of a  $4 \times 3$  group so that all the elements are utilized.

Binning the metasurface elements reduces the total number of elements that must be determined and strengthens the relative change in cavity scattering parameters when driving a single effective element. Binning also promotes generality, as a metasurface with smaller elements can always approximate one with larger elements. This provides an alternative aspect of our approach and allows us to explore the use of deep-learning models in simpler configurations before working our way up to the more difficult cases. We used four different metasurface binning configurations, as shown in Fig. 7, to progressively decrease the number of elements.

### APPENDIX E: DATA PREPARATION AND COLLECTION

A major concern with deep learning is the amount of data required for training, which grows with the complexity of the problem being solved. To work within the constraint of reasonable training time, we wish to limit the number of data sets that must be collected. Therefore, acquiring good training data is of critical importance to ensure we cover the full range of possible responses. As found in earlier work [18], a diverse set of measurements requires variations in the number of active elements, spatial frequencies of active elements, and local groupings of active elements. Therefore, we utilized a random biased

coin toss approach with the bias itself a uniformly distributed random number to assign values to the elements for training data generation. To speed up operation as much as possible, the microwave network analyzer was configured to only provide  $S_{21}$  measurements vs frequency. With averaging disabled, collecting 4000 sets of data took a little under 1.5 h, while collecting 10 000 sets of data took roughly 3.5 h. Training was performed on a computer running Ubuntu 20.04 equipped with an NVIDIA RTX 3080 GPU, and took roughly 30 min for 10 000 training sets.

For the initial experiment, we collected 4000 sets of data in each of the specified binning configurations; the  $5 \times 4$  configuration allows 4096 unique metasurface combinations, so we collected 4096 sets (covering all possible combinations) in that case. With the exception of the  $5 \times 4$  binning configuration, the number of sets collected was far smaller than the number of possible configurations of the metasurface. As discussed in Sec. N, we found we needed 10 000 sets of data for the  $2 \times 2$  binning case.

## APPENDIX F: DEEP LEARNING AND NEURAL NETWORK LAYERS

The concept of depth in deep learning comes from complexity theory as defined for circuits, with depth being the longest path from an input to an output [127]. The number of potential paths or ways to reuse features grows exponentially with depth, which leads to progressively more abstract features [128,129]. Depth is therefore an important characteristic of a network to leverage as it enhances the expressive power of the network and allows it to learn a rich, hierarchical feature representation [130,131]. In physics, deep learning has repeatedly been shown to be unreasonably effective for extremely complicated problems [132].

A deep network is divided into layers, with the interior layers often referred to as hidden, as they are unobservable from the input or the output. Networks come in many different shapes and sizes; no single type is optimal for all problems, a consequence of the “no free lunch” theorem [133]. The networks described in this paper utilize four different types of layers. (1) Dense, linear, or fully connected layers are characterized by the number of neurons. The output is a linear combination of the inputs. (2) Convolutional layers are characterized by the number of filters and the length of the kernel (see Appendix G). The output is the result of convolving the inputs with the kernels. (3) Pooling layers are characterized by the pool size. The output is either the maximum or average value over a sliding window of width given by the pool size. These layers serve to reduce the size of the feature map and help ensure the learning process is position invariant. (4) Dropout layers characterized by the drop out rate. Dropout layers randomly set the specified percentage of inputs to 0

at each iteration in the training process, providing coarse regularization and simplifying the model.

Identifying the optimal deep-learning network topology for a given binning configuration took a significant amount of time to iterate over many potential designs (e.g., for our computing resources often several weeks) and was performed offline. Once the deep-learning network architecture was determined, we switched to an online, closed loop configuration where the data collection and training processes were separated by a few hours rather than days or weeks. The determined metasurface commands were directly applied to the metasurface and the resulting  $S_{21}$  responses were measured by the network analyzer, closing the loop. The online configuration also serves as a “field test” for the deep-learning network, further validating it against data not seen during training, as well as testing performance against potential small variations in measurement noise and the scattering configuration of the cavity itself.

## APPENDIX G: ONE-DIMENSIONAL CONVOLUTION

An aspect that is not well understood outside of the signal processing community is how convolutional layers are implemented for inputs containing multiple features. In signal processing, the feature dimensionality is referred to as the number of channels and is sometimes defined as the width or the depth of the data. This arises from color image processing with three color channels for red, green, and blue. To perform the convolution over the desired dimension and ensure all the features are captured, the convolution kernel is multidimensional as shown in Fig. 8. For a specified kernel length,  $k$ , the size of the kernel for a 1D convolutional layer with an input containing  $N$  features is  $k \times N$ . The kernel will only be shifted along a single dimension, the local frequency window in our case, but will contain optimized weights for each element. This means that the number of trainable parameters for a 1D convolutional layer scales as  $kN$ , not just  $k$ . For an input data set  $X$ , the output,  $y$ , of the convolutional layer with kernel  $K$  is given by

$$y[n] = \sum_{i=0}^{k-1} \sum_{j=0}^{N-1} K[i,j] X[n-i,j]. \quad (\text{G1})$$

For our purposes, we will zero-pad the input data by appending  $(k-1)/2$  rows of zeros to either end and keeping the central part of the result, so the number of points along the convolution dimension is constant in the output. By designing a convolutional layer consisting of  $N_f$  filters, there will be  $N_f$  such outputs or new features for the next layer.

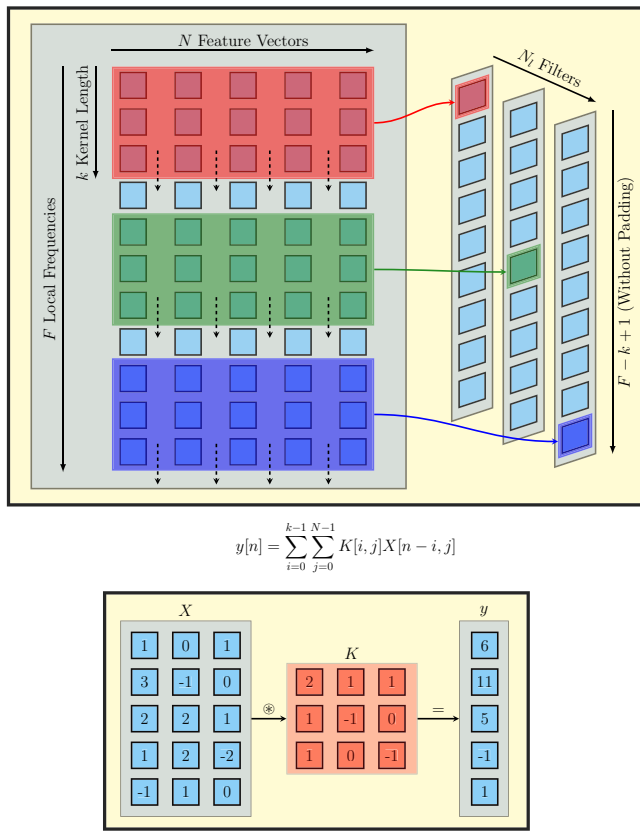


FIG. 8. One-dimensional convolution with multiple features. Top: Graphical representation showing the  $N$  feature vectors and  $F$  local frequencies processed by three different filters with kernel length  $k$ . The kernel only moves along a single dimension (vertically) even though the data are represented in a 2D format. Each position of the kernel results in a single point in the output vector which has length  $F - k + 1$  if zero-padding is not used and length  $F$  if padded as described in the text. The weights for each of the  $kN_f$  elements of the kernel are computed collectively, but can be different. Bottom: Numerical example with three features containing five points each convolved with a kernel of length 3. The input data are zero-padded with a row of zeros at the top and bottom and the outputs for the five central rows are kept.

## APPENDIX H: NETWORK TRAINING SETUP

For training the networks, the data were split into 75% training data and 25% validation data. The validation data are used to score the performance after each training run and are not used during the training process itself, so that validation is unbiased. The data were randomly shuffled prior to splitting and each network was trained several times to ensure results were in family and that the training process was unbiased as well.

To score the performance, we need to define metrics for loss and accuracy. The loss function was selected as mean absolute error to emphasize outliers in the data, and we define accuracy as the fraction of sets of commands that were predicted without error. To clarify the difference, the

loss function is defined as the average of the sum of the absolute value of the true commands,  $T_j$ , subtracted from the predicted commands,  $P_j$ , for set  $j$ , computed over  $N$  sets of  $M$  elements:

$$L = \frac{1}{NM} \sum_j |P_j - T_j|. \quad (\text{H1})$$

The loss function is then computed on a per-element basis and tells us how close the prediction was on average for each element. The output of the network is floating-point rather than binary, so the loss function does not necessarily provide an indication of the total number of incorrect predictions. The accuracy metric is defined as the percentage of sets predicted without a single error. It is evaluated on a per-set basis and explicitly uses the rounded output (0 or 1) from the network. Because accuracy is computed on a per-set basis, it is dependent on the number of elements in a command set and provides a more conservative estimate of performance for the various binning configurations. Accuracy is also more volatile, especially when the loss function is large. The loss function is continuous and more appropriate for training where we need to compute a gradient, while accuracy is a better metric for scoring the overall performance.

The networks were trained for 100–200 epochs using SGD with momentum. The basic SGD algorithm has potential problems with pathological curvature, or narrow ravines, which are common around local optima, and the response tends to oscillate back and forth across the ravine. To address this, we can use momentum [134], effectively forgetting a portion of the previous gradient. Momentum can be thought of as a very coarse approximation of the curvature or second derivative. To accelerate the training, we explicitly use Nesterov momentum [135].

The networks were trained in batches, meaning multiple data sets were evaluated at each iteration prior to updating the weights. This allows several samples of data to be processed simultaneously so that the effect of changes in the weights is observed over multiple sets of data, improving robustness and desensitizing the response to noise [136]. A batch size of 100 was used by default.

