Physics-informed learning in artificial electromagnetic materials

View Online　　Export Citation　　CrossMark

Y. Deng,[1] (iD) K. Fan,[2] (iD) B. Jin,[2] (iD) J. Malof,[3] (iD) and W. J. Padilla[1,a] (iD)

AFFILIATIONS

[1]Pratt School of Engineering, Duke University, Durham, North Carolina 27710, USA

[2]Research Institute of Superconductor Electronics (RISE) & Key Laboratory of Optoelectronic Devices and Systems with Extreme Performances of MOE, School of Electronic Science and Engineering, Nanjing University, Nanjing, China

[3]Department of Electrical and Computer Engineering, University of Missouri, Columbia, Missouri 65201, USA

[a]Author to whom correspondence should be addressed: willie.padilla@duke.edu.
URL: https://padillalab.pratt.duke.edu/welcome-lab

ABSTRACT

The advent of artificial intelligence—deep neural networks (DNNs) in particular—has transformed traditional research methods across many disciplines. DNNs are data driven systems that use large quantities of data to learn patterns that are fundamental to a process. In the realm of artificial electromagnetic materials (AEMs), a common goal is to discover the connection between the AEM's geometry and material properties to predict the resulting scattered electromagnetic fields. To achieve this goal, DNNs usually utilize computational electromagnetic simulations to act as ground truth data for the training process, and numerous successful results have been shown. Although DNNs have many demonstrated successes, they are limited by their requirement for large quantities of data and their lack of interpretability. The latter results because DNNs are black-box models, and therefore, it is unknown how or why they work. A promising approach which may help to mitigate the aforementioned limitations is to use physics to guide the development and operation of DNNs. Indeed, this physics-informed learning (PHIL) approach has seen rapid development in the last few years with some success in addressing limitations of conventional DNNs. We overview the field of PHIL and discuss the benefits of incorporating knowledge into the deep learning process and introduce a taxonomy that enables us to categorize various types of approaches. We also summarize deep learning principles which are critical to PHIL understanding and the Appendix covers some of the physics of AEMs. A few specific PHIL works are highlighted and serve as examples of various approaches. Finally, we provide an outlook detailing where the field is currently and what we can expect in the future.

Published under an exclusive license by AIP Publishing. https://doi.org/10.1063/5.0232675

TABLE OF CONTENTS

I. INTRODUCTION

Deep learning (DL) is a sub-field of artificial intelligence wherein the goal is to automate the process of identifying patterns in data. DL has generated tremendous excitement in recent years due to its capability to accurately model (i.e., uncover patterns) in highly complex real-world datasets, including the modeling and design of artificial electromagnetic materials (AEMs), such as electromagnetic metamaterials, metasurfaces, photonic crystals, artificial dielectrics, and plasmonics.[1–4] Within the AEM community, DL has found success performing various tasks: for example, predicting AEM scattering,[5] AEM material design,[6,7] and physics discovery.[8] Perhaps the most widespread application of DL has been to accelerate the prediction of electromagnetic scattering based on AEM properties, wherein DL-based methods have been shown to accurately predict the scattering of highly complex AEM materials and do so significantly faster than alternative approaches such as computational electromagnetic simulations (CEMSs) (e.g., by a factor of 10^5).[9] Consequently, DL has garnered significant attention within the AEM community,[1–3] evidenced in part by the rapidly growing number of publications on the topic.[1] Despite DL's success, it still suffers from several limitations that significantly undermine its impact. One of the most important of these impediments is its requirement for relatively large quantities of *data*. In practice, DL refers to a broad class of algorithms, called deep neural networks (DNNs), which are *trained* to perform a particular task (e.g., predicting AEM scattering based on its material properties) using a collection of training data. These data are typically procured through physical experimentation, or more often, through CEMS. Recent research applications have involved dataset sizes on the order of $10^4 - 10^5$, making the procurement of training data from CEMS computationally intensive and significantly undermining the value of using DNNs. Furthermore, these computational costs typically grow rapidly with respect to the complexity of the AEM problem under consideration (e.g., its dimensionality and the non-linearity of underlying relationships in the data), creating a *data bottleneck* that ultimately renders many modern problems impracticable for DNNs. This problem transcends AEMs and represents a significant limitation of DNNs across science and engineering.

One emerging strategy for mitigating the data bottleneck problem of DNNs is by incorporating existing physical knowledge into their design and training. This is because conventional DNNs often produce models that are inconsistent with existing physical laws and constraints. By incorporating physical knowledge, we can exclude candidate DNN models from the training process that exhibit this behavior, improving the quality of the final solution. For example, we may wish to train a DNN to predict AEM scattering, encoded in the form of an absorption or reflection spectrum, which means that we can preferentially select DNNs that make predictions consistent with the laws of physics. In another example, we may know apriori that the training data are governed by a partial differential equation (PDE), which implies that we can exclude any DNNs that do not also conform to that PDE. We refer to this general strategy of utilizing prior physical knowledge to guide DNN-based modeling as *physics-informed learning*

(PHIL). Recent research has shown that the net effect of PHIL is often to either (i) significantly reduce the training data needed to infer good DNN parameters, (ii) drastically reduce the model size, or (iii) improve the accuracy and generalization of the trained DNN.

In this work, we discuss the recent promising progress and results of PHIL. The PDE example above illustrates just one source of physical knowledge, but recent research has revealed many potential sources of physics information that can guide the DNN modeling process: examples include algebraic equations, differential equations, knowledge graphs, simulations, and human feedback. Additionally, there are a variety of different ways to incorporate this information into the DNN modeling process. Figure 1 shows a high-level schematic of the deep learning workflow, and we specify three regions where knowledge K may be incorporated. In this paper, we taxonomize and review PHIL methods that fall within the context of AEM. Our aim is that this review serves as a guide to the AEM community in this exciting new field. This involves creating a taxonomy of existing methods, explaining how these methods compare and contrast, and identifying open challenges. Although our review focuses on applying PHIL to AEM, the methods, taxonomy, and approaches discussed here have broad applicability. We encourage scientists and engineers across various disciplines to explore PHIL.

This paper is organized as follows. In Sec. II, we discuss the principles of machine learning/deep learning which are essential for understanding PHIL. For readers who are already versed in deep learning, they can skip this section. In Sec. III, we discuss the benefits of incorporating domain knowledge into the deep learning process and introduce a taxonomy that enables us to categorize various types of PHIL approaches. Next in Sec. IV, we overview specific examples from the literature where physics has been integrated into deep neural networks and highlight a few specific examples. Finally, in Sec. V, we provide a forecast on where the field is currently headed and discuss some
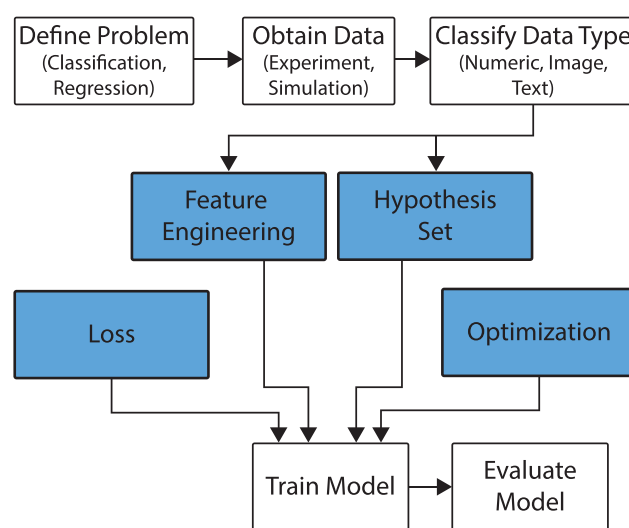

FIG. 1. Schematic depicting the deep learning workflow. The process begins in the top left where the problem is defined, before gathering and classifying a dataset *D*. The central blue boxed region shows the areas in which knowledge can be incorporated. The model is then trained and evaluated.

exciting opportunities for near future research. The physics of AEMs is summarized in the Appendix.

II. REVIEW OF DEEP LEARNING

In this section, we provide a brief review of deep learning, with a focus on key concepts for understanding PHIL. This is not a comprehensive review, and we assume the reader has some familiarity with general machine learning concepts. We begin by outlining a typical DL workflow in AEM, which is primarily intended to remind readers of core concepts in DL and establish basic terminology. We then conceptualize DL as a semi-automated implementation of the scientific method, making it more apparent how physics knowledge can be integrated into DL.

A. A typical DL workflow within AEM

DL and machine learning methods are often placed into one of three major categories: supervised, unsupervised, and reinforcement. Each of these categories has been investigated in the context of AEM, but supervised learning comprises the large majority of existing work, and therefore, we focus on it in this section. In supervised learning, we assume the availability of a dataset of the form $D = (x_i, y_i)_{i=1}^N$ and that there is some relationship present in these data so that we can make predictions of each y_i based on each x_i. In an AEM context, x is a numerical vector that often quantifies experimental parameters such as AEM geometry (e.g., height, width, depth, and periodicity), its bulk construction material, or the illumination (incident angle, polarization, frequency, etc.) of the material. By contrast, the variable y is a numerical vector that often represents some experimental outcomes, such as the electric/magnetic field values,[4] or absorption,[10] reflection,[11] and transmission[12] spectra, measured at specific frequencies. The goal of supervised learning is to build a *model* of the form $y = f(x)$ based on the example (x_i, y_i) pairs in D, which we can then use to make predictions of y for novel settings of x.

DL methods have proven highly successful in solving supervised learning problems, including AEM problems. In practice, DL usually takes the form of a deep neural network (DNN), which is a broad class of methods for solving general machine learning problems, including supervised learning problems. A DNN is a *parameterized* function of the form f_θ, where θ represents some parameters of the DNN that influence the relationship between its input and output. By altering the setting of θ, DNNs can approximate a wide array of highly complex input/output relationships. A typical DNN workflow is illustrated in Fig. 2, where the goal is to find θ such that f_θ makes accurate predictions of y for values of x that are present in D, as well as novel settings of x that are not in D.