To prevent the networks from simply training on noise, we introduce an additional regularization step on the loss function. By enforcing an  $\mathcal{L}_2$  regularization scheme, the regularized loss function  $L^*$  is computed from the loss function  $L$ , and the vector of weights for the current iteration,  $\mathbf{w}_i$ , as  $L^* = L + \lambda \|\mathbf{w}_i\|^2$ . The value  $\lambda$  is referred to as a weight decay. The learning rate,  $\gamma$ , is the step size along the gradient, so the weights are incremented at each iteration as

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \gamma \nabla_{\mathbf{w}_i} L - 2\gamma \lambda \mathbf{w}_i. \quad (\text{H2})$$



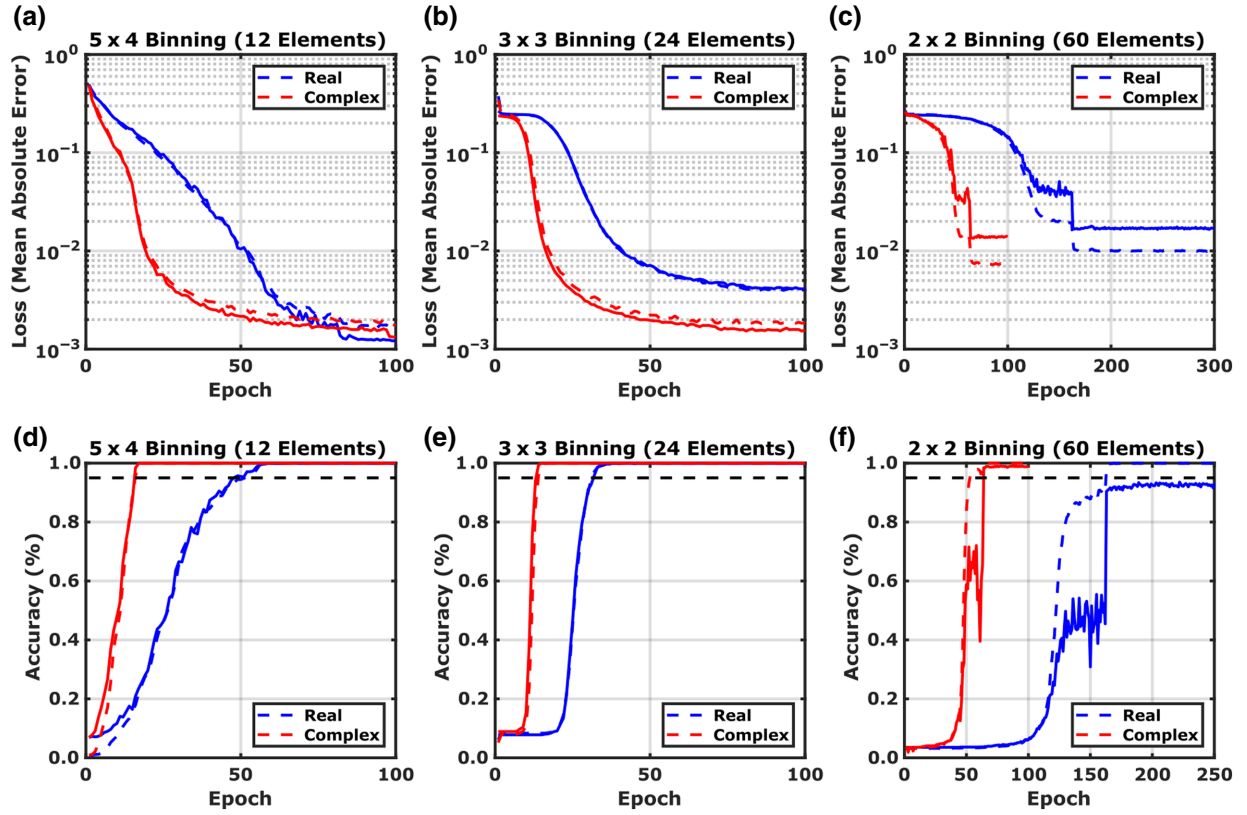


FIG. 9. Impact of using complex network layers. (a)–(c) Evolution of the loss function (mean absolute error) for the  $5 \times 4$ ,  $3 \times 3$ , and  $2 \times 2$  binning cases. (e)–(f) Evolution of the accuracy for the  $5 \times 4$ ,  $3 \times 3$ , and  $2 \times 2$  binning cases. The blue lines indicate real-valued network layers, the red lines indicate complex-valued network layers, the dashed lines indicate the training set, and the solid lines indicate the validation set. The dashed black line on the accuracy plots indicates 95% accuracy. Training was performed for 100 epochs in all cases except for the  $3 \times 3$  binning case with real-valued network layers, which was trained for 300 epochs. In each case, training with the complex-valued layers converged faster than training with the real-valued layers. For the  $2 \times 2$  binning case, the complex-valued network achieved higher accuracy for the training set (99.2%) than the real-valued network did (93.6%).

Finally, the learning rate is stepped down when the loss function plateaus, which allows the network to continue learning when it stalls due to the rate being too high.

## APPENDIX I: COMPLEX NETWORK LAYERS AND EXISTING DEEP-LEARNING FRAMEWORKS

With complex values, the mechanics of a network layer are the same as for the real-valued counterpart but they incur four times the computational cost due to having both real and imaginary components as well as the cross terms. Our initial deep-learning implementation leveraged Keras [137] and TensorFlow [138]. These provide an excellent, high-level framework that is very easy to use. Unfortunately this ease of use complicates things when attempting to develop custom complex-valued modules. Complex dense layers are straightforward to implement, but batch normalization, convolution, and recurrent layers are not. While there are repositories with complex deep networks containing some of these modules in Keras [139]

and Caffe [140], they are not actively maintained and are not formally supported by the frameworks. In the case of Keras, changes to the way the back end is handled in the most recent version (v2.4) mean that the complex library [139] is no longer functional and would require significant modification to bring up to date.

This leads us to utilize PyTorch [141], another deep-learning framework. The interface to PyTorch is lower level than Keras, which means it requires more knowledge of PYTHON to use effectively, but that it is also easier to implement custom modules. In addition, there is an open source complex library written by Sebastien Popoff [59] that includes complex versions of dense, convolutional, and batch normalization layers. We were able to utilize this library with only minor modifications to the batch normalization implementation to handle our multiple feature data sets.

Figure 9 shows the impact of using complex network layers. In each of the panels, the blue plots indicate results for a real-valued network while the red plots indicate results for a complex-valued network. In addition, the solid

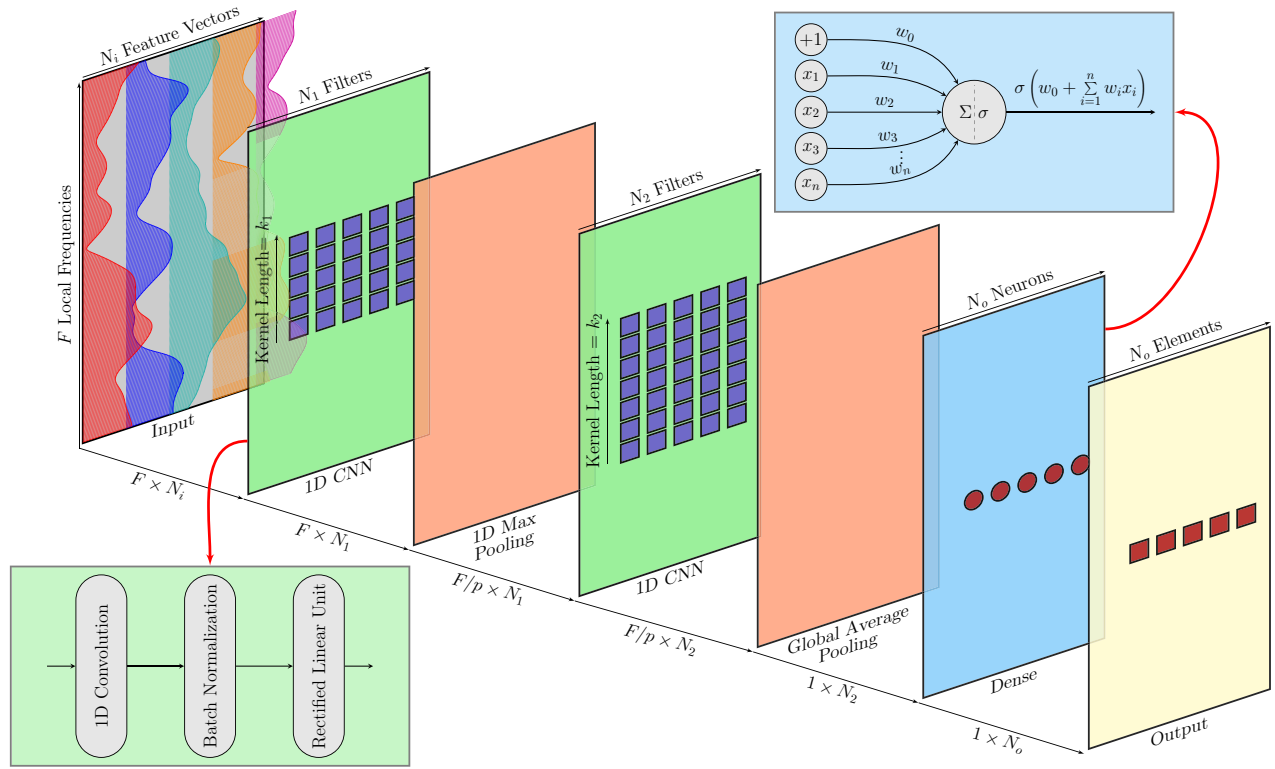


FIG. 10. Sequential network layer architecture. The input layer consists of  $N_i$  feature vectors containing  $S_{21}$  measurements in a local 10 MHz window of  $F$  points, for a total size of  $F \times N_i$ . This is followed by a series of 1D convolutional layers defined by the number of filters and the kernel length. Each convolutional layer includes a 1D convolution, a batch normalization to keep the distribution statistics constant throughout the network, and a rectified linear unit activation function. Interspersed with the convolutions are 1D max pooling layers defined by the pool size,  $p$ , that serve to reduce the dimensionality of the local frequency window. The output stage consists of a global average pooling layer to further reduce dimensionality and convert the complex-valued signals to magnitude, followed by a dense or fully-connected layer to ensure the correct number of outputs. The dense layer produces outputs that are linear combinations of the outputs from the global average pooling layer, and is followed by a sigmoid activation function to approximate binary values at the output.

lines show results for the validation set while the dashed lines show the results for the training set. The complex-valued networks all converged faster than the real-valued networks, and the complex-valued network achieved better accuracy on the training set than the real-valued network did for the  $2 \times 2$  binning case. The only differences in training were for the real-valued network in the  $2 \times 2$  binning case. The two differences here were that (1) the “patience” parameter, or number of epochs to wait before reducing the learning rate, had to be increased significantly because the training converged slowly even with an aggressive learning rate; and (2) the number of epochs had to be increased from 100 to 300 in order to capture the converged model. The increased patience parameter leads to a longer period of oscillations in the validation set loss function.

The acceleration in training comes with a caveat in that the overall computational time for the purely real-valued deep-learning network is still less than that of the complex-valued deep-learning network. Complex-valued

layers increase the computation requirements for multiplication and convolution by a factor of 4 to handle the real and imaginary terms as well as the cross terms. In addition, highly optimized and efficient implementations of purely real-valued layers are readily available through the NVIDIA CUDA deep neural network library (cuDNN), but are not available for their complex-valued counterparts.

## APPENDIX J: NETWORK ARCHITECTURE FOR SEQUENTIAL LAYERS

Figure 10 presents the generalized architecture used for sequential layers. The input consists of  $N_i$  feature vectors containing the local 10 MHz windows with  $F$  points in each vector. One-dimensional CNN layers with  $N_l$  filters and a kernel length of  $k_l$  at the  $l$ th layer perform the feature extraction. As shown in the lower inset, each CNN layer includes a 1D convolution followed by a batch normalization and a rectified linear unit (ReLU) activation function. The batch normalization is used to ensure the distribution

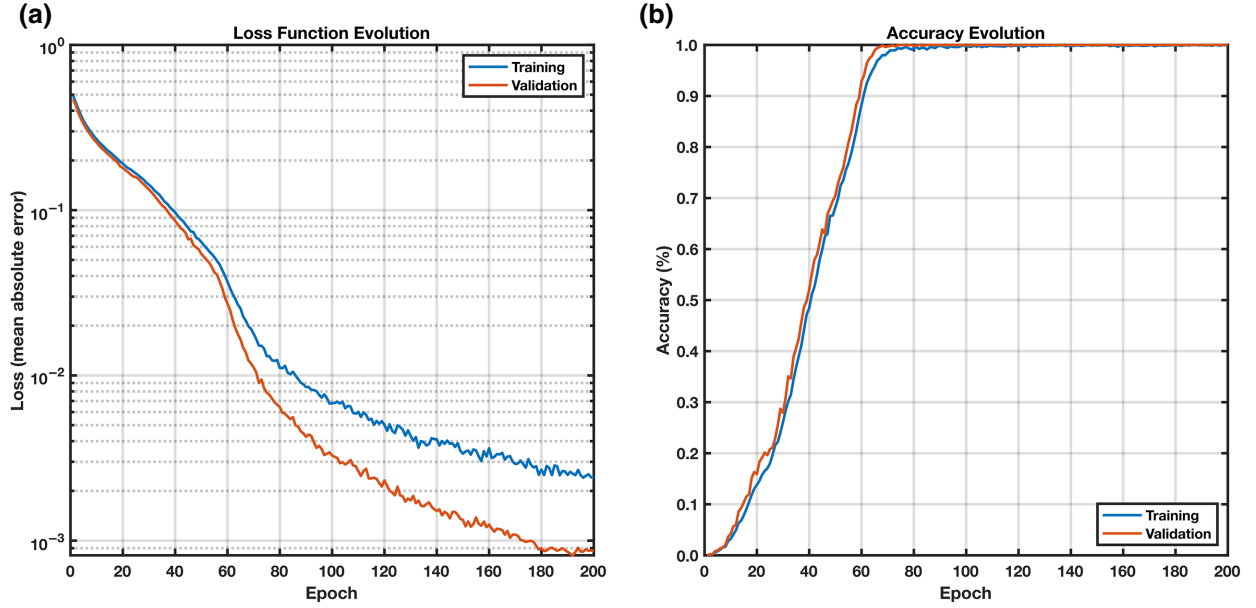


FIG. 11. Deep-learning performance with complex-valued layers for  $5 \times 4$  Binning. (a) Evolution of the loss function for the training and validation sets over 200 epochs. The loss function measures the average prediction error per element and provides an estimate of the confidence in the prediction. (b) Evolution of the accuracy for the training and validation sets over 200 epochs. Accuracy provides the relative number of sets of commands that were predicted without error, and shows that perfect prediction was achieved on both the training and validation sets in fewer than 100 epochs. The loss function continues decreasing after the accuracy saturates at 100% because it is continuous and evaluated on the floating-point predicted values, and the decrease indicates the network is still learning and improving its estimate.

of the data (mean and variance) remains relatively constant throughout the network; changing distributions between layers induces internal covariate shift and leads to convergence issues during training [142]. The ReLU activation function is used in virtually all deep-learning networks as it does not experience a vanishing gradient due to saturation, and leads to expedited convergence and generally better solutions than sigmoid like functions [143].

The CNN layers are grouped together to form stages and are interspersed with 1D max pooling layers defined by the pool size,  $p$ . Also included are dropout layers for regularization, which are not explicitly shown in Fig. 10. The output stage contains a global average pooling layer that averages along the  $F$  dimension, reducing the feature maps to a single dimension. It also converts the complex-valued signals to magnitude and is followed by a dense layer that provides the correct number of outputs,  $N_o$ . As shown in the upper inset, the output of the dense layer is a linear combination of the outputs from the global pooling layer, with a sigmoid activation function used to clip the output between 0 and 1.

#### APPENDIX K: OFFLINE TRAINING RESULTS FOR $5 \times 4$ BINNING

When using  $5 \times 4$  binning, the metasurface is effectively partitioned into 12 elements, for 4096 possible sets

of commands. We measured all 4096 combinations, and the 75%/25% split yielded 3072 sets for training and 1024 sets for validation. The batch size was set to 64 as a result. A purely sequential deep network was utilized, following the layout given in Fig. 10. Four CNN layers, a max pooling layer, and a dropout layer were combined into a stage. Four stages were then used, with the output provided by a global average pooling layer and a dense layer with a sigmoid activation function.

Initial experiments used purely real-valued deep-learning layers, in which case the global average pooling layer only provided dimensionality reduction. We were able to establish excellent prediction performance, regularly achieving fewer than 10 total prediction errors over the validation set or better than 99% accuracy after training for 500 epochs. When switching to complex-valued layers with the same architecture, we were able to regularly achieve perfect prediction (100%) over both the training and validation sets after training for fewer than 100 epochs. This improvement demonstrates that the complex-valued deep-learning network is able to exploit phase as well as amplitude to better fit the relationship between metasurface commands and transmitted power.

The training results for the  $5 \times 4$  binning case with complex-valued layers are shown in Fig. 11. Figure 11(a) shows the loss function evolution for the training and validation sets while Fig. 11(b) shows the accuracy evolution.

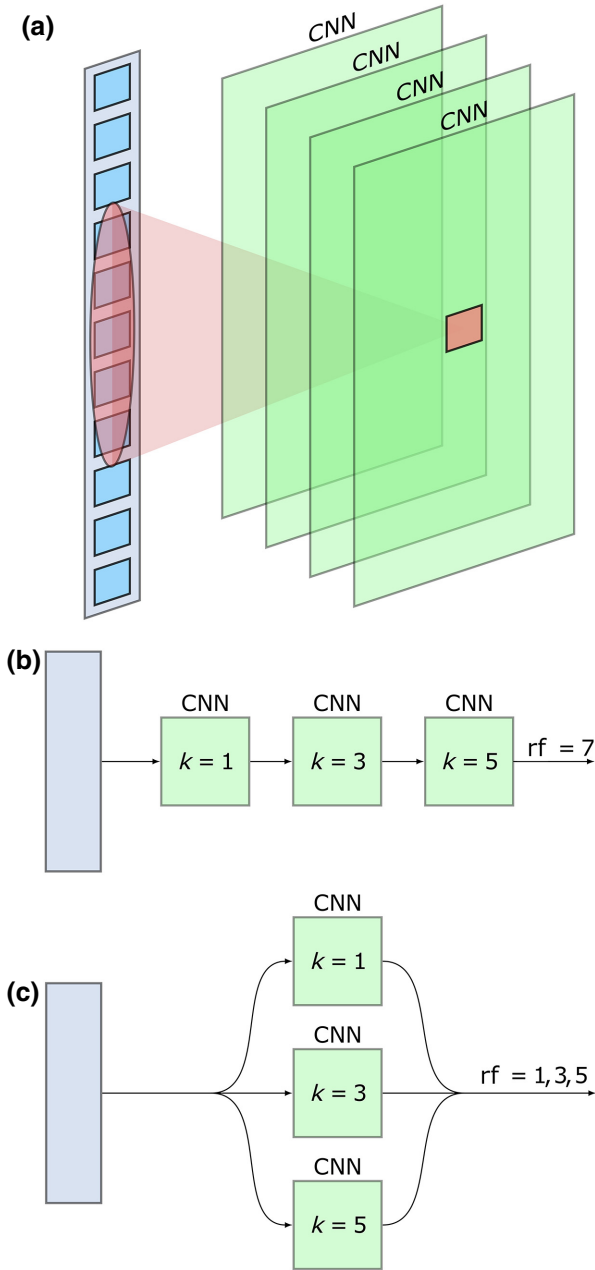


FIG. 12. Receptive field. (a) The receptive field of a convolutional neural network (CNN) layer indicates the number of points in input space that contribute to a single point at a given layer. (b) For a purely sequential architecture, the receptive field increases monotonically. (c) A parallel architecture with concatenation produces multiple receptive fields, with each available for subsequent layers, promoting sparsity in the representations.

The performance is excellent, achieving perfect prediction over both the training and validation data sets. Note that the loss function continues to improve even after the accuracy saturates at 100%. This is because the loss function is continuous and computed using the floating-point predicted values rather than the rounded binary values; it shows that

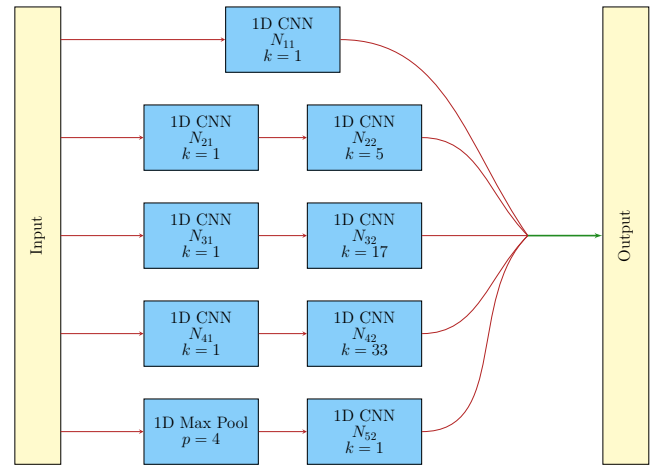


FIG. 13. Terrapin Module architecture. Five parallel branches with eight 1D CNN layers and a max pool layer are used in the module. The module operates on the pseudo-2D data format discussed in the text, and the input layer can ingest either the raw measured  $S_{21}$  parameters or the outputs from a previous Terrapin Module. The output layer is then connected either to a subsequent Terrapin Module for additional processing or to the final output layer for conversion to metasurface commands. Each CNN includes a 1D convolution, a batch normalization, and a rectified linear unit activation function. The second-level CNNs have kernel lengths of 5, 17, and 33 to increase the receptive field by 125 kHz, 500 kHz, and 1 MHz, respectively. A 1D max pooling layer with pool size of 4 is included to provide a pooling window of 125 kHz as well.  $N_{xx}$  denotes a tunable parameter for the number of convolutional filters at each branch and stage, acting as a dense or fully connected layer for the global correlations. The convolutions with unit length kernels serve to buffer and condition the inputs to each stage, and the single-layer first branch maintains the receptive field sizes from previous modules. The outputs of each branch are concatenated together to form the module output, preserving the receptive field sizes for subsequent layers.

the network is still learning and continuing to increase its confidence in the prediction.