The workflow begins in step (1) with the collection of the dataset D, which may be collected via physical experiments, but is more often obtained via CEMS. In step (2), a subset of these data, $D_{tr} \subset D$, termed the *training dataset*, is utilized to find an effective setting of θ through DNN *training*. During training, the setting of θ is iteratively adjusted to minimize the prediction error of f_θ on all of the input/output pairs available in D_{tr}. Prediction error is measured through a user-chosen measurement, termed a *loss function*, denoted $\mathscr{L}$, which takes a ground truth target value $y_i \in D_{tr}$ and the prediction—denoted by $(\hat{\ })$—made by the model $\hat{y}_i = f_\theta(x_i)$ and produces a scalar value, e_i, indicating the relative disagreement between y_i and $\hat{y}_i$. The most common choice for $\mathscr{L}$ is squared error, in which case the model is trained
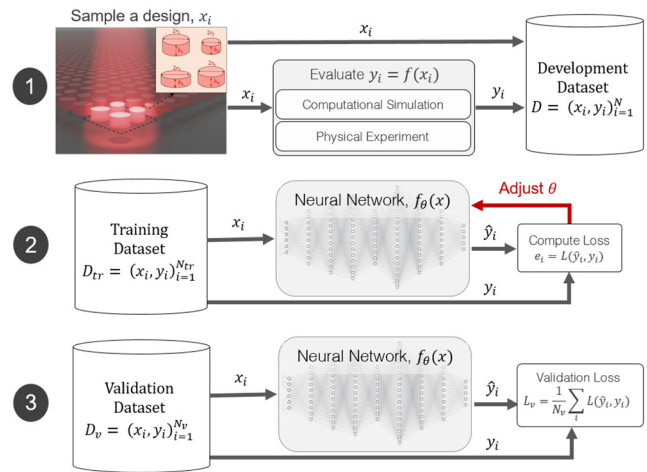

FIG. 2. Illustrations of a typical supervised DL workflow. In step 1, the design parameters x_i of an AEM are evaluated using CEMS (or via an experiment) and together with the output y_i constitute the development dataset D. In step 2, a portion of the dataset—termed the training dataset D_{tr}—is used for training of a neural network. After the neural network achieves an acceptable value of loss, in step 3 a validation dataset D_v, from the development dataset D, is used as a final measure of the networks performance.

to minimize the average error of f_θ over the training data, commonly referred to as the mean square error (MSE). The adjustments to the parameters are made using a procedure called stochastic gradient descent, or its variations (e.g., Adam and RMSprop), which utilize the derivative of the loss with respect to the model parameters, $dL/d\theta$ to determine how to adjust θ in a way that will reduce the loss.

Once trained, f_θ can make predictions of y for settings of x that were not present in D_{tr}. In step (3), our goal is to estimate the loss (or other error metric) of the DNN's predictions. It is well known that the model's loss on the training data is optimistic, and therefore, it is best practice to set aside a subset of data, termed the *validation dataset*, $D_{val} \subset D$, which must be disjoint with respect to D_{tr}, that can be used for the purpose of accurately estimating the loss of the trained DNN. Notably, there are some parameters of the DNN that are not inferred during training and must instead be chosen by the designer; these are known as *hyperparameters*. It is common practice to train multiple DNNs, each with different settings of the hyperparameters. In this case, the validation dataset can be utilized to compare the error rate of the models and choose the best one. However, any loss computed on the validation dataset becomes optimistic, and in these cases, a third disjoint (holdout) dataset should be used to estimate accuracy, called the *testing dataset*.

B. Deep learning as the scientific method

DL can be broadly conceived as one means to semi-automate the scientific method,[13] wherein the goal is to identify a hypothesis (e.g., a mathematical model) that best explains some observed data. This conception of DL can be seen by precisely defining three key components of the scientific method: the *hypothesis space* ($\mathscr{H}$), the *loss function* ($\mathscr{L}$), and the *Optimizer* ($\mathscr{O}$).

1. The hypothesis space, $\mathscr{H}$

A scientist may consider several hypotheses to explain some observed data. In DL, we formalize this notion by precisely defining *all* mathematical models that we will consider when attempting to model our observed data. For example, in the DL workflow in Sec. II A, we searched over settings of θ that would cause f_θ to obtain accurate predictions on the training dataset. Therefore, $\mathscr{H}$ comprised the set of all functions, f_θ, such that $\theta \in \mathbb{R}^W$, where W is the number of model parameters. Therefore, the functions considered in $\mathscr{H}$ depend on θ, and we can expand (or contract) the set of functions in $\mathscr{H}$ by increasing (or decreasing) the size of W. The set $\mathscr{H}$ is also greatly influenced by the functional form of f_θ, which represents the *architecture* of the DNN. Given the same setting of W, we can obtain very different $\mathscr{H}$ depending on whether we use a feedforward DNN, a recurrent DNN, a transformer, or another model. DNN hyperparameters such as the model's width (parameters per layer) and depth (number of layers) can also influence both the functional form of f_θ, as well as the size of W. Another major influence on $\mathscr{H}$ is the choice of input variables, or *features*, x, that are provided as input to the DNN. The designer must not only choose, which explanatory variables to include as input, but also whether they should be processed in some manner before being provided to the model. Processing may include transformations of individual features (e.g., taking their logarithm and square), creation of new features based on transformations of existing features (e.g., products and ratios of features), or normalization of features (e.g., removing the mean value of each feature). These design choices, broadly referred to as *feature engineering*, can have a major impact on the hypothesis space.

The scientific method works best when we have few hypotheses to consider, and where at least one of the candidate hypotheses is a good model for the data. By contrast, the $\mathscr{H}$ associated with DNNs is extremely large, which is useful for general-purpose modeling; this makes it likely that $\mathscr{H}$ will contain at least one good hypothesis for any application encountered. However, when working on a specific application we can often remove hypotheses, i.e., we can exclude hypotheses from $\mathscr{H}$ that are inconsistent with existing physical laws or knowledge, as well as add hypotheses to $\mathscr{H}$ that may be well motivated based on the existing knowledge.

2. The loss function, $\mathscr{L}$

Given a set of hypotheses, we must rank them and then choose the one that best "explains" or "fits" the observed data. The loss function, or just "loss," makes it possible to automate this process, by precisely defining what we mean when we say that a model "explains" or "fits" the data well or does so better than another model. The loss, denoted $\mathscr{L}$, is a function that can take any hypothesis in $\mathscr{H}$ as input, and potentially other quantities (e.g., the training data), and then assigns a number indicating that hypothesis' relative ability to "explain" the data. As a convention, lower loss values are assumed to indicate better models. Typically, loss functions involve some measure of model error on the training data, such as the squared error illustrated in Fig. 2. Another common strategy in ML and DL is to assign a low loss to models that have both low error *and low complexity*, i.e., when two models have similar error, we tend to prefer the simpler model. In some applications, such as physics, we may know that the model must possess some additional properties, which can be incorporated into the loss. One of the most common examples is the so-called physics-informed neural network (PINN) framework, wherein we know that our DNN model f_θ is the solution to a differential equation. In this case, we can increase the loss of models that do not obey our differential operator.

3. Optimizer, $\mathscr{O}$

With a well-defined loss function and hypothesis space, it is possible in principle to *automatically* assign a loss value to each hypothesis in $\mathscr{H}$ and then identify the model with the lowest loss. However, the hypothesis spaces utilized in DL are typically large, far surpassing what a brute force approach could reasonably search. Therefore, another key component of DL systems is the Optimizer, denoted $\mathscr{O}$, which is a well-defined process for efficiently searching over candidate models for the one with the lowest loss. By far, the most commonly used optimizer for DL is stochastic gradient descent, and its variants such as Adam, RMSProp, and Nesterov. In principle, one could utilize physics to improve $\mathscr{O}$; however, we find no instances of this type of PHIL.

III. PHYSICS-INFORMED LEARNING

We begin this section by first providing a broad overview of PHIL. At a high level, PHIL could be defined as any method which integrates physics derived from *first principles* into a learning process. By this definition, PHIL has been a fundamental part of scientific progress since the dawn of civilization. However, we instead narrow the scope of PHIL to the era of modern computing, explicitly focusing on machine-based PHIL. In that case, we observe a spectrum similar to the discussion in Ref. 14, ranging from *physics model-based learning* to fully *data-driven learning* approaches. Physics model-based learning is at one extreme of the spectrum, strictly grounded in physics and typically formulated through partial differential equations. In AEM applications, methods, including finite element methods, coupled mode theory, and circuit theory model-based learning approaches, adhere to the first principles of physics and maintain physical accuracy as long as the assumptions and approximations made during their derivation remain valid. A key advantage of model-based approaches is their ability to operate data-free, relying solely on physics for inference. Other extremes of the spectrum are data-driven approaches. The AEM community leaped from model-based methods to data-driven approaches as traditional model-based approaches struggled to handle increasingly complex problems.

While conventional deep learning methods have revolutionized numerous fields,[15–17] there are, as mentioned, still many challenges that lie ahead.[1,3,18] A few critical issues stand out: the data bottleneck problem,[19] and the lack of interpretability—sometimes termed "black-box opacity,"[1,3,18,20] and the ever-increasing model sizes. Overcoming these limitations demands approaches that excel with limited data, efficient model architecture, and which uncover the underlying physics.[21–23] Addressing these difficulties has the potential to not only advance AEM applications but also deepen our fundamental understanding of physics through discovery.

In response to these challenges, AEM researchers have been trying to find "middle ground" approaches between explicit physics on one end, and data-driven methods on the other end, which span the spectrum of PHIL. These transformative approaches are driven by the pursuit of three crucial goals: enhanced data efficiency, reduced model size, and improved interpretability. Past works have highlighted the

benefit and necessity of moving away from black-box models toward more interpretable approaches.[24,25] PHIL models excel in data-constrained environments,[22] leverage domain-specific knowledge, and incorporate physics into the deep learning framework as shown in Fig. 1.[21,26,27] This not only yields accurate predictions with less data, or smaller model sizes, but also has the potential to uncover the underlying physics, granting us a deeper understanding of AEM phenomena.[1,3,6,28] Beyond enhancing data-driven models in scientific regression tasks, PHIL approaches have also been utilized to discover unknown physics from data.[29–32] Considering the limited demonstrations in the literature of physics discovery using PHIL on AEM problems, we will not further discuss that line of research here. Nonetheless, we encourage readers to explore this promising area.

Although PHIL can yield the aforementioned benefits, there can be significant implementation challenges—most prominently, the requirement of extensive domain expertise to merge physical knowledge with PHIL models accurately. The methods of integration—whether through feature engineering,[33] architectural adaptation,[34] or loss function modification[8]—demand attention and specialized knowledge. Such dependency on expertise can serve as a formidable barrier, constraining the applicability of PHIL to diverse fields. As we explore specific PHIL approaches, we aim to clarify these methods, adhering to the outlined taxonomy presented in Fig. 3. We will detail the process of integrating physical knowledge—originating from varied sources and presented in diverse representations—into PHIL at distinct stages of the deep learning workflow.

A. PHIL taxonomy

To provide a clear and concise description of the various types of PHIL, in Fig. 3, we characterize each PHIL method in terms of three key characteristics, which can then be utilized to taxonomize each approach.