## APPENDIX L: RECEPTIVE FIELD AND INCEPTION MODULE

The receptive field of a CNN defines the number of points in input space that contribute to the result at a single point in a given layer. Our CNNs use zero-padding to keep the output size fixed, and the stride and dilation are always set to 1. This means the receptive field at any layer,  $r_l$ , is given by a simple recursive equation dependent on the receptive field at the previous layer,  $r_{l-1}$ , and the kernel length of the current layer,  $k_l$  [144]:

$$r_l = r_{l-1} + k_l - 1. \quad (\text{L1})$$

As shown in Fig. 12, the receptive field for a sequential architecture grows monotonically with depth, with each layer only seeing the receptive field from the preceding



layer. An architecture that utilizes parallel branches along with concatenation conserves the intermediate receptive fields, making them available for all subsequent layers. This introduces width as well as depth to the network and provides the motivation for the inception module.

An inception module is designed to promote sparse feature representation using available dense components [61] and works by optimizing the receptive field coverage of a convolutional network. The receptive field is the number of points in input space that contribute to a point at a given layer of the deep-learning network. Through the use of parallelization and concatenation, the receptive field sizes at a layer are conserved for subsequent layers to utilize, extracting features through the width of the deep-learning network as well as its depth.

The original inception module was developed for image processing and operates in a true 2D space, with full 2D convolutions. It uses four parallel paths with CNN layers containing unit length kernels for buffering and conditioning, along with three and five sample length kernels for feature extraction, and a max pool layer to improve performance [61]. There have been several variations of the inception module; however, none operate in the pseudo-2D space we desire. For our “images,” the frequency spacing along columns is the resolution of the network analyzer (31.25 kHz), while the frequency spacing along rows is the separation between local windows (10 MHz). The difference in sampling means we need to treat the rows and columns accordingly and avoid traditional 2D convolutions that assume uniform sampling. Therefore, we modified the general architecture of the inception module to perform 1D convolutions over the 10 MHz local frequency windows. The 1D convolutional filters then extract local features over the 10 MHz windows, while the relationship between the filters acts as a dense or fully connected layer, extracting global features over the full 1 GHz measurement window.

From previous work with the cavity, we found that the mean mode spacing is approximately 125 kHz and we demonstrated the ability to generate strong nulls over a 500 kHz bandwidth [18]. This suggests we should use a pooling window of 125 kHz and allow the receptive field to increase by 125 kHz and 500 kHz at each stage or layer in the deep-learning network. After experimenting, we found that adding a fifth stage which increased the receptive field by 1 MHz helped to further improve performance. We refer to the final version of our module as a “Terrapin Module,” a block diagram of which is shown in 13.

#### APPENDIX M: OFFLINE TRAINING RESULTS FOR $3 \times 3$ BINNING

For the  $3 \times 3$  binning configuration, the purely sequential network did not perform very well and was unable

to learn the relationships for either the training or validation sets. This inspired the modified inception module that we defined as the Terrapin Module in the main paper. The performance difference between the sequential CNN model and the Terrapin Module is shown in Fig. 14, which presents training results for the  $5 \times 4$ ,  $3 \times 3$ , and  $2 \times 2$  binning cases. The sequential CNN is not able to train very well for the more complicated systems ( $3 \times$  and  $2 \times 2$  binning cases), while the Terrapin Module is able to exploit the more complicated relationships and provide similar performance to the sequential CNN model on the  $5 \times 4$  binning case.

The results for the  $3 \times 3$  binning case with complex-valued layers are shown in Fig. 15. The impact of reducing the learning rate on a plateau can be seen at epoch 54, where a drop in the learning rate by a factor of 10 induces a drop in the loss function by approximately a factor of 2.

#### APPENDIX N: OFFLINE TRAINING RESULTS FOR $2 \times 2$ BINNING

For the  $2 \times 2$  configuration with 4000 sets of data, we were able to achieve better than 98% accuracy on the training set, but were limited to approximately 50% accuracy on the validation set. The discrepancy between training and validation results is a hallmark of overtraining. In this particular case, the validation results were improving but stalled as the training results approached 100% accuracy. The error landscape became extremely small with a negligible gradient, so there was no direction to take and continue learning. The network therefore learned specific features of the training set rather than general features of the full range of possible responses. This suggests the overtraining is due to having a limited amount of data (only 4000 sets). We captured a larger amount of data (10 000 sets) and were able to achieve better than 95% accuracy on both the training and validation sets. Perfect accuracy for the validation set may be possible with the collection of an even larger amount of data.

#### APPENDIX O: SCATTERING FIDELITY LOSS

Figure 16 shows the decay in scattering fidelity for online validation at the 4, 5, and 9 day marks. The accuracy is still better than 85% after 5 days, but the number of sets with more than one prediction error has increased. After 9 days, the accuracy drops to 65.5% and many cases with 2, 3, and even 4 prediction errors are found.

#### APPENDIX P: PERFORMANCE VS REVERBERATION TIME

An additional set of experiments was performed to determine the impact of cavity reverberation time on the performance of the deep-learning network. To increase the losses in the cavity (and decrease the reverberation time),

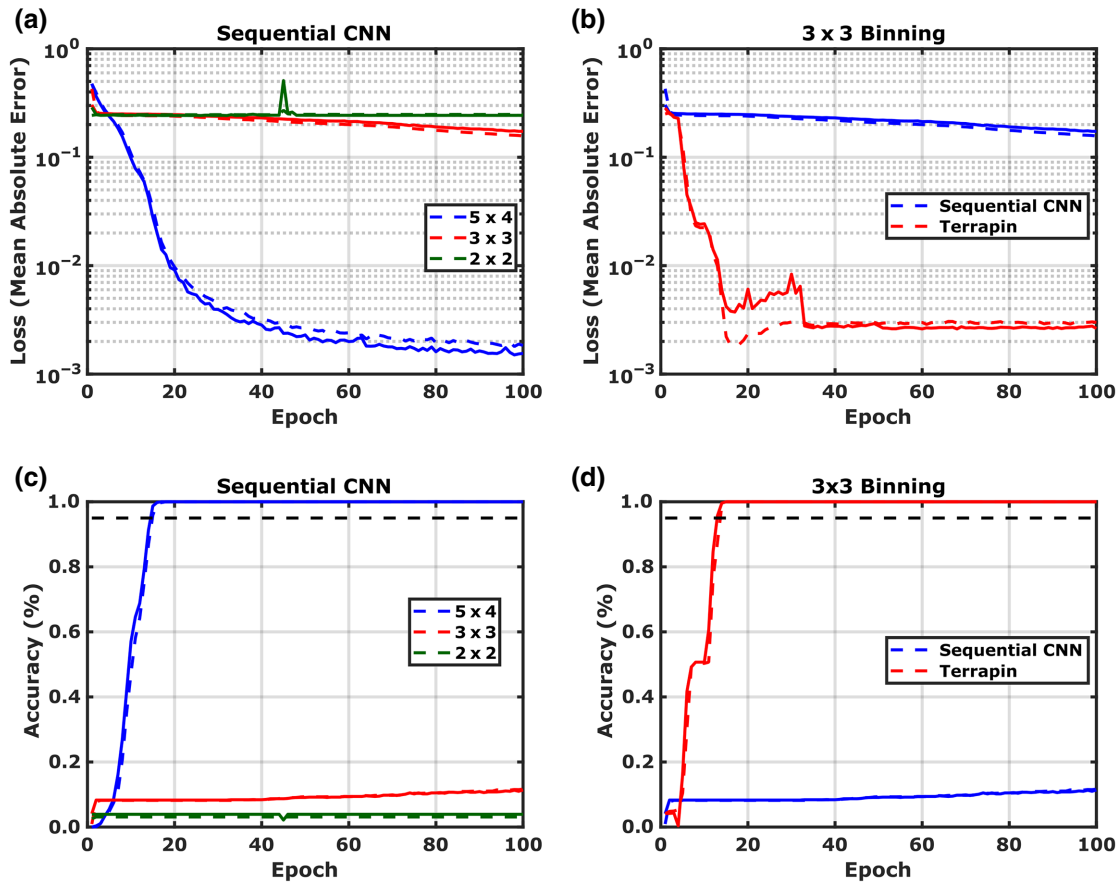


FIG. 14. Sequential neural network performance with complex scattering systems. The solid lines indicate results for the validation data while the dashed lines indicate results for the training data. (a),(c) Evolution of the loss function and accuracy for the  $5 \times 4$ ,  $3 \times 3$ , and  $2 \times 2$  binning cases using the sequential CNN model. Only the  $5 \times 4$  binning case is able to significantly reduce the loss function and provide reasonable accuracy. There is no separation between the validation results and the training results, indicating that there is not an issue with too little data. (b),(d) Evolution of the loss function and accuracy for the  $3 \times 3$  binning case using the sequential CNN and Terrapin Modules. The Terrapin Module provides similar loss and accuracy to the  $5 \times 4$  binning case with the sequential CNN.

rf absorbent materials were placed inside the cavity. For each loss configuration, an ensemble of measurements was collected using the mechanical mode stirrer and the reverberation time was estimated from the power delay profile [145]. The mode stirrer was then set to a fixed position and another ensemble was collected for training data with 10 000 random metasurface configurations (following a biased random coin toss approach). The correlation coefficient was computed over all possible measurement pairs, for  $5 \times 10^7$  combinations, to assess how highly correlated the training sets were. The deep-learning network was then trained using the same network and parameters as previously discussed and the results are shown in Fig. 17. The cavity reverberation time ranged from 23 ns to 179 ns. The statistics of the correlation coefficients are shown relative to the left-hand axis, and show the median value, quartiles, and full extent. The achieved accuracy of the deep-learning network on the training set is shown as the dashed red line relative to the right-hand axis and indicates

that the accuracy and correlation coefficients are inversely related. The deep-learning network is capable of operating in extremely complicated scattering environments, but the performance degrades as the cavity losses increase. This is because ray trajectories do not persist as long for high-loss systems; the number of bounces for a given trajectory is reduced, which means there are fewer rays intercepted by the metasurface.

## APPENDIX Q: FUTURE DIRECTIONS

Future directions will refine our technique to intentionally scramble (or unscramble) waves propagating through a complex scattering environment. We mention three aspirational goals. The first is tuning the scattering responses through a controller that optimizes the system for a given application at arbitrary frequencies and bandwidths. Specific metrics include minimizing transmitted power for

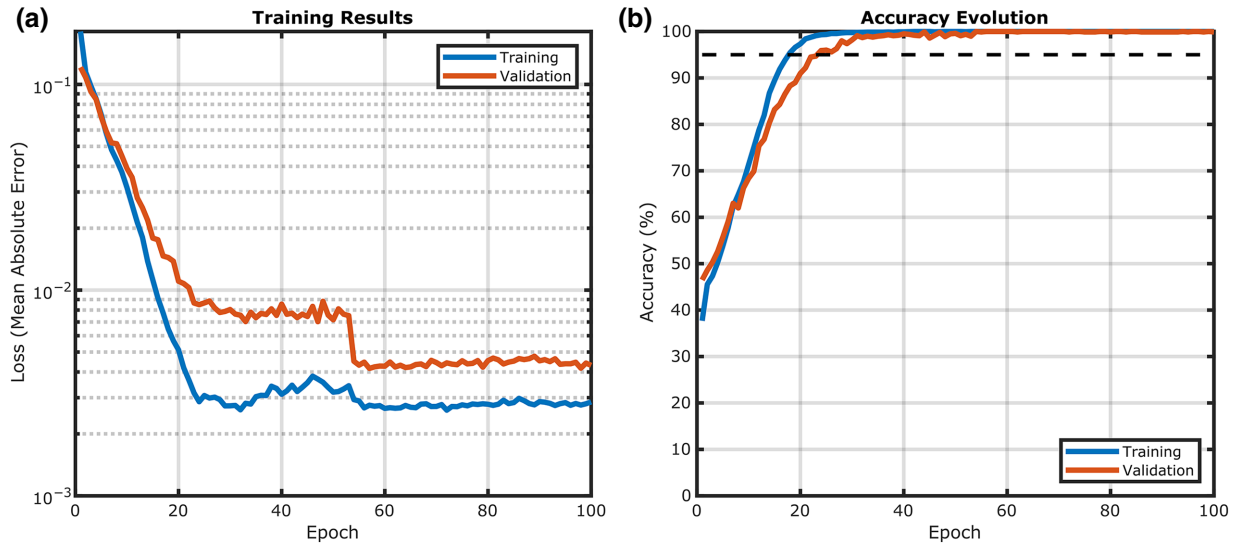


FIG. 15. Deep-learning performance with complex-valued layers for  $3 \times 3$  binning. (a) Evolution of the loss function for the training and validation sets over 100 epochs. The loss function hits a plateau at approximately epoch 33 but shows an additional drop at epoch 54 when the learning rate is reduced. (b) Evolution of the accuracy for the training and validation sets over 100 epochs. Accuracy provides the relative number of sets of commands that were predicted without error, and shows that perfect prediction was achieved on both the training and validation sets in fewer than 100 epochs. The loss function continues decreasing after the accuracy saturates at 100% because it is continuous and evaluated on the floating-point predicted values, and the decrease indicates the network is still learning and improving its estimate.

cold spot generation, minimizing scattering matrix eigenvalue magnitudes for coherent perfect absorption, or minimizing the bit-error rate for wireless communication. The second aspirational goal is to introduce feedback from the environment to dynamically update the controller and react to changing environmental conditions. The third aspirational goal is to realize a fully autonomous system that

enables persistent and robust smart radio environments that do not require human intervention.

For on-the-fly learning and adaptation to changing environmental conditions, we propose the future use of reinforcement learning [56,146], which is at the intersection of artificial intelligence and optimal control. Reinforcement learning uses an agent that interacts with an environment

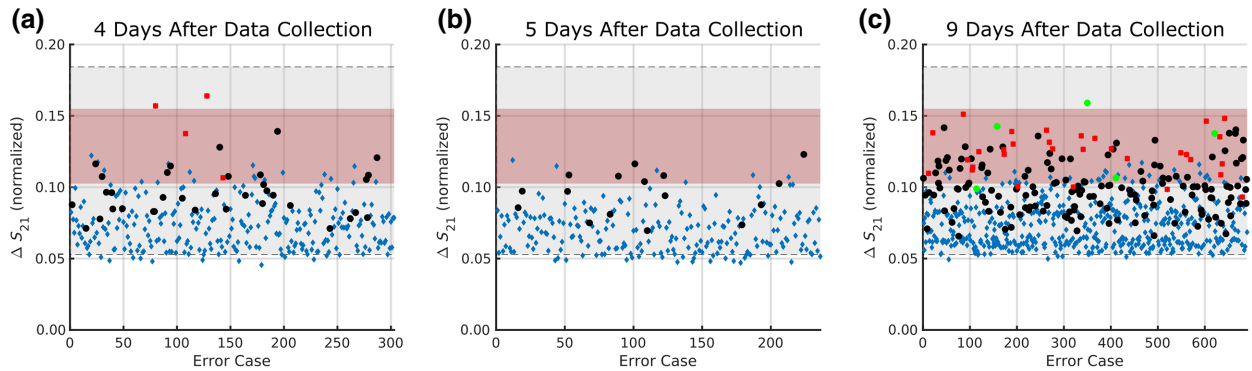


FIG. 16. Scattering fidelity loss over time.  $\Delta S_{21}$  for online validation sets taken a specified time after the training data were collected. The shaded regions show the extent of the single-element Hamming distance results from the training data. The gray region shows the full range from maximum to minimum, and the red region shows the 25th and 75th percentiles. The blue diamonds indicate cases with a single prediction error, the black circles indicate cases with two prediction errors, the red squares indicate cases with three prediction errors, and the green circles indicate cases with four prediction errors. These panels show that the  $\Delta S_{21}$  for prediction errors is very small, and in the lower region of the statistics covered by observed cases with single-element Hamming distances. (a) Validation 4 days after collecting training data; 2000 sets of commands were tested, with 303 mispredictions for an accuracy of 84.9%. (b) Validation 5 days after collecting training data; 2000 sets of commands were tested, with 236 mispredictions for an accuracy of 88.2%. (c) Validation 9 days after collecting training data; 2000 sets of commands were tested, with 690 mispredictions for an accuracy of 65.5%.

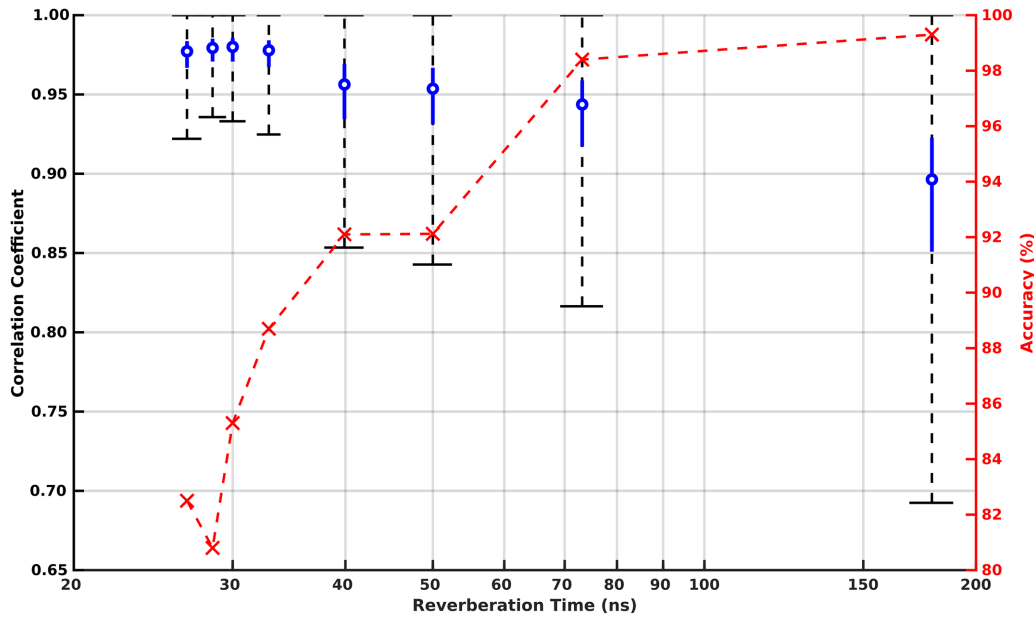


FIG. 17. Deep-learning performance vs cavity reverberation time. Performance of the deep-learning network for different cavity loss configurations as specified by the cavity reverberation time ( $x$  axis). The reverberation time is shown on a log scale to highlight the behavior for high-loss configurations (short reverberation times). The statistics of the correlation coefficient over the approximately 50 million combinations of measurement sets are shown relative to the left-hand  $y$  axis. The blue line shows the extent between the quartiles, the blue circle indicates the median value, and the dashed black line shows the full extent. The achieved accuracy of the deep-learning network is shown as the dashed red line relative to the right-hand  $y$ -axis, with individual points represented by a cross. The trend shows an inverse relationship between the correlation coefficient and the achieved accuracy of the deep-learning network, indicating that the deep-learning network struggles to identify features in the data when they are highly correlated.

to learn about it and then manipulate it in order to maximize (minimize) a reward (cost function), leading to the development of optimal control policies. In particular, the subset of reinforcement learning known as deep or double-deep “Q” learning is gaining traction as a method for controlling quantum states [147–149]. Deep “Q” learning uses a deep-learning network to estimate a quality matrix that scores the result of taking a particular action, while double-deep “Q” learning uses two estimates to limit the implementation of poor control policies from over-estimation [150]. The deep-learning network architecture developed in this paper is well suited for estimation of this quality matrix.

Specification of an arbitrary scattering condition in the current implementation is cumbersome, as the complete  $S_{21}$  response over the full 3–4 GHz measurement window must be defined. For practical engineering applications, we prefer a simpler method of defining a desired wave scattering condition. Deep reinforcement learning also helps in this case, as it scores the performance of an agent through a scalar, positive, and real-valued metric. The agent uses the deep-learning network to learn the relationship between metasurface commands and  $S_{21}$  responses, but the complicated details are hidden from the user. There are therefore two learning components to deep reinforcement learning: an inner deep-learning network that learns how to map

$S_{21}$  responses onto metasurface commands, and an outer agent-based loop that learns how to use the inner deep-learning network to optimize the desired metric. This metric can be the total power in a specified bandwidth for cold spot generation, the magnitude of the eigenvalues of the full  $S$ -matrix at a given frequency for coherent perfect absorption, or the bit error rate for communications systems.