(1) *Source* defines the origin of the physical knowledge incorporated into the model, which can be scientific knowledge derived from established physical laws or expert knowledge from domain experts.
(2) *Representation* of physical knowledge within PHIL models can be extremely diverse. This review examines physical knowledge in the form of differential equations, algebraic equations,

knowledge graphs, simulation results, and feedback from domain experts.
(3) *Integration* pertains to the specific steps in the deep learning workflow through which represented knowledge is integrated into the PHIL learning process. The methodology of integration varies significantly across different approaches.

In the following subsection, we will explore the specific characteristics of PHIL in detail and offer a comprehensive overview of its taxonomy.

B. PHIL source

A key desire of the application of PHIL to AEM entails the formulation of deep learning techniques that are consistent with Maxwell's equations (11)—presented in Sec. VI. These equations represent the fundamental laws dictating the physics of AEM and serve as the principal source of scientific knowledge. PHIL models are meticulously designed to be consistent with these principles, ensuring adherence to the equations that govern the dynamics of AEM systems.[35–42] This integration can be further refined by incorporating advanced physical theories such as the discrete dipole approximation[43] and temporal coupled mode theory,[36,44,45] enhancing the models' ability to not only conform to the foundational Maxwell's equations but also to capture more complex electromagnetic extensions.

We consider two types of PHIL sources: scientific knowledge (SK) and expert knowledge (EK). The above-mentioned Maxwell's equations and derived models are examples of scientific knowledge that can directly guide the training of a DNN. We note that expert knowledge is also derived from scientific knowledge, as expertise is developed through the mastery of scientific knowledge. However, we distinguish between these two knowledge sources to highlight instances of physical knowledge that do not explicitly stem from physical laws or that are not readily expressed through mathematical formulations. For example, although no exact analytical equations connect complex AEM geometry and its corresponding electromagnetic responses, an AEM practitioner can infer that specific geometry parameters play more critical roles in impacting the responses and input these parameters with more weight.

As illustrated in Sec. III D, it is apparent that most PHIL models applied to AEM are based on the scientific knowledge—most often Maxwell's equations. For a concise overview of the physical theories relevant to the scientific knowledge discussed in this review, we direct readers to Sec. VI.

C. PHIL representation

It is important to distinguish between the different types of PHIL representations. Different representations largely determine the best integration approach and the formalism used to integrate physics into the models.

Maxwell's equations are undoubtedly the most common representation of physics used in PHIL for AEMs. Examples such as PINNs[46] and other physics-informed works[47,48] have demonstrated the effectiveness of using differential equations in PHIL for AEMs. Similarly, various material models, such as the Drude–Lorentz model, presented in Sec. VI B, describe AEMs through algebraic equations for permittivity and permeability. When combined with the transfer matrix method, presented in Sec. VI C, the Drude–Lorentz model has
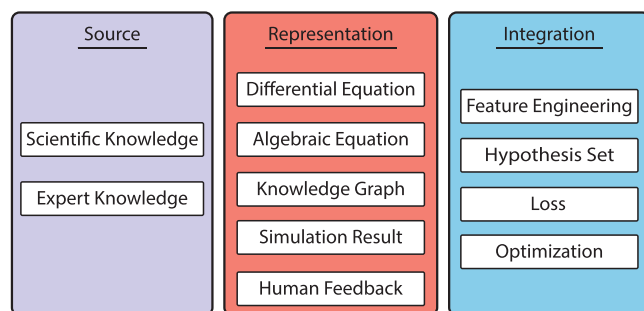
FIG. 3. Schematic showing the taxonomy of PHIL within the realm of AEM. The three main categories of PHIL to be specified consist of *source*, *representation*, and *integration*. The particular PHIL approach can then be taxonomized through a choice (or choices) from each category shown.

Figure contents:
- Source: Scientific Knowledge; Expert Knowledge
- Representation: Differential Equation; Algebraic Equation; Knowledge Graph; Simulation Result; Human Feedback
- Integration: Feature Engineering; Hypothesis Set; Loss; Optimization

been successfully applied in PHIL to guide DNN training. Additionally, the connection between the AEM geometry and material properties to its electromagnetic response can be represented as a knowledge graph. Using a knowledge graph to depict AEM geometry and electromagnetic responses can harness the power of a graph neural network to achieve a physics-informed graph neural network.[49,50] Typically, simulation results are used as ground truth data in training conventional black-box models. However, a model can be trained on a large quantity of low-fidelity data first and then later on high-fidelity data, which is accurate, noise-free, and/or explicitly contains physics. The high-fidelity data can be used to provide more physics guidance to the model or for additional optimization.[33] An AEM expert can also provide intuitive or hidden physics insights[51] akin to additional simulations in this context.

D. PHIL integration

For the remainder of this review, we will concentrate on delineating each integration method along with corresponding examples. Emphasizing integration does not diminish the importance of PHIL sources and representations when tackling a new AEM problem. We presume that readers of this review possess a solid understanding of AEM physics, which mitigates the need for an elaborate discussion on sources and representations, as these aspects are physics-intensive.

As we shift our focus to integration approaches, it is appropriate to clarify what we mean by "explicit." Given the lack of a concrete measure or metric for quantifying the extent to which a PHIL approach integrates physical knowledge, we will use the term "explicit" in the following section to abstractly denote the degree of physical integration. An intuitive concept is illustrated in Ref. 22: The more physics one understands, the less data one requires. It is reasonable to infer that when a PHIL approach integrates physical knowledge more explicitly, the model is more informed by physics and thus requires less data.

Having provided a general overview of the PHIL taxonomy, we now present the major types of PHIL approaches applied to AEM problems in Table I. Over the past five years, *physics-informed learning* has gained considerable attention and is known by various names, such as *physics-guided learning*, *physics-augmented learning*, and *physics-constrained learning*. We chose the term *physics-informed learning* as it is the most widely recognized within the scientific community, aligning with the terminology used in prior reviews such as Ref. 22. We opted not to specify the name as *physics-informed deep learning*, as the scope of PHIL spans the entire spectrum, from purely physics

model-based learning to fully data-driven deep learning. With centuries of expertise in model-based learning, the AEM community has gained substantial insights into deep learning over the past decade. However, the community has largely leaped from one end of the PHIL spectrum to the other. Through this review, we aim to help readers identify a midpoint on the spectrum that achieves a better balance between physics and data. Table I serves to familiarize readers with the most commonly available PHIL techniques. In the subsequent sections, we will explore the detailed variants of these different PHIL approaches.

We organize three integration approaches defined by taxonomy with a trend of increasing explicitness, guiding readers through the progression of PHIL approaches and examples that grow in physical knowledge and become increasingly data-efficient.

IV. PHYSICS INTEGRATION APPROACHES AND EXAMPLES

The successful integration of physics into PHIL models demands extensive balanced domain expertise, meaning an adequate understanding of both the physics underpinning AEMs and the complexities of machine learning algorithms. This dual necessity presents a significant implementation challenge, requiring synthesizing skills and knowledge traditionally segmented into separate research disciplines. Moreover, the inherent diversity within AEM[41,52–54] dictates that a one-size-fits-all approach to PHIL is infeasible. Various materials and geometry within AEM necessitate distinct adaptations across the three dimensions of the PHIL taxonomy.

Given the pivotal role of physics integration in PHIL methodologies, it is imperative to focus on the dimension of *integration* to explore the intricacies of PHIL thoroughly. The span of physics integration is categorized into four principal strategies (see Fig. 3) outlined by taxonomy: feature engineering, hypothesis set, loss, and optimization. These approaches highlight distinct methods for addressing the complexities of PHIL in AEMs. The following subsections will detail these strategies, with specific examples from Table II. The following discussions aim to showcase how these strategies contribute to the expansion of PHIL in AEM by effectively combining the precision of physics with the adaptability of deep learning.

A. Feature engineering

Feature engineering focuses on selecting and processing PHIL model inputs, often termed *features*, based on the physics knowledge.

TABLE I. Major types of physics-informed learning. SK—scientific knowledge, EK—expert knowledge, DE—differential equation, AE—algebraic equation, KG—knowledge graph, SR—simulation results, HF—human feedback, FE—feature engineering, HS—hypothesis set.

PHIL approach	Source		Representation					Integration		
	SK	EK	DE	AE	KG	SR	HF	FE	HS	Loss
Physics based	✓		✓						✓	
PINN	✓		✓							✓
Neural operator	✓						✓		✓	
Graph neural network	✓				✓				✓	
Lorentz neural network	✓			✓					✓	
Neural network		✓				✓		✓		

TABLE II. List of PHIL approaches in AEM problems based on the taxonomy: *source*, *representation*, and *integration*, as described in Sec. III A. PHIL works are organized base on their *integration* approach, and Feature means *feature integration*, Hypo is *hypothesis set*. We use the abbreviation Hypo for hypothesis in the last two rows where these works used a combination of integration methods.