Learning from scratch can be slow and may not be fast enough to adapt to changing environmental conditions. In this case, transfer learning, or using information about a similar problem to accelerate training for another one, can be incorporated into the reinforcement learning strategy [151].

- 
- [1] L. Subrt and P. Pechac, Intelligent walls as autonomous parts of smart indoor environments, *IET Commun.* **6**, 1004 (2012).
  - [2] M. D. Renzo, M. Debbah, D.-T. Phan-Huy, A. Zapone, M.-S. Alouini, C. Yuen, V. Sciancalepore, G. C. Alexandropoulos, J. Hoydis, H. Gacanin, J. de Rosny, A. Bounceur, G. Lerosey, and M. Fink, Smart radio environments empowered by reconfigurable ai meta-surfaces: An idea whose time has come, *EURASIP J. Wirel. Commun. Netw.* **129**, 2450 (2019).



- [3] M. Di Renzo, A. Zappone, M. Debbah, M. S. Alouini, C. Yuen, J. de Rosny, and S. Tretyakov, Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead, *IEEE J. Sel. Areas Commun.* **38**, 2450 (2020).
- [4] M. Horodynski, M. Kühmayer, A. Brandstötter, K. Pichler, Y. V. Fyodorov, U. Kuhl, and S. Rotter, Optimal wave fields for micromanipulation in complex scattering environments, *Nat. Photonics* **14**, 149 (2020).
- [5] M. Frazier, B. Taddese, T. Antonsen, and S. M. Anlage, Nonlinear Time Reversal in a Wave Chaotic System, *Phys. Rev. Lett.* **110**, 063902 (2013).
- [6] B. Xiao, T. M. Antonsen, E. Ott, and S. M. Anlage, Focusing waves at arbitrary locations in a ray-chaotic enclosure using time-reversed synthetic sonas, *Phys. Rev. E* **93**, 052205 (2016).
- [7] I. M. Vellekoop and A. P. Mosk, Focusing coherent light through opaque strongly scattering media, *Opt. Lett.* **32**, 2309 (2007).
- [8] A. Mosk, A. Lagendijk, G. Lerosey, and M. Fink, Controlling waves in space and time for imaging and focusing in complex media, *Nat. Photonics* **6**, 283 (2012).
- [9] M. Booth, D. Andrade, D. Burke, B. Patton, and M. Zurauskas, Aberrations and adaptive optics in super-resolution microscopy, *Microscopy* **64**, 251 (2015).
- [10] M. Collini, F. Radaelli, L. Sironi, N. G. Ceffa, L. D'Alfonso, M. Bouzin, and G. Chirico, Adaptive optics microspectrometer for cross-correlation measurement of microfluidic flows, *J. Biomed. Opt.* **24**, 1 (2019).
- [11] M. Dupré, P. del Hougne, M. Fink, F. Lemoult, and G. Lerosey, Wave-Field Shaping in Cavities: Waves Trapped in a Box with Controllable Boundaries, *Phys. Rev. Lett.* **115**, 017701 (2015).
- [12] N. Kaina, M. Dupré, G. Lerosey, and M. Fink, Shaping complex microwave fields in reverberating media with binary tunable metasurfaces, *Sci. Rep.* **4**, 6693 (2015).
- [13] P. del Hougne, F. Lemoult, M. Fink, and G. Lerosey, Spatiotemporal Wave Front Shaping in a Microwave Cavity, *Phys. Rev. Lett.* **117**, 134302 (2016).
- [14] P. del Hougne, M. Davy, and U. Kuhl, Optimal Multiplexing of Spatially Encoded Information across Custom-Tailored Configurations of a Metasurface-Tunable Chaotic Cavity, *Phys. Rev. Appl.* **13**, 041004 (2020).
- [15] J.-B. Gros, P. del Hougne, and G. Lerosey, Tuning a regular cavity to wave chaos with metasurface-reconfigurable walls, *Phys. Rev. A* **101**, 061801 (2020).
- [16] M. F. Imani, D. R. Smith, and P. del Hougne, Perfect absorption in a disordered medium with programmable meta-atom inclusions, *Adv. Funct. Mater.* **30**, 2005310 (2020).
- [17] P. del Hougne, D. V. Savin, O. Legrand, and U. Kuhl, Implementing nonuniversal features with a random matrix theory approach: Application to space-to-configuration multiplexing, *Phys. Rev. E* **102**, 010201 (2020).
- [18] B. W. Frazier, T. M. Antonsen, S. M. Anlage, and E. Ott, Wavefront shaping with a tunable metasurface: Creating cold spots and coherent perfect absorption at arbitrary frequencies, *Phys. Rev. Res.* **2**, 043422 (2020).
- [19] P. del Hougne, K. B. Yeo, P. Besnier, and M. Davy, On-demand coherent perfect absorption in complex scattering systems: Time delay divergence and enhanced sensitivity to perturbations, *Laser. Photon. Rev.* **15**, 2000471 (2021).
- [20] A. Palermo and A. Marzani, Control of love waves by resonant metasurfaces, *Sci. Rep.* **8**, 7234 (2018).
- [21] Y. He, T. Chen, and X. Song, Manipulation of seismic rayleigh waves using a phase-gradient rubber metasurface, *Int. J. Mod. Phys. B* **34**, 2050142 (2020).
- [22] R. Bekenstein, I. Pikovski, H. Pichler, E. Shahmoon, S. F. Yelin, and M. D. Lukin, Quantum metasurfaces with atom arrays, *Nat. Phys.* **16**, 676 (2020).
- [23] I. Vellekoop and A. Mosk, Phase control algorithms for focusing light through turbid media, *Opt. Commun.* **281**, 3071 (2008).
- [24] C. Dorrer and J. Qiao, Direct binary search for improved coherent beam shaping and optical differentiation wavefront sensing, *Appl. Opt.* **57**, 8557 (2018).
- [25] S. Ma, B. Xiao, R. Hong, B. Addissie, Z. Drikas, T. Antonsen, E. Ott, and S. Anlage, Classification and prediction of wave chaotic systems with machine learning techniques, *Acta Phys. Polonica A* **136**, 757 (2019).
- [26] J. Peurifoy, Y. Shen, L. Jing, Y. Yang, F. Cano-Renteria, B. G. DeLacy, J. D. Joannopoulos, M. Tegmark, and M. Soljačić, Nanophotonic particle simulation and inverse design using artificial neural networks, *Sci. Adv.* **4**, eaar4206 (2018).
- [27] D. Liu, Y. Tan, E. Khoram, and Z. Yu, Training deep neural networks for the inverse design of nanophotonic structures, *ACS Photonics* **5**, 1365 (2018).
- [28] C. C. Nadell, B. Huang, J. M. Malof, and W. J. Padilla, Deep learning for accelerated all-dielectric metasurface design, *Opt. Express* **27**, 27523 (2019).
- [29] S. An, C. Fowler, B. Zheng, M. Y. Shalaginov, H. Tang, H. Li, L. Zhou, J. Ding, A. M. Agarwal, C. Rivero-Baleine, K. A. Richardson, T. Gu, J. Hu, and H. Zhang, A deep learning approach for objective-driven all-dielectric metasurface design, *ACS Photonics* **6**, 3196 (2019).
- [30] T. Qiu, X. Shi, J. Wang, Y. Li, S. Qu, Q. Cheng, T. Cui, and S. Sui, Deep learning: A rapid and efficient route to automatic metasurface design, *Adv. Sci.* **6**, 1900128 (2019).
- [31] I. Sajedian, H. Lee, and J. Rho, Double-deep Q-learning to increase the efficiency of metasurface holograms, *Sci. Rep.* **9**, 10899 (2019).
- [32] C. Yeung, J.-M. Tsai, B. King, Y. Kawagoe, D. Ho, M. W. Knight, and A. P. Raman, Elucidating the behavior of nanophotonic structures through explainable machine learning algorithms, *ACS Photonics* **7**, 2309 (2020).
- [33] P. R. Wiecha and O. L. Muskens, Deep learning meets nanophotonics: A generalized accurate predictor for near fields and far fields of arbitrary 3D nanostructures, *Nano Lett.* **20**, 329 (2020).
- [34] J. Jiang, M. Chen, and J. A. Fan, Deep neural networks for the evaluation and design of photonic devices, *Nature Reviews Materials* (2020).
- [35] R. Unni, K. Yao, and Y. Zheng, Deep convolutional mixture density network for inverse design of layered photonic structures, *ACS Photonics* **7**, 2703 (2020).
- [36] A. Mall, A. Patil, A. Sethi, and A. Kumar, A cyclical deep learning based framework for simultaneous inverse and forward design of nanophotonic metasurfaces, *Sci. Rep.* **10**, 19427 (2020).