Reference	AEM problem	Physical knowledge	Source	Representation	Integration
Nadell et al.[9]	Metasurface	Maxwell's equations	Expert	Algebraic equation	Feature
Song et al.[55]	Nanophotonics	Maxwell's equations	Scientific	Feedback	Feature
Paseri et al.[56]	Metasurface	Method of moments	Scientific	Algebraic equation	Feature
Pearson et al.[57]	Metasurface	Method of moments	Scientific	Algebraic equation	Feature
Noureen et al.[58]	Metasurface	Materials information	Expert	Feedback	Feature
Sol et al.[59]	Scattering	Coupled mode theory	Scientific	Differential equation	Hypothesis
Blanchard-Dionne et al.[60]	Nanophotonics	Lorentz model	Scientific	Algebraic equation	Hypothesis
Khatib et al.[34]	Metasurface	Lorentz model	Scientific	Algebraic equation	Hypothesis
Wiecha et al.[61]	Nanophotonics	Dipole approximation	Scientific	Algebraic equation	Hypothesis
Majorel et al.[62]	Metasurface	Dressed polarizability	Scientific	Algebraic equation	Hypothesis
Li et al.[63]	Metasurface	Dipole approximation	Scientific	Algebraic equation	Hypothesis
Liang et al.[64]	Nanophotonics	Coupled mode theory	Scientific	Differential equation	Hypothesis
Hu et al.[65]	CEMS	Maxwell's equations	Scientific	Differential equation	Hypothesis
Tang et al.[51]	AEMs	Time-series resonance	Expert	Feedback	Hypothesis
Khoram et al.[49]	Metasurface	Maxwell's equations	Scientific	Knowledge graph	Hypothesis
Kuhn et al.[50]	CEMS	Maxwell's equations	Scientific	Knowledge graph	Hypothesis
Augenstein et al.[66]	Nanophotonics	Maxwell's equations	Scientific	Differential equation	Hypothesis
Zhang et al.[67]	Metasurface	Coupled mode theory	Scientific	Differential equation	Hypothesis
Xu et al.[68]	Metasurface	Coupled mode theory	Scientific	Differential equation	Hypothesis
Jiang et al.[33]	Metasurface	Maxwell's equations	Scientific	Simulation results	Loss
Chen et al.[8]	Metasurface	Maxwell's equations	Scientific	Differential equation	Loss
Fang et al.[69]	Metamaterials	Maxwell's equations	Scientific	Differential equation	Loss
Lu et al.[70]	CEMS	Maxwell's equations	Scientific	Differential equation	Loss
Chen et al.[46]	Scattering	Maxwell's equations	Scientific	Differential equation	Loss
Lim et al.[71]	CEMS	Maxwell's equations	Scientific	Differential equation	Loss
Dove et al.[72]	CEMS	Time-reversal consistency	Scientific	Differential equation	Loss
Gigli et al.[73]	Nonlinear Optics	Maxwell's equations	Scientific	Differential equation	Loss
Hu et al.[74]	Scattering	Maxwell's equations	Scientific	Differential equation	Loss
Medvedev et al.[75]	Nanophotonics	Maxwell's equations	Scientific	Differential equation	Loss
Scheinker et al.[76]	CEMS	Maxwell's equations	Scientific	Differential equation	Loss
Sarkar et al.[77]	Metamaterial	Maxwell's equations	Scientific	Algebraic equation	Loss
Zhelyeznyakov et al.[48]	Metasurface	Maxwell's equations	Scientific	Differential equation	Loss
Park et al.[78]	Nanophotonics	Adjoint method	Scientific	Differential equation	Loss
Jeong et al.[79]	Multiphysics	Maxwell's equations	Scientific	Differential equation	Loss
Peng et al.[80]	CEMS	Green's function	Scientific	Differential equation	Loss
Pan et al.[81]	Waveguide	Maxwell's equations	Scientific	Algebraic equation	Hypo & Loss
Mao et al.[82]	CEMS	Maxwell's equations	Scientific	Differential equation	Hypo & Loss

These features often include material properties, geometrical correlations, and electromagnetic field characteristics. By incorporating these physical features into PHIL models, this integration method enhances both the accuracy and efficiency of the models.

For example, conventional DNNs may struggle to fully capture complex physics interactions using only geometrical AEM features, which can lead to less accurate predictions. By incorporating features that align with electromagnetic theory, such as permittivity, permeability, and specific geometrical characteristics, the models are better equipped to reflect electromagnetic interactions within materials.[1,3,9,34,58] This extensive integration deepens the models understanding of exotic AEM behaviors, enabling it to capture more comprehensive responses of AEM physics, thus surpassing the accuracy and efficiency of conventional DNNs.

Consider, for instance, the design of a metasurface intended to exhibit target farfield scattering properties. Features such as the

incident angle of incoming fields and the materials' composition play a substantial role in this context. A model employing feature engineering integration, leveraging the additional features mentioned above, is expected to predict the metasurface's response more accurately than a model relying solely on geometrical data. Integrating these physics-based features into PHIL models results in accurate and insightful predictions, thereby steering researchers toward more effective AEM designs, as demonstrated in the next subsection.

B. Feature engineering example

In our first example of feature engineering, one study utilized additional geometrical ratios as physically informative inputs.[9] This study exemplifies feature engineering in PHIL for AEM by focusing on the design of all-dielectric metasurfaces (ADMs) characterized by a supercell structure with four silicon cylindrical resonators through feature-engineered PHIL. As shown in Fig. 4, the network processes a vectorized input geometry representing the ADM's geometrical parameters through a sequence of fully connected, 1D transposed convolutional, and 1D convolutional layers. The model maps geometrical parameters inputs to frequency-dependent transmittance outputs, achieving a MSE of 1.16e−3.

This study's key PHIL innovation lies in its use of additional physically informative features. The authors note that an 8-dimensional geometrical parameter set would sufficiently describe the ADM geometry; however, the additional geometrical terms consisting of the ratio of the cylindrical resonator's radius to height were included. The input vector used in the study consists of the ADM geometry and is denoted as $x = \{h_1, h_2, h_3, h_4, r_1, r_2, r_3, r_4, \frac{r_1}{h_1}, \frac{r_2}{h_2}, \ldots\}$. This integration of extra physically informative features significantly enhances the network's performance. This radius-to-height driven improvement resonates with findings where it was found that the radius and height of a perfect absorbing cylindrical resonator are interrelated through relative permittivity ε_r (see the Appendix) and resonant frequency f_0, given by[83]

$$h = \frac{c}{2f_0} \varepsilon_r^{-\frac{1}{2}}, \tag{1}$$

$$r = \frac{3.83c}{2\pi f_0} (\varepsilon_r - 1)^{-\frac{1}{2}}. \tag{2}$$

The authors hypothesized that these ratios are crucial for calculating the spectrum. Their emphasis on the radius-to-height ratio—a critical factor in determining the electromagnetic modes within the resonator—highlights the effectiveness of integrating physically informative features into PHIL models. This example illustrates the potential benefits of feature engineering.

Prior studies have shown that including features that possess relevant physical information, beyond just the basic inputs, can significantly improve the efficiency of DNNs. This results in better accuracy, faster convergence times, and better generalizability to new data. However, blindly adding more features as inputs does not necessarily guarantee an improved DNN. Successful feature engineering relies on the DNN learning the underlying physics encoded within these features. However, in some cases, DNNs might develop a negative bias toward these engineered features, ultimately harming its ability to generalize and learn accurately. Additionally, AEMs often involve physically informative features that lack analytical constraints or closed-form solutions, making them challenging to integrate effectively. The following sections explore methods for PHIL integration that overcome these limitations, leading to more effective and generalizable models.

C. Hypothesis set

Integration of a *hypothesis set* as a PHIL approach emphasizes the design of the DNN architecture to reflect the physical principles underlying AEMs, such as Maxwell's equations. This method goes beyond the more typical method of feature engineering, where physical knowledge is implicitly included through the data used to train the DNN. Instead, hypothesis set PHIL tailors the model framework and architecture to parallel the dynamics and properties of electromagnetic phenomena, thereby explicitly confining the model to operate within a physically meaningful parameter space.

Complex electromagnetic interactions often complicate the behavior of seemingly simple AEM geometries, rendering immediate analysis challenging. Similarly, nonlinear activation functions complicate the inherently linear transformations of layers in DNNs, enabling them to effectively model the complex electromagnetic AEM response. If the architecture of a PHIL model can further mimic electromagnetic interactions in AEMs, it can play a crucial role in enhancing the accuracy and relevance of predictions. Drawing inspiration from the success of convolutional neural networks (CNNs) in image processing,[84] transformers in natural language processing,[85] and recurrent neural networks (RNNs) in sequential data,[86] the value of a tailored PHIL framework becomes evident. One can begin by identifying the type of AEM data to be studied and applying the appropriate network architecture. More advanced approaches could involve an architecture explicitly designed for the AEM problem to boost model performance.

Examples of hypothesis set integration in AEM include CNNs structured to handle spatial hierarchies in material structures[61] and neural operators that can accurately learn functional mappings from permittivity to field.[66,82] Another instance is using graph neural networks (GNNs) to represent and analyze interrelations within complex
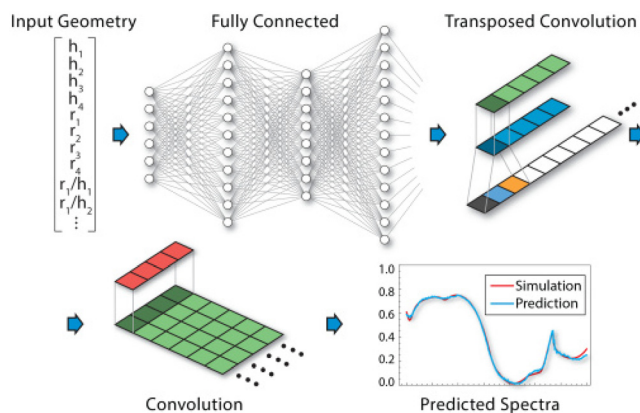
FIG. 4. Illustration of the neural network architecture. Fully connected layers are fed a set of geometric inputs. Data are then smoothed and upsampled in a learnable manner via transpose convolution layers (top row). The final layer (bottom row) is a convolutional layer, which produces a predicted spectra, shown as the blue curve, compared to the ground truth (red curve). Reproduced with permission from Nadell *et al.*, Opt. Express **27**, 27523 (2019). Copyright 2019 Optical Society of America.

materials at the subwavelength level, mirroring physical knowledge in AEMs.[49,50] We will discuss some of these concepts in detail in the following examples.

D. Hypothesis set example

Past studies have used a 3D CNN with a symmetric "U-net" structure, training it on a discretized 3D mesh of nanostructure geometry to predict complex electric fields.[4] This work applied the discrete dipole approximation (DDA) for simulating nanostructures, where each mesh point represents a discrete dipole. As shown in Fig. 5, the 3D CNN's discretization mirrors the 3D mesh in DDA, capturing the spatial information of the dipoles. This approach enables the network to quickly determine internal fields of three-dimensional nanostructures, offering a more efficient alternative to CEMS. Additionally, the network can extract secondary physical attributes, like near/far-field data, from its primary DDA solution, allowing for diverse

electromagnetic responses without separate predictive models. This research demonstrates the ability of a 3D CNN to capture spatial physics priors and augment PHIL outputs into various electromagnetic attributes. While this example marks a significant advancement in the use of 3D CNNs in AEM, there is room for improvement.

We note that the DDA method solves a scattering problem consisting of an array of dipoles ($j = 1, \ldots, N$), each with a polarizability of α_j, located at $\mathbf{r}_j$.[43] The polarization for each dipole is $P_j = \alpha_j \mathbf{E}_j$, where $\mathbf{E}_j$ is the electric field at $\mathbf{r}_j$ induced by incident field $\mathbf{E}_{inc,j}$ and the $N - 1$ dipoles in the array

$$\mathbf{E}_j = \mathbf{E}_{inc,j} - \sum_{j \neq k} \mathbf{A}_{jk} \mathbf{P}_k, \qquad (3)$$

where $-\mathbf{A}_{jk}\mathbf{P}_k$ is the electric field at $\mathbf{r}_j$ from dipole $\mathbf{P}_k$ at $\mathbf{r}_k$. $\mathbf{A}_{jk}$ can be derived from the free-space dyadic Green's function.

Although a 3D CNN may indeed well approximate localized interactions between dipoles, there is no guarantee that it will
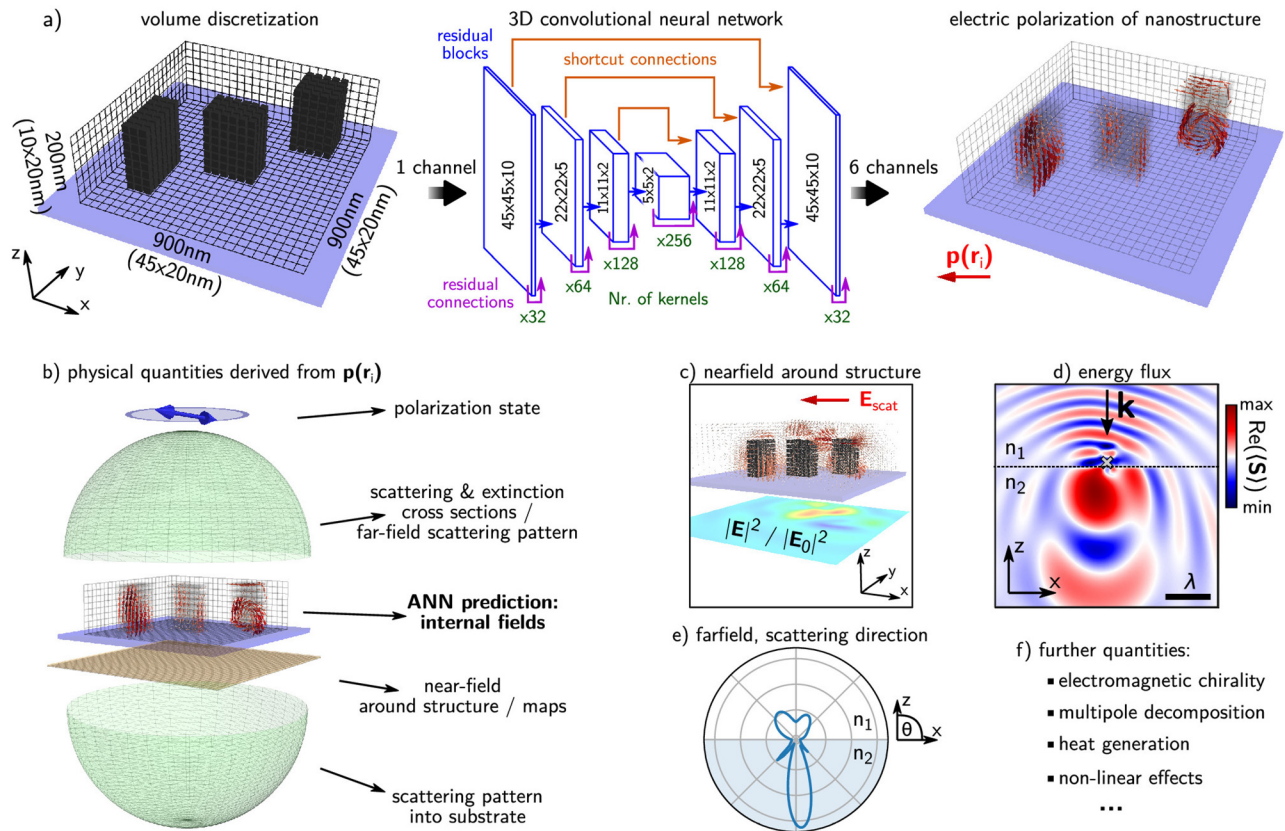

FIG. 5. (a) Sketch of the neural network model used for the silicon nanostructure model. The volume discretization of the three-dimensional geometry is shown in the top left, and consists of $45 \times 45 \times 10$ meshpoints with 20 nm step, and fixed height of 200 nm. These voxels are then fed into the neural network. The three-dimensional convolutional network follows an encoder–decoder architecture and is organized in a sequence of residual blocks. The principal layout of these blocks, the number of kernels, as well as the layer dimensions of the silicon predictor are shown in the sketch of the network (top middle). The six output channels of the network contain the real and imaginary parts of the x, y, and z components of the complex electric field inside the nanostructure. Following the calculation of the self-consistent electric polarization inside the structure, the latter can be interpreted as dipole moments $\mathbf{p}_{\mathbf{r}_i}$ of the single mesh cells at $\mathbf{r}_i$. (b) Various physical quantities in the near field and far field, as illustrated, can be derived from $\mathbf{p}_{\mathbf{r}_i}$. This includes electric or magnetic near-field (c), Poynting vector (d), or far-field scattering patterns (e), among many others (f). The model includes a dielectric substrate ($n_{subst} = 1.45$). The structure is illuminated by a linearly polarized plane wave from the top with $\lambda_0 = 700$ nm. Reproduced with permission from Wiecha *et al.*, Nano Lett. **20**, 329 (2019). Copyright 2019 American Chemical Society.

accurately learn the dyadic Green's function. This uncertainty underscores the importance of more explicitly integrating physical knowledge into the deep learning workflow, which leads us to the next example.

Following observations from the previous example, we turn to an approach that directly addresses this gap: the Lorentz neural network (LNN). In contrast to a 3D CNN, the LNN includes a Lorentz layer that outputs the causal complex frequency-dependent permittivity and permeability, integrating the condition of causality through parameters from the Lorentz oscillator model. This integration ensures that the network inherently aligns with causality in physics, illustrating a method by which an explicit physics model can lead to more accurate and physically reliable PHIL models in AEM.

In terms of its architectural design, which is delineated in Fig. 6, the LNN is fashioned as a feedforward neural network consisting of 100-250-250-100 neurons in each layer. The physics integration of the LNN is achieved by tailoring the final layer to represent the parameters of Lorentz oscillator models, as detailed in Eqs. (20) and (21). This final layer accounts for four frequency-dependent complex oscillators each, both for the permittivity and permeability. Including two additional terms—the ε_∞ and μ_∞ terms described in the Appendix—the Lorentz layer comprises 26 parameters in total. The Lorentz parameters predicted by the network are used to construct the Lorentz resonator, which is then applied in the transfer matrix method to calculate complex reflection and transmission coefficients. The loss function is defined to incorporate additive MSE based on the ground truth, defined as

$$\mathscr{L}_{LNN} = \alpha \left[\overline{(\Delta R)^2} + \overline{(\Delta T)^2} + \overline{(\Delta \tilde{r}^*)^2} \right], \qquad (4)$$

where $\Delta R \equiv |\tilde{r}|^2 - |\tilde{r}_p|^2$, $\Delta T \equiv |\tilde{t}|^2 - |\tilde{t}_p|^2$, and $\Delta \tilde{r}^* \equiv \Im(\tilde{r}) - \Im(\tilde{r}_p)$, p subscripts denote predicted quantities, and $\Im(\cdot)$ is the imaginary part. An imaginary reflection coefficient term is also included to correctly assign modes, as electric and magnetic modes exhibit opposite signs in the reflected phase. The LNN achieved an MSE of
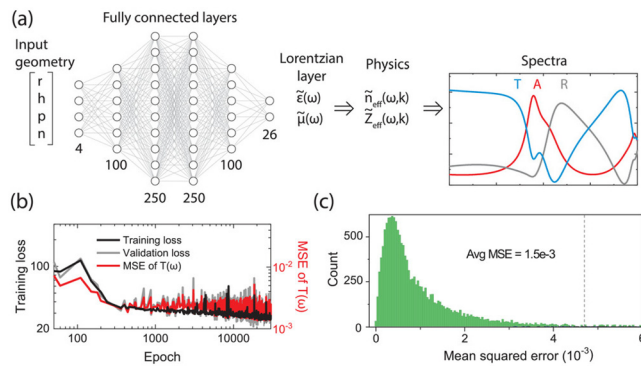

FIG. 6. (a) Illustration of the Lorentz neural network (LNN). (b) Training loss (black), and validation loss (gray) for the LNN as a function of epoch. The MSE of the predicted transmission spectra (red curve) is shown on the right axis. Sharp spikes in the loss at higher epochs are from frequent warm restarts. (c) Histogram of the MSE for the predicted transmission spectra in the hidden test set, where 95% of the data have an MSE of $< 4.6 \times 10^{-3}$, as indicated by the gray dashed line. Reproduced with permission from Khatib et al., Adv. Opt. Mater. **10**, 2200097 (2022). Copyright 2022 Wiley.

1.5×10^{-3} for transmission spectra on the test set. The LNN exemplifies the successful learning of physical knowledge by adhering to the causal Lorentz oscillator model. It has been shown that the LNN requires a much smaller model size (two orders of magnitude) to achieve similar accuracy compared to a conventional DNN, and it outperforms a conventional DNN with the same order of model size. The benefits of the LNN extend beyond improved accuracy. The physically meaningful outputs—Lorentz oscillators—allow for further physical analysis, such as the determination of electromagnetic modes, spatial dispersion, and oscillator characteristics.

We note that the example from Ref. 34 is not the first to apply Lorentz oscillators to neural networks. The authors in Ref. 60 demonstrated that neural networks could be trained in optics by integrating harmonic oscillators. However, we selected Ref. 34 as the example for LNN because it studies metasurfaces in the form of dielectric resonators, which more closely resemble the complexity of AEMs. Nonetheless,[60] showcased a 20-fold improvement in data predictions outside the training space through its study on two parameters of periodic gratings, the slit width a and grating periodicity d.

The LNN underscores the benefit of more explicitly integrating physical knowledge by demonstrating enhanced generalization and a reduced need for data. However, the LNN is not without its shortcomings. One significant issue arises from the Lorentz layer, which still mandates the use of the transfer matrix method to transition from Lorentz oscillators to spectral outputs, which is required to compute the loss. Such complications can lead to vanishing or exploding gradients during backpropagation. Although small batch sizes and warm restarts can mitigate the problem, training the model remains difficult. To bypass vanishing or exploding gradient issues, we can seek PHIL integration approaches that do not require additional transformations beyond the DNN. Building upon the insights from the LNN, our exploration progresses to PHIL integration approaches that directly customize DNN architectures for physics problems.

As highlighted in Refs. 49 and 50, the deep learning architectures we have discussed, such as feedforward and convolutional networks, share a common limitation: a fixed input size. These models are trained on a specific input size that represents AEM geometry, fields, or material properties and are unable to generalize to different input dimensions, with their performance collapsing outside the trained input size. Both Refs. 49 and 50 address this issue using GNNs. In Ref. 49, GNNs were used to learn the coupling between scatterers (individual metasurface unit cells), enabling the model to extrapolate near-field interactions across hundreds of wavelengths.

Similarly, Kuhn et al.[50] addressed the same limitation by doing FDTD simulations through GNNs. The proposed GNN is explicitly designed to predict the propagation of electromagnetic fields in both space and time, adhering to the formalism of FDTD. While we encourage readers to explore the detailed GNN architectures presented in the original work,[50] our focus here is on graph representation, which serves as the foundation for physics-informed learning in GNNs. Figure 7 illustrates the graph representation for both the Yee-grid and unstructured mesh. By learning the field interactions within a small domain, GNNs can extrapolate to much larger domains and accommodate a wide variety of scatterer geometries, showcasing their scalability and adaptability in AEM applications.

Beyond GNNs' ability to address the fixed input challenge, their parallelism with FDTD highlights their potential for application in
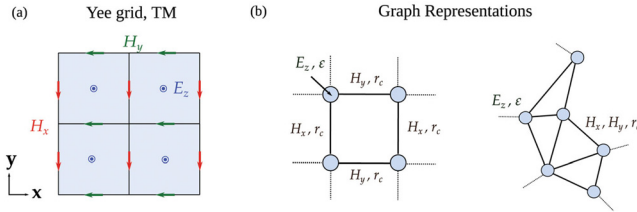
FIG. 7. (a) Yee-grid in TM polarization. (b) Graph representations as a grid and an unstructured mesh. The node features encode the electric field E_z and permittivity ε, and the edge features hold $H_{x,y}$, and local Cartesian vector r_c. Reproduced with permission from Kuhn *et al.*, APL Photonics **8**, 036109 (2023). Copyright 2023 Author(s), licensed under a Creative Commons Attribution (CC BY) license.

FDTD and, more broadly, in CEMS. GNNs are inherently well suited for representing grids and meshes, as graph representations can be easily constructed for such structures. However, not all problems require or can benefit from a graph representation. GNNs can introduce significant computational and memory bottlenecks due to the rapid growth of adjacency matrices and node features in larger systems. Nevertheless, GNNs prove particularly useful for numerical simulations, as they can learn to propagate grid or mesh representations of electromagnetic field data effectively. Connecting back to the hypothesis set $\mathcal{H}$, using GNNs confines $\mathcal{H}$ to the graph representation of grids or meshes, creating an explicitly physics-informed space. Furthermore, as demonstrated in Ref. 50, the successful application of GNNs to mesh data suggests broader applicability in finite element methods (FEM), extending their utility to various CEMS approaches.

In our discussion thus far, we have examined a range of architectures derived from classic deep neural network architectures, such as feedforward, convolutional, and graph neural networks. As we continue exploring hypothesis-set-based PHIL approaches, we now turn our attention to an architecture that departs from conventional DNNs. Introduced in Ref. 27, the Fourier neural operator (FNO) distinguishes itself by its unique approach to embedding Fourier transformation into the neural operator.[87] The subsequent example will present a baseline application of the FNO on an electromagnetic problem designed to effectively solve PDEs that are governed by Maxwell's equations.

Some recent works have demonstrated a distinct approach, under the scope of the hypothesis set, by infusing physics knowledge implicitly.[66] The authors applied a FNO, an operator-based model that utilizes a Fourier transform to efficiently handle the complexities associated with electromagnetic scattering problems. The FNO can perform global convolutions that are more suited for electromagnetic problems compared to traditional local convolutions used in other architectures such as CNNs.

Initially, the input $v(x)$ is elevated to a higher dimensional representation $y(x)$ through the transformation

$$y(x) = C_{in}(v(x)) \in \mathbb{R}^w, \tag{5}$$

where x represents a point on the computational mesh, $C_{in} : \mathbb{R}^{d_{in}} \to \mathbb{R}^w$ is a transformation dictated by the input dimension d_{in} and the output dimension w. The dimension of the input is typically small, while the output dimension, or "width" of the FNO, represents a hyperparameter that determines the number of kernels in each Fourier layer. After the input dimension is expanded using Eq. (5), the data proceed through a sequence of n Fourier blocks. Each Fourier block consists of a transformation in Fourier space $K_m(y) = \mathscr{F}^{-1}(R_m \cdot \mathscr{F}(y))$, followed by a linear update and a batch normalization layer. This sequence concludes with an activation function, typically ReLU (e.g., GELU is used in Ref. 66) expressed as

$$u(y) = \sigma_{BN}(\mathscr{F}^{-1}(R_m \cdot \mathscr{F}(y)) + Wy + b), \tag{6}$$

where $\mathscr{F}$ is the Fourier transform. R_m, W, and b are the trainable parameters, with R_m being the kernel matrix of the convolution in Fourier space, W being the weight, and b being the bias locally on y. After a total of n Fourier blocks, the data are then converted to the desired output dimension d_{out} through another transformation by linear layer

$$z(x) = C_{out}(y(x)) \in \mathbb{R}^{d_{out}}. \tag{7}$$

A detailed illustration of the FNO architecture is shown in Fig. 8. As suggested by the authors,[27,66] by controlling the number of Fourier terms in each Fourier layer, the FNO serves as a low pass filter that smooths the output, which is particularly useful for PDEs such as Maxwell equations with wave-like solutions. The authors provide empirical evidence of the FNO's superior data efficiency and prediction accuracy compared to traditional U-Net architectures. They demonstrate that the FNO requires fewer training samples to achieve the same accuracy compared to U-net, making it more desirable for complex electromagnetics surrogate models. The paper also showcases detailed examples of how the FNO has been effectively applied in the inverse design of nanophotonic devices, including free-form, three-dimensional electromagnetic scatterers, through the aid of variational autoencoders.[66]
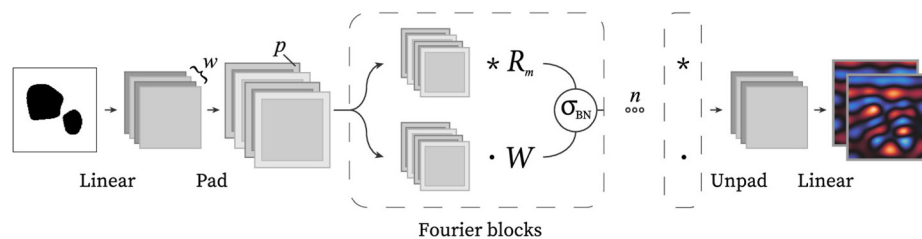

FIG. 8. Illustration of the FNO architecture. The input is expanded via a linear layer to w channels and then zero-padded by p pixels. Data are then passed through a series of n "Fourier blocks," each containing a linear layer and a convolution in Fourier space using the learned kernel R_m, a batch normalization layer as well as a GELU nonlinearity. Finally, the padding is removed, and the channel dimensions are reduced via a linear layer to the desired output dimensions. Reproduced with permission from Augenstein *et al.*, ACS Photonics **10**, 1547 (2023). Copyright 2023 American Chemical Society.

TABLE III. Table of general numerical approaches to solve AEM problems. Here, general PDE includes conventional computational electromagnetic simulations, and PDE learning and operator learning are two of the most common DL approaches.

	Type	Formulation	Data-free	Mesh-free	Retraining-free
Non-DL	General PDE	$s_t + \mathcal{N}[s; g] = 0$	Yes	No	No
DL	PDE Learning	$\hat{f}_t + \mathcal{N}[\hat{f}; g] = 0$	Yes[a]	Yes	No
	Operator Learning	$\hat{s} = \hat{f}(g)$	No	Yes[b]	Yes

[a]PDE learning methods can potentially be data-free, though not all methods are.
[b]Operator Learning methods can be mesh-free, but not all operator learning methods are mesh-free.

Now that we have gone through various examples under different PHIL integration approaches. It is important to note that most, if not all, DNNs can be adapted to a PHIL model through one or more physics integration methods. We can observe that some approaches are inherently *physics-informed* such as PINNs. Meanwhile others, such as feature engineering, are predominantly data-driven and operate as black boxes—however, integrating more explicit physics at various stages can transform these into PHIL.

In the preceding sections, DNNs were characterized as surrogate models $\hat{f}$, approximating the true forward function f, commonly expressed as $y = \hat{f}(x)$, where y represents the output and x represents the input features. We described the parameterized nonlinear PDE problem in a general form[21] and tailored it for AEM problems

$$s_t + \mathcal{N}[s; g] = 0, \quad \mathbf{x} \in \Omega, \quad (8)$$

where $s = s(\mathbf{x})$ is the solution to the PDE, s_t is the time derivative of s, $\mathcal{N}$ represents a nonlinear operator, $g = g(\mathbf{x})$ denotes the parameters of the PDE, $\mathbf{x} = x_1, \ldots, x_d$, and Ω is a subset of $\mathbb{R}^d$. We can regard Eq. (8) as a simplified representation of the PDE problems that CEMS addresses. When comparing CEMS with popular deep learning approaches in AEMs, each can be similarly formulated by substituting $\hat{f}$ either with solution s (PDE learning) or directly training $\hat{f}$ as the functional operator from g to s (Operator learning). Table III summarizes the type, formulation, and characteristics of three major approaches to solving AEM problems, including CEMS.

We first examine the operator learning method which is arguably the most popular approach in the field of AEM. Operator learning using DNNs to map from space $g(\mathbf{x})$ to $s(\mathbf{x})$, effectively learns the operator $\mathcal{M} : g \to s$. One example of operator learning in AEM is: $\mathcal{M} : \varepsilon(\mathbf{x}) \to \mathbf{H}(\mathbf{x})$, where $\varepsilon(\mathbf{x})$ and $\mathbf{H}(\mathbf{x})$ are permittivity and magnetic field at spatial coordinate $\mathbf{x}$.[61,88] However, operator learning isn't restricted to mapping between spaces with the same functional dependence (e.g., $\mathcal{M} : \varepsilon(\mathbf{x}) \to \mathbf{H}(\mathbf{x})$). We can consider inputs and outputs that are functions of $\varepsilon(\mathbf{x})$ and $\mathbf{H}(\mathbf{x})$. For example, we may have vector representations as input,[5,9] scattering responses as output,[6,89] or problems involving broad bandwidth spectra.[7,90] In these cases, DNNs effectively learn an operator mapping from the space $f_{input}(\varepsilon(\mathbf{x}))$ to the space of $f_{output}(\mathbf{H}(\mathbf{x}))$.

While the operator learning approaches traditionally learn physics through training with data generated from ground-truth physics (i.e., CEMS or experiments) K,[5–7,9,88,90] they only learn the statistical distribution within the physics space defined by the data. There is no guarantee that the learned operator accurately approximates the actual physics operator or can generalize outside the data-defined space.

Neural operators,[26,27] including the FNO example,[66] learn the operator $\mathcal{M}$ that is the solution operator for an underlying PDE. In the previous example of FNO[66] and AEM problems, the FNO is learning the solution operator for Maxwell's equations. All operators learning can be written as shown in Table III.