- [37] S. An, B. Zheng, M. Y. Shalaginov, H. Tang, H. Li, L. Zhou, J. Ding, A. M. Agarwal, C. Rivero-Baleine, M. Kang, K. A. Richardson, T. Gu, J. Hu, C. Fowler, and H. Zhang, Deep learning modeling approach for metasurfaces with high degrees of freedom, *Opt. Express* **28**, 31932 (2020).
- [38] L. Li, Y. Shuang, Q. Ma, H. Li, H. Zhao, M. Wei, C. Liu, C. Hao, C.-W. Qiu, and T. J. Cui, Intelligent metasurface imager and recognizer, *Light: Sci. Appl.* **8**, 97 (2019).
- [39] L. Li, H. Ruan, C. Liu, Y. Li, Y. Shuang, A. Alù, C.-W. Qiu, and T. J. Cui, Machine-learning reprogrammable metasurface imager, *Nat. Commun.* **10**, 1082 (2019).
- [40] P. del Hougne, M. F. Imani, A. V. Diebold, R. Horstmeyer, and D. R. Smith, Learned integrated sensing pipeline: Reconfigurable metasurface transceivers as trainable physical layer in an artificial neural network, *Adv. Sci.* **7**, 1901913 (2020).
- [41] H.-Y. Li, H.-T. Zhao, M.-L. Wei, H.-X. Ruan, Y. Shuang, T. J. Cui, P. Del Hougne, and L. Li, Intelligent electromagnetic sensing with learnable data acquisition and processing, *Patterns* **1**, 100006 (2020).
- [42] P. del Hougne, Robust position sensing with wave fingerprints in dynamic complex propagation environments, *Phys. Rev. Res.* **2**, 043224 (2020).
- [43] M. del Hougne, S. Gigan, and P. del Hougne, Deeply Subwavelength Localization with Reverberation-Coded Aperture, *Phys. Rev. Lett.* **127**, 043903 (2021).
- [44] C. Qian, B. Zheng, Y. Shen, L. Jing, E. Li, L. Shen, and H. Chen, Deep-learning-enabled self-adaptive microwave cloak without human intervention, *Nat. Photonics* **14**, 383 (2020).
- [45] T. Shan, X. Pan, M. Li, S. Xu, and F. Yang, Coding programmable metasurfaces based on deep learning techniques, *IEEE J. Emerg. Sel. Top. Circuits Syst.* **10**, 114 (2020).
- [46] J. A. Hart, T. M. Antonsen, and E. Ott, Effect of short ray trajectories on the scattering statistics of wave chaotic systems, *Phys. Rev. E* **80**, 041109 (2009).
- [47] J.-H. Yeh, J. A. Hart, E. Bradshaw, T. M. Antonsen, E. Ott, and S. M. Anlage, Experimental examination of the effect of short ray trajectories in two-port wave-chaotic scattering systems, *Phys. Rev. E* **82**, 041114 (2010).
- [48] E. Ott, *Chaos in Dynamical Systems* (Cambridge University Press, Cambridge, U.K.; New York, 2002), 2nd ed.
- [49] F. Haake, *Quantum Signatures of Chaos* (Springer, Berlin, Heidelberg, 2010).
- [50] R. Höhmann, U. Kuhl, H.-J. Stöckmann, L. Kaplan, and E. J. Heller, Freak Waves in the Linear Regime: A Microwave Study, *Phys. Rev. Lett.* **104**, 093901 (2010).
- [51] R. Schäfer, H.-J. Stöckmann, T. Gorin, and T. H. Seligman, Experimental Verification of Fidelity Decay: From Perturbative to Fermi Golden Rule Regime, *Phys. Rev. Lett.* **95**, 184102 (2005).
- [52] B. T. Taddese, J. Hart, T. M. Antonsen, E. Ott, and S. M. Anlage, Sensor based on extending the concept of fidelity to classical waves, *Appl. Phys. Lett.* **95**, 114103 (2009).
- [53] B. T. Taddese, T. M. Antonsen, E. Ott, and S. M. Anlage, Sensing small changes in a wave chaotic scattering system, *J. Appl. Phys.* **108**, 114911 (2010).
- [54] R. L. Schmid, D. B. Shrekenhamer, O. F. Somerlock, A. C. Malone, T. A. Sleasman, and R. S. Awadallah, in *2020 IEEE Radio and Wireless Symposium (RWS)* (IEEE, San Antonio, TX, USA, 2020), p. 91.
- [55] Y. Deng, S. Ren, K. Fan, J. M. Malof, and W. J. Padilla, Neural-adjoint method for the inverse design of all-dielectric metasurfaces, *Opt. Express* **29**, 7526 (2021).
- [56] D. P. Bertsekas, *Reinforcement Learning and Optimal Control* (Athena Scientific, Boston, 2019).
- [57] O. Moran, P. Caramazza, D. Faccio, and R. Murray-Smith, in *Advances in Neural Information Processing Systems*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Inc., 2018), Vol. 31, p. 3280.
- [58] P. Caramazza, O. Moran, R. Murray-Smith, and D. Faccio, Transmission of natural scene images through a multi-mode fibre, *Nat. Commun.* **10**, 2029 (2019).
- [59] S. Popoff, complexpytorch, <https://github.com/wavefrontshaping/complexPyTorch> (2019).
- [60] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, Deep Complex Networks, *arXiv:1705.09792* [cs] (2018).
- [61] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Boston, MA, USA, 2015), p. 1.
- [62] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Las Vegas, NV, USA, 2016), p. 2818.
- [63] B. T. Taddese, G. Gradoni, F. Moglie, T. M. Antonsen, E. Ott, and S. M. Anlage, Quantifying volume changing perturbations in a wave chaotic system, *New J. Phys.* **15**, 023025 (2013).
- [64] Nuand, bladeRF, <https://www.nuand.com>.
- [65] S. Han, H. Mao, and W. J. Dally, Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding, *arXiv:1510.00149* [cs] (2016).
- [66] NVIDIA, Jetson Embedded Systems, <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/>.
- [67] H. W. Babcock, The possibility of compensating astronomical seeing, *Publ. Astron. Soc. Pacific* **65**, 229 (1953).
- [68] R. A. Duffner, *The Adaptive Optics Revolution, A History* (University of New Mexico Press, Albuquerque, NM, 2009).
- [69] J. W. Hardy, *Adaptive Optics for Astronomical Telescopes* (Oxford University Press, New York, NY, 1998).
- [70] M. J. Booth, Adaptive optics in microscopy, *Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci.* **365**, 2829 (2007).
- [71] C. B. Hogge, in *Optics in Adverse Environments I*, edited by E. B. G. and H. V. Winsor, International Society for Optics and Photonics (SPIE, 1978), Vol. 0121, p. 140.
- [72] H. J. White, D. W. Gough, R. Merry, and S. Patrick, in *Advanced Free-Space Optical Communications Techniques and Technologies*, edited by M. Ross and A. M. Scott, International Society for Optics and Photonics (SPIE, 2004), Vol. 5614, p. 119.

- [73] M. T. Gruneisen, M. L. Eickhoff, S. C. Newey, K. E. Stoltenberg, J. F. Morris, M. Bareian, M. A. Harris, D. W. Oesch, M. D. Oliker, M. B. Flanagan, B. T. Kay, J. D. Schiller, and R. N. Lanning, Adaptive-Optics-Enabled Quantum Communication: A Technique for Day-time Space-To-Earth Links, *Phys. Rev. Appl.* **16**, 014067 (2021).
- [74] P. S. Salter and M. J. Booth, Adaptive optics in laser processing, *Light: Sci. Appl.* **8**, 110 (2019).
- [75] R. H. Freeman and J. E. Pearson, Deformable mirrors for all seasons and reasons, *Appl. Opt.* **21**, 580 (1982).
- [76] M. A. Ealey and J. Wellman, in *Precision Engineering and Optomechanics*, edited by D. Vukobratovich, International Society for Optics and Photonics (SPIE, 1989), Vol. 1167, p. 66.
- [77] M. A. Ealey, in *Active and Adaptive Optical Components*, edited by M. A. Ealey, International Society for Optics and Photonics (SPIE, 1992), Vol. 1543, p. 346.
- [78] A. Wirth, J. Cavaco, T. Bruno, and K. M. Ezzo, in *High-Power, High-Energy, and High-Intensity Laser Technology; and Research Using Extreme Light: Entering New Frontiers with Petawatt-Class Lasers*, edited by G. Korn, L. O. Silva, and J. Hein, International Society for Optics and Photonics (SPIE, 2013), Vol. 8780, p. 122.
- [79] M. Horenstein, T. Bifano, S. Pappas, J. Perreault, and R. Krishnamoorthy-Mali, Real time optical correction using electrostatically actuated mems devices, *J. Electrostat.* **46**, 91 (1999). selected Papers from the 1998 Joint ESA/IEJ Symposium on Electrostatics
- [80] J. A. Perreault, T. G. Bifano, B. M. Levine, and M. N. Horenstein, Adaptive optic correction using micro-electro-mechanical deformable mirrors, *Opt. Eng.* **41**, 561 (2002).
- [81] D. Casasent, Spatial light modulators, *Proc. IEEE* **65**, 143 (1977).
- [82] G. Esposito, S. amd Brusa, and D. Bonaccini, in *Proceedings of the ICO-16 (International Commission for Optics) Satellite Conference on Active and adaptive optics* (ESO, 1993), p. 289.
- [83] I. M. Vellekoop and A. P. Mosk, Universal Optimal Transmission of Light through Disordered Materials, *Phys. Rev. Lett.* **101**, 120601 (2008).
- [84] D. Veksler, E. Maguid, N. Shitrit, D. Ozeri, V. Kleiner, and E. Hasman, Multiple wavefront shaping by metasurface based on mixed random antenna groups, *ACS Photonics* **2**, 661 (2015).
- [85] P. del Hougne, M. Fink, and G. Lerosey, Shaping Microwave Fields Using Non-Linear Unsolicited Feedback: Application to Enhanced Energy Harvesting, *Phys. Rev. Appl.* **8**, 061001 (2017).
- [86] L. Ljung, *System Identification: Theory for the User* (Pearson, Upper Saddle River, NJ, 1997), 2nd ed.
- [87] R. K. Tyson and B. W. Frazier, *Field Guide to Adaptive Optics*, Field Guide Series (SPIE Press, Bellingham, WA, 2012), 2nd ed.
- [88] A. Neubauer, On the ill-posedness and convergence of the shack-hartmann based wavefront reconstruction, *J. Inverse and Ill-Posed Probl.* **18**, 551 (2010).
- [89] C. R. Vogel, *Computational Methods for Inverse Problems* (SIAM, Philadelphia, 2006).
- [90] P. C. Hansen, J. G. Nagy, and D. P. O’Leary, *Deblurring Images: Matrices, Spectra, and Filtering* (SIAM, Philadelphia, 2006).
- [91] V. Hutterer and R. Ramlau, Nonlinear wavefront reconstruction methods for pyramid sensors using landweber and landweber-kaczmarz iterations, *Appl. Opt.* **57**, 8790 (2018).
- [92] V. Hutterer, R. Ramlau, and I. Shatokhina, Real-time adaptive optics with pyramid wavefront sensors: Part II. accurate wavefront reconstruction using iterative methods, *Inverse Probl.* **35**, 045008 (2019).
- [93] S. Hemmady, X. Zheng, T. M. Antonsen, E. Ott, and S. M. Anlage, Universal statistics of the scattering coefficient of chaotic microwave cavities, *Phys. Rev. E* **71**, 056215 (2005).
- [94] S. Hemmady, X. Zheng, J. Hart, T. M. Antonsen, E. Ott, and S. M. Anlage, Universal properties of two-port scattering, impedance, and admittance matrices of wave-chaotic systems, *Phys. Rev. E* **74**, 036213 (2006).
- [95] T. R. O’Meara, The multidither principle in adaptive optics, *J. Opt. Soc. Am.* **67**, 306 (1977).
- [96] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control* (Wiley, Hoboken, NJ, 2003).
- [97] M. A. Vorontsov and V. Kolosov, Target-in-the-loop beam control: Basic considerations for analysis and wave-front sensing, *J. Opt. Soc. Am. A* **22**, 126 (2005).
- [98] M. A. Vorontsov, G. W. Carhart, and J. C. Ricklin, Adaptive phase-distortion correction based on parallel gradient-descent optimization, *Opt. Lett.* **22**, 907 (1997).
- [99] M. A. Vorontsov and V. P. Sivokon, Stochastic parallel-gradient-descent technique for high-resolution wave-front phase-distortion correction, *J. Opt. Soc. Am. A* **15**, 2745 (1998).
- [100] T. Weyrauch and M. A. Vorontsov, Free-space laser communications with adaptive optics: Atmospheric compensation experiments, *J. Opt. Fiber Commun. Rep.* **1**, 355 (2004).
- [101] V. Rodoplu and T. H. Meng, Bits-per-joule capacity of energy-limited wireless networks, *IEEE Trans. Wirel. Commun.* **6**, 857 (2007).
- [102] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, Reconfigurable intelligent surfaces for energy efficiency in wireless communication, *IEEE Trans. Wirel. Commun.* **18**, 4157 (2019).
- [103] Z. B. Drikas, J. Gil Gil, S. K. Hong, T. D. Andreadis, J.-H. Yeh, B. T. Taddese, and S. M. Anlage, Application of the random coupling model to electromagnetic statistics in complex enclosures, *IEEE Trans. Electromagn. Compat.* **56**, 1480 (2014).
- [104] S. Hemmady, X. Zheng, E. Ott, T. M. Antonsen, and S. M. Anlage, Universal Impedance Fluctuations in Wave Chaotic Systems, *Phys. Rev. Lett.* **94**, 014102 (2005).
- [105] J.-H. Yeh, J. A. Hart, E. Bradshaw, T. M. Antonsen, E. Ott, and S. M. Anlage, Universal and nonuniversal properties of wave-chaotic scattering systems, *Phys. Rev. E* **81**, 025201 (2010).
- [106] X. Zheng, T. M. Antonsen, and E. Ott, Statistics of impedance and scattering matrices in chaotic microwave