As we conclude our exploration of the hypothesis set examples, it becomes evident that these models predominantly incorporate physics knowledge by the nature of their framework or architecture rather than explicitly through direct regularization based on known physics PDEs or governing equations. To transition into a more explicit approach for embedding AEM physics into DNNs, we will next delve into loss integration approaches that directly leverage PDEs derived from Maxwell's equations or other AEM-specific formulations. This shift marks a critical progression in our discussion, focusing on approaches that explicitly integrate PDEs or physics laws into model regularization.

E. Loss

A notable technique that provides for more explicit physics integration is the aforementioned PINN,[21] as shown in Table III under PDE learning. PINNs use a DNN to approximate the solution s, where the operator $\mathcal{N}$ is known, making PINNs inherently *physics-informed*.[91–93] In AEM, $\mathcal{N}$ takes the form of Maxwell's equations, meaning $\hat{f}$ must satisfy both the loss from data and the loss from the PDEs of Maxwell's equations. Further details of the PINN framework and the loss formulation will be elaborated in examples. We note that PINNs can potentially be data-free and purely trained on the PDE loss.[21]

It is important to note that while PINNs have achieved considerable popularity, they represents just one approach within the broader PHIL framework.[94–96] The architecture of the PINN restricts it to solve only a single PDE problem at a time, and therefore it has limited utility as a rapid surrogate model for various classes of PDEs.[22] To extend PHIL to a broader class of PDEs, we can revisit the operator learning approach with diverse physics representations.

In the loss integration approach, the emphasis is placed on adapting the loss function—the core of a model's learning process—to embed physical laws and principles. This approach enables DNNs to learn physical phenomena guided not only by data but also by explicit physics laws such as PDEs.

The loss function, typically a measure of the difference between the model's predictions and the actual data, is modified to include terms that represent physical constraints or objectives. For example, in AEM, these modifications might involve ensuring that the model's

predictions adhere to Maxwell's equations or other electromagnetic theories. In a most general form, the loss function of a PHIL model with loss-based integration is

$$\mathscr{L}_{PHIL} = \mathscr{L}_{data} + \alpha\mathscr{L}_{physics}, \tag{9}$$

where $\mathscr{L}_{data}$ is the conventional loss computed from the data, and $\mathscr{L}_{physics}$ is the loss based on the known physics knowledge (e.g., PDEs) with a weighting term α. This approach can be particularly beneficial in scenarios where empirical data are scarce or noisy, as physical loss provides an additional layer of guidance to the model, ensuring that its predictions remain physically meaningful.

The most well-known loss integration framework in PHIL is the PINN. The development of PINNs primarily occurred to address general nonlinear parameterized PDE problems and was capable of handling noisy data and simplifying the complexities of solving high-dimensional parameterized PDEs.[21] While they gained significant popularity in mechanics, notably in fluid dynamics governed by Navier–Stokes equations,[97] their utility in other science fields has been shown.[92,93] Recently, PINNs have shown promise in the domain of electromagnetism,[8,48] finding applications in problems described by Maxwell's equations. However, as we will see from the examples, PINNs are inherently limited to solving one PDE at a time and lacking the potential to serve as a fast neural surrogate model for classes of PDEs. In the following subsection, we will start with an example of PINNs in AEM and proceed to provide a more comprehensive overview of available loss integration approaches.

F. Loss example

The first loss integration example applied PINNs to metamaterial and nanophotonics problems, showcasing its ability in solving inverse scattering problems.[8] We will specifically explore how PINNs are used for effective permittivity retrieval, which is defined as "inverse meta-material design" in this example, illustrating their potential in AEM problems. In the PINN example illustrated in Fig. 9, the model solves for the effective permittivity of nanocylinders arranged in a two-dimensional space. This setup is particularly relevant for metamaterial researchers seeking efficient methods to retrieve spatial effective permittivity. Unlike typical PINN applications, this problem, which focuses on solving electric fields and effective permittivity across the entire computational domain, does not utilize the common boundary loss term in Ref. 21. Instead, the total loss term includes the PDE loss term and a field observation loss term.[8] The PINN is regularized with a Helmholtz equation for weakly inhomogeneous two-dimensional media under TM polarization excitation, defined as follows:

$$\nabla^2 E_z(x,y) + \varepsilon_r(x,y)k_0^2 E_z = 0, \tag{10}$$

where $E_z(x,y)$ is the z component of electric field, $\varepsilon_r(x,y)$ is the relative permittivity, and $k_0 = 2\pi/\lambda_0$ is the free space wave number.

By training the PINN model with observed electric field data points, the embedded DNN was optimized to predict the electric field accurately. They also retrieved the effective permittivity term from the Helmholtz equation for weakly inhomogeneous two-dimensional media through data-driven training. The effective permittivity obtained aligns with theoretical expectations (1.25 to 1.35 ± 0.056). The authors conducted a few more simulations to validate the PINNs capability in AEM applications including the consideration of
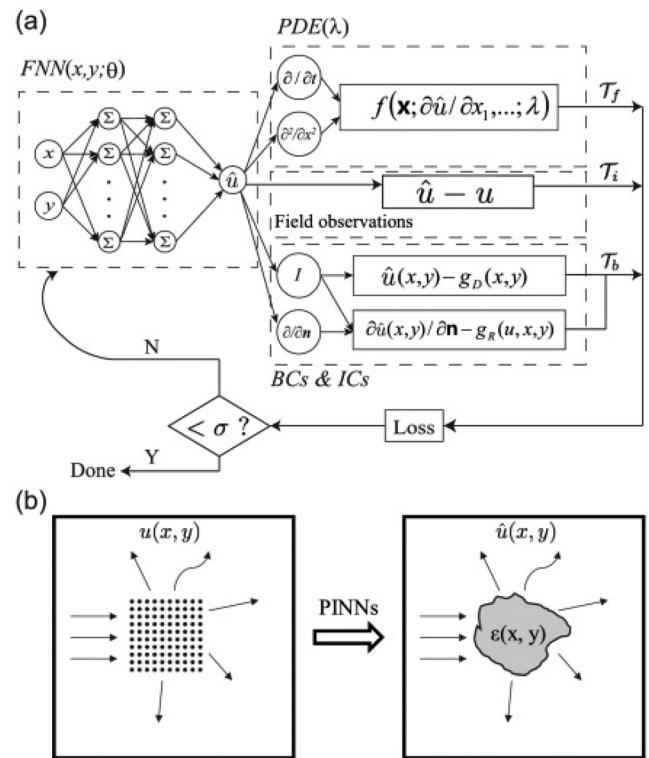

FIG. 9. (a) Schematic of a PINN for solving inverse problem in photonics based on partial differential equations. The left portion shows a neural network which represents a surrogate model $\hat{u}$ of the PDE solution. The right part shows the loss function used to restrict $\hat{u}$ to satisfy the PDE in the domain Ω, boundary conditions (BCs) $\hat{u}(x,y) = g_D(x,y)$ on $\Gamma_D \subset \delta\Omega$, and $\frac{\delta\hat{u}}{\delta n}(x,y) = g_D(x,y)$ on $\Gamma_D \subset \delta\Omega$. The initial conditions (ICs) are treated as a special type of boundary conditions. For the inverse problem, we also have a loss term from $\hat{u} - u$ on residual points. The neural network's weights and biases are optimized to obtain loss smaller than σ. (b) Process shows the process of PINNs to reconstruct the permittivity profile ε from a known scattering field profile dataset. Reproduced with permission from Chen *et al.*, Opt. Express **28**, 11618 (2020). Copyright 2020 Optical Society of America.

boundary conditions. It was concluded that the PINNs are valuable in aiding the design of new-generation metamaterials through inverse scattering and effective property retrieval.[8] The PINN is particularly useful if a problem can be constructed in terms of differentiable physics[21,98] so that there is a precise PDE formulation. Infusing the PDE as the underlying physics prior, PINNs have clear advantages over purely data-driven methods, including better interpretability and improved generalizability.

Beyond applying PINNs to inverse scattering problems,[8,46,74] exploring PINNs as a data-free alternative[48,70,99] to traditional numerical simulation tools is intriguing. Since PINNs were introduced, there has been significant research into using them for differential physics problems, either as data-driven or data-guided/data-free methods. The case of inverse scattering in metamaterials[8] demonstrates how data-driven PINNs solve for parameters of partial differential equations effectively. Recall that the PINN has the framework $\hat{f} + \mathscr{N}[\hat{f};g] = 0$, and therefore, the network is essentially learning the unknown PDE parameters g through data. Alternatively, PINNs can operate entirely

without data or with a small amount of data as a guide, relying chiefly on well-defined PDE loss, including the governing PDEs and initial and boundary conditions, to predict the solution field $\hat{f}(\mathbf{x})$. This data-guided/data-free approach, anchored in explicit physics priors, circumvents the need for data collection and ensures compliance with physical laws. In one study, the application of a data-free PINN in optimizing a large-area metasurface was demonstrated.[48]

A PHIL approach distinct from PINNs is termed WaveY-Net, which we will discuss next.[31] The WaveY-Net is a data-driven approach which is balanced with explicit physics priors, and it constitutes a multitasking PHIL model. We note that multitasking in this context is limited to different nanophotonics materials and geometry.

An exhaustive multiphysics/multiscale model would require much more data and computational resources to train.

WaveY-Net represents a distinct loss integration approach that blends both data-driven methodologies and physics loss. The underlying architecture of WaveY-Net is a U-net structure shown in Fig. 10. The convolutional neural network is designed to predict electromagnetic field distributions across multiple dielectric nanophotonic structures. The core of the physics integration lies in its loss function. The network is trained on magnetic field distributions, utilizing Maxwell's equations to derive electric fields and incorporating Maxwell's equations on electromagnetic fields as the physical loss. Consequently, WaveY-Net efficiently predicts near-fields in high dielectric contrast
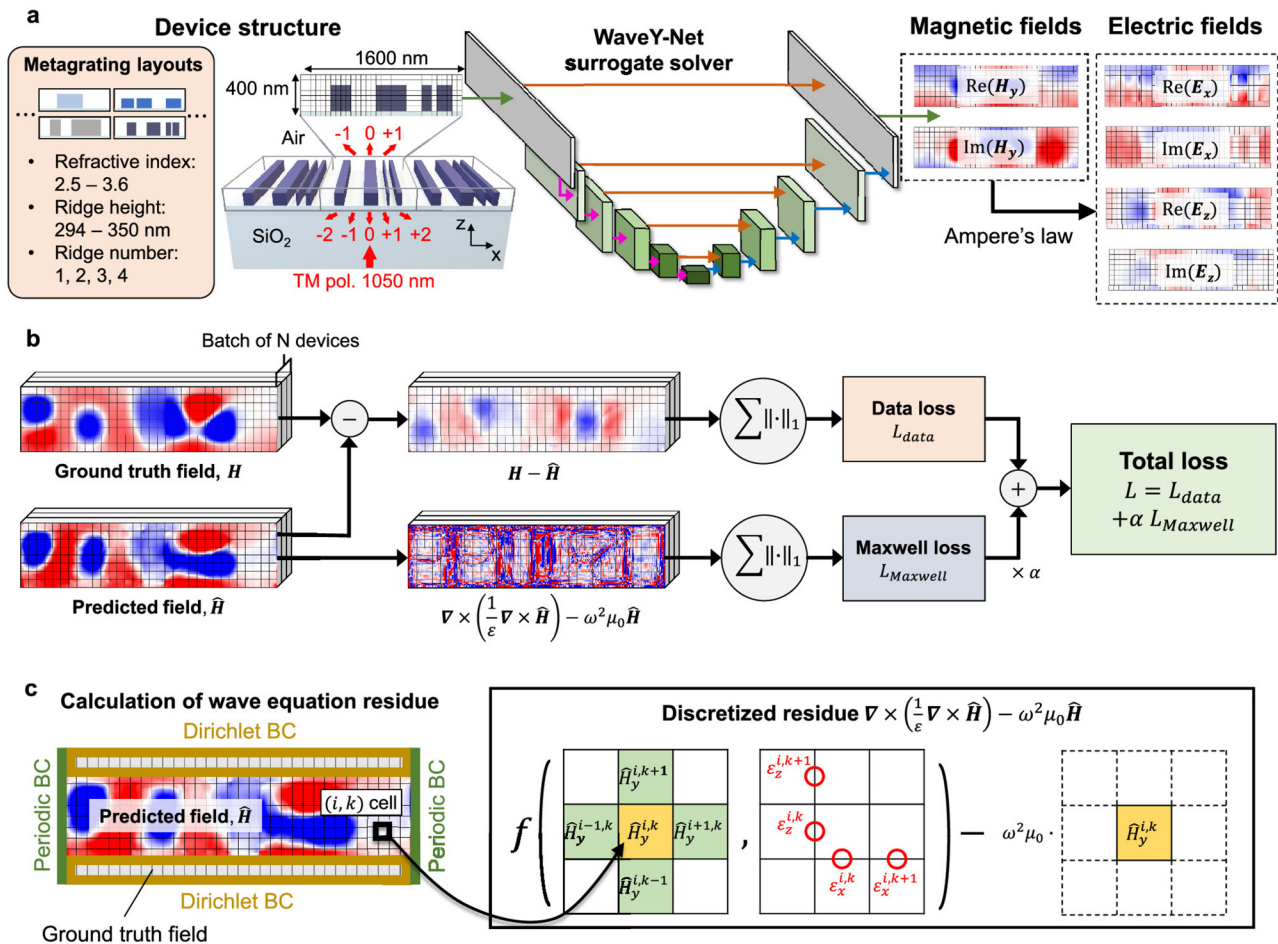

FIG. 10. Overview of the WaveY-Net network architecture and training procedure. (a) UNet architecture is used to learn the electromagnetic behavior of a dielectric-based metagratings (left). The input is an image of a single unit cell of the metagrating (center-right), and the outputs are magnetic field maps (center-right). Electric field maps are calculated from the magnetic field maps using a discrete version of Ampère's law (right). The devices being studied are nanoridge-based freeform metagratings with different grating materials, varying number of nanoridges, and varying ridge heights. Orange arrows: shortcut connections. Pink arrows denote periodic convolution and maxpooling operations. Blue arrows are periodic convolution and upsampling operations. (b) Computational graph of the loss function, which is comprised of two terms: data loss, which quantifies deviations between ground truth and predicted magnetic field maps, and Maxwell loss, which quantifies the deviation of the predicted magnetic field maps from Maxwell's equations. (c) Calculation of Maxwell loss. Periodic boundary conditions along vertical boundaries and Dirichlet boundary conditions along horizontal boundaries enforce a well-posed loss expression. Maxwell loss within the magnetic field maps is calculated using a discretized version of the wave equation residue, which imposes relationships between magnetic field values at a given voxel and those of its nearest neighbors. Reproduced with permission from Chen *et al.*, ACS Photonics **9**, 3110 (2022). Copyright 2022 American Chemical Society.

nanostructure arrays and synergizes with gradient-based algorithms, enhancing the design speed of nanophotonics devices.

While parallels might be drawn between WaveY-Net and the principles of PINNs, closer examination reveals distinct operational differences. Unlike PINNs, which primarily rely on spatial coordinates to produce continuous field distributions, WaveY-Nets operate on the discrete dielectric distribution, yielding discrete field results. While PINNs are typically tailored for a specific dielectric distribution or a single nanophotonics geometry, WaveY-Net's design enables it to cater to a broader class of configurations. This effectively positions WaveY-Net in the realm of "operator learning" with respect to Maxwell's equations. In terms of computational efficiency, WaveY-Net boasts the ability to perform rapid forward simulations and accelerated gradient-based inverse design, in contrast to the more time-intensive processes associated with PINNs. In Table III, we highlight that WaveY-Net takes the form of operator learning, in contrast to the PINN form.

Central to the WaveY-Net approach is the total loss function taking the form of Eq. (9), defined as $L_{total} = L_{data} + \alpha L_{Maxwell}$, where α serves as a weight factor. This α factor modulates the balance between data-driven modeling and physical constraints. The network's adaptability, contingent on the value of α, allows it to bridge the gap between pure data-driven models and those echoing the characteristics of PINNs. In essence, WaveY-Net embodies a harmonized approach, aiming to merge the computational advantages of deep learning with the invaluable insights derived from physics.

The integration of neural network models with physical constraints to solve complex problems has garnered attention in recent physics literature. Another contribution to this domain is the physics-constrained neural network (PCNN) approach tailored to address Maxwell's equations pertaining to electromagnetic fields of intense relativistic charged particle beams.[76] Fig. 11 shows that this work employs a 3D convolutional PCNN that maps time-varying current and charge densities $J(r, t)$ and $\rho(r, t)$ to the corresponding vector and scalar potentials $\mathbf{A}(r, t)$ and $\phi(r, t)$, respectively, where $r = (x, y, z)$ is 3D spatial coordinate and t is the time—see Sec. VI A. From these potentials, electromagnetic fields are subsequently derived in accordance with Maxwell's equations (see Sec. VI A), represented as $\mathbf{B} = \nabla \times \mathbf{A}$ and $\mathbf{E} = -\nabla \phi - \frac{\partial \mathbf{A}}{\partial t}$. The inherent design of PCNN ensures the satisfaction of hard constraints like $\nabla \cdot \mathbf{B} = 0$. Additionally, soft

FIG. 11. Various NN approaches to generate a magnetic field $\hat{\mathbf{B}}$ (an estimate of $\mathbf{B}$) from current density $\mathbf{J}$ are shown. Reproduced with permission from Scheinker et al., APL Mach. Learn. **1**, 026109 (2023). Copyright 2023 Author(s), licensed under a Creative Commons Attribution (CC BY) license.

constraints are implemented to guide $\mathbf{A}$ and ϕ toward satisfying the Lorenz gauge.

The 3D PCNN stands out for several reasons. First, it signifies the melding of the principles behind WaveY-Net and the capabilities of 3D convolutional neural networks. Inclusion of the charge density and potentials in its framework enhances its applicability, making it a versatile tool for a broad spectrum of electromagnetic simulations. While acknowledging the contributions and flexibility of PINNs, this research highlights that 3D PCNN offers a fast surrogate model for a class of electromagnetic problems.

The combination of data and physics loss fundamentally guides both WaveY-Net and 3D PCNN. The central tenet is the utilization of loss functions embedded with physical constraints (Maxwell's equations) to guide the training of PHIL models. Though PINNs also use physical loss in their training, the fundamentally different frameworks indicate that operator learning approaches (WaveY-Net and 3D PCNN) will tackle a different class of problem than those handled by PINNs.

We have covered three integration approaches and their corresponding examples to illustrate how each integration approach functions. It is straightforward to see that each integration approach occurs at a different stage of the deep learning workflow, which makes the parallelization of all three integrations simultaneously compatible. Consequently, we will likely see more hybrid PHIL models that utilize all possible integration approaches. For the final example of this review, we will examine a model that concurrently integrates physics as both a *hypothesis set* and in the *loss function*.

The authors proposed a method termed the specialized neural accelerator-powered domain decomposition methods (SNAP-DDM), a Fourier neural operator-based network designed to solve arbitrary PDE problems and boundary conditions.[82] Although FNO was successfully applied to 2D electromagnetic problems, In contrast to a u-net approach, both of these methods are data-driven.[66] Here, we reiterate the three limitations of neural operators stated in Ref. 82: (1) curse of dimensionality: the dimensionality of PDE problems grows exponentially with the scale of the problem. (2) Network size: it is challenging to continuously scale the network to fit the increasing number of parameters. (3) Data-hungry nature: Neural operators still function as data-driven black-box models, requiring a large amount of data for training.

To mitigate these problems, SNAP-DDM employs several techniques in addition to FNO to enhance its performance. We will discuss these techniques under *hypothesis set* and *loss function* integration. SNAP-DDM naturally falls under classification as *hypothesis set* integration as its foundational network architecture is FNO.[27] Beyond the underlying FNO architecture, SNAP-DDM leverages domain decomposition methods[100,101] demonstrated in Fig. 12. To handle arbitrary global boundary conditions, SNAP-DDM subdivides the global boundary conditions into sub-domains that are feasible for an FNO to solve. A separate FNO manages each divided sub-domain. After achieving the necessary high accuracy in each subdomain, the sub-solutions undergo an iterative and self-consistent algorithm (Schwarz DDM) to form the global solution.

The SNAP-DDM architecture adopts a similar approach to WaveY-Net[47] for its loss integration. By incorporating both data loss and PDE/Physics loss, the SNAP-DDM reduces extensive data requirements and mitigates the data-hungry nature of FNO. Although SNAP-DDM also utilizes a self-modulating module to enhance model
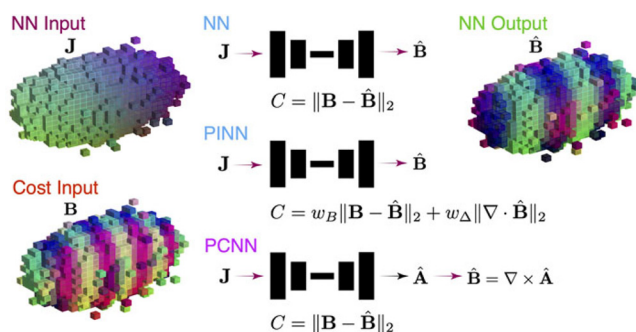