- cavities: Single channel case, *Electromagnetics* **26**, 3 (2006).
- [107] X. Zheng, T. M. Antonsen, and E. Ott, Statistics of impedance and scattering matrices of chaotic microwave cavities with multiple ports, *Electromagnetics* **26**, 37 (2006).
- [108] M. V. Berry, Regular and irregular semiclassical wavefunctions, *J. Phys. A: Math. General* **10**, 2083 (1977).
- [109] S. Hemmady, J. Hart, X. Zheng, T. M. Antonsen, E. Ott, and S. M. Anlage, Experimental test of universal conductance fluctuations by means of wave-chaotic microwave cavities, *Phys. Rev. B* **74**, 195326 (2006).
- [110] B. Addissie, J. Rodgers, and T. Antonsen, Extraction of the coupling impedance in overmoded cavities, *Wave Motion Innovations in Wave Modelling II* **87**, 123 (2019).
- [111] J. Gil Gil, Z. B. Drikas, T. D. Andreadis, and S. M. Anlage, Prediction of induced voltages on ports in complex, three-dimensional enclosures with apertures, using the random coupling model, *IEEE Trans. Electromagn. Compat.* **58**, 1535 (2016).
- [112] D. M. Pozar, *Microwave Engineering* (Wiley, Hoboken, NJ, 2011), 4th ed.
- [113] S. Hemmady, T. M. Antonsen, E. Ott, and S. M. Anlage, Statistical prediction and measurement of induced voltages on components within complicated enclosures: A wave-chaotic approach, *IEEE Trans. Electromagn. Compat.* **54**, 758 (2012).
- [114] J. Yeh, E. Ott, T. Antonsen, and S. Anlage, Fading statistics in communications - a random matrix approach, *Acta Phys. Polonica A* **120**, A (2011).
- [115] J.-H. Yeh, T. M. Antonsen, E. Ott, and S. M. Anlage, First-principles model of time-dependent variations in transmission through a fluctuating scattering environment, *Phys. Rev. E* **85**, 015202 (2012).
- [116] M. Simon and M.-S. Alouini, *Digital Communication Over Fading Channels* (Wiley-Interscience, Hoboken, NJ, 2005), 2nd ed.
- [117] K. Ward, R. Tough, and S. Watts, *Sea Clutter, Scattering, the K Distribution and Radar Performance* (Institution of Engineering and Technology, London, United Kingdom, 2013), 2nd ed.
- [118] C. L. Holloway, H. A. Shah, R. J. Pirkel, W. F. Young, D. A. Hill, and J. Ladbury, Reverberation chamber techniques for determining the radiation and total efficiency of antennas, *IEEE Trans. Antennas Propag.* **60**, 1758 (2012).
- [119] Y. B. Li, L. L. Li, B. B. Xu, W. Wu, R. Y. Wu, X. Wan, Q. Cheng, and T. J. Cui, Transmission-type 2-bit programmable metasurface for single-sensor and single-frequency microwave imaging, *Sci. Rep.* **6**, 23731 (2016).
- [120] L. Zhang, X. Q. Chen, S. Liu, Q. Zhang, J. Zhao, J. Y. Dai, G. D. Bai, X. Wan, Q. Cheng, G. Castaldi, V. Galdi, and T. J. Cui, Space-time-coding digital metasurfaces, *Nat. Commun.* **9**, 4334 (2018).
- [121] L. Shao, W. Zhu, M. Y. Leonov, and I. D. Rukhlenko, Dielectric 2-bit coding metasurface for electromagnetic wave manipulation, *J. Appl. Phys.* **125**, 203101 (2019).
- [122] S. Iqbal, S. Liu, G. D. Bai, M. Furqan, H. A. Madni, and T. J. Cui, Dual-band 2-bit coding metasurface for multifunctional control of both spatial waves and surface waves, *J. Opt. Soc. Am. B* **36**, 293 (2019).
- [123] Y. D. Chong, L. Ge, H. Cao, and A. D. Stone, Coherent Perfect Absorbers: Time-Reversed Lasers, *Phys. Rev. Lett.* **105**, 053901 (2010).
- [124] L. Chen, T. Kottos, and S. M. Anlage, Perfect absorption in complex scattering systems with or without hidden symmetries, *Nat. Commun.* **11**, 5826 (2020).
- [125] P. Ambichl, A. Brandstötter, J. Böhm, M. Kühmayer, U. Kuhl, and S. Rotter, Focusing inside Disordered Media with the Generalized Wigner-Smith Operator, *Phys. Rev. Lett.* **119**, 033903 (2017).
- [126] D. A. Hill, *Electromagnetic Fields in Cavities, Deterministic and Statistical Theories*, IEEE Press Series on Electromagnetic Wave Theory (Wiley, Hoboken, N.J., 2009).
- [127] Y. Bengio, Deep learning of representations for unsupervised and transfer learning, *Proc. Machine Learning Res.* **27**, 17 (2012).
- [128] Y. Bengio, Deep Learning of Representations: Looking Forward, [arXiv:1305.0445](https://arxiv.org/abs/1305.0445) [cs] (2013).
- [129] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, in *Advances in Neural Information Processing Systems*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Curran Associates, Inc., 2014), Vol. 27, p. 2924.
- [130] K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) [cs] (2015).
- [131] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, in *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 70, edited by D. Precup and Y. W. Teh (PMLR, International Convention Centre, Sydney, Australia, 2017), p. 2847.
- [132] T. J. Sejnowski, The unreasonable effectiveness of deep learning in artificial intelligence, *Proc. Natl. Acad. Sci.* **117**, 30033 (2020).
- [133] D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* **1**, 16 (1997).
- [134] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, in *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 28, edited by S. Dasgupta and D. McAllester (PMLR, Atlanta, Georgia, USA, 2013), p. 1139.
- [135] Y. Nesterov, A method of solving a convex programming problem with convergence rate  $o(1/k^2)$ , *Soviet Math. Doklady* **27**, 372 (1983).
- [136] L. N. Smith, A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay, [arXiv:1803.09820](https://arxiv.org/abs/1803.09820) [cs, stat] (2018).
- [137] F. a. o. Chollet, Keras, <https://github.com/fchollet/keras> (2015).
- [138] M. Abadi *et al.*, TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, [arXiv:1603.04467](https://arxiv.org/abs/1603.04467) [cs] (2016).
- [139] C. Trabelsi, Deep complex networks, [https://github.com/ChihebTrabelsi/deep\\_complex\\_networks](https://github.com/ChihebTrabelsi/deep_complex_networks) (2018).
- [140] P. Virtue, Complex caffe, <https://github.com/pvirtue/caffe/tree/complex> (2018).



- [141] A. Paszke *et al.*, in *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), p. 8024.
- [142] S. Ioffe and C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, *Proc. 32nd Int. Conf. Machine Learning*, PMLR **37**, 448 (2015).
- [143] K. He, X. Zhang, S. Ren, and J. Sun, in *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), p. 1026.
- [144] A. Araujo, W. Norris, and J. Sim, Computing receptive fields of convolutional neural networks, [Distill](#) (2019).
- [145] C. L. Holloway, H. A. Shah, R. J. Pirkel, K. A. Remley, D. A. Hill, and J. Ladbury, Early time behavior in reverberation chambers and its effect on the relationships between coherence bandwidth, chamber decay time, RMS delay spread, and the chamber buildup time, [IEEE Trans. Electromagn. Compat.](#) **54**, 714 (2012).
- [146] C. J. C. H. Watkins, Ph.D., King's College, 1989.
- [147] M. Bukov, A. G. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Reinforcement Learning in Different Phases of Quantum Control, *Phys. Rev. X* **8**, 031086 (2018).
- [148] J. Mackeprang, D. B. R. Dasari, and J. Wrachtrup, A reinforcement learning approach for quantum state engineering, [Quantum Mach. Intelligence](#) **2**, 5 (2020).
- [149] M. M. Wauters, E. Panizon, G. B. Mbeng, and G. E. Santoro, Reinforcement-learning-assisted quantum optimization, [Phys. Rev. Res.](#) **2**, 033446 (2020).
- [150] H. van Hasselt, A. Guez, and D. Silver, Deep Reinforcement Learning with Double Q-learning, [arXiv:1509.06461 \[cs\]](#) (2015).
- [151] M. E. Taylor and P. Stone, Transfer learning for reinforcement learning domains: A survey, *J. Mach. Learn. Res.* **10**, 1633 (2009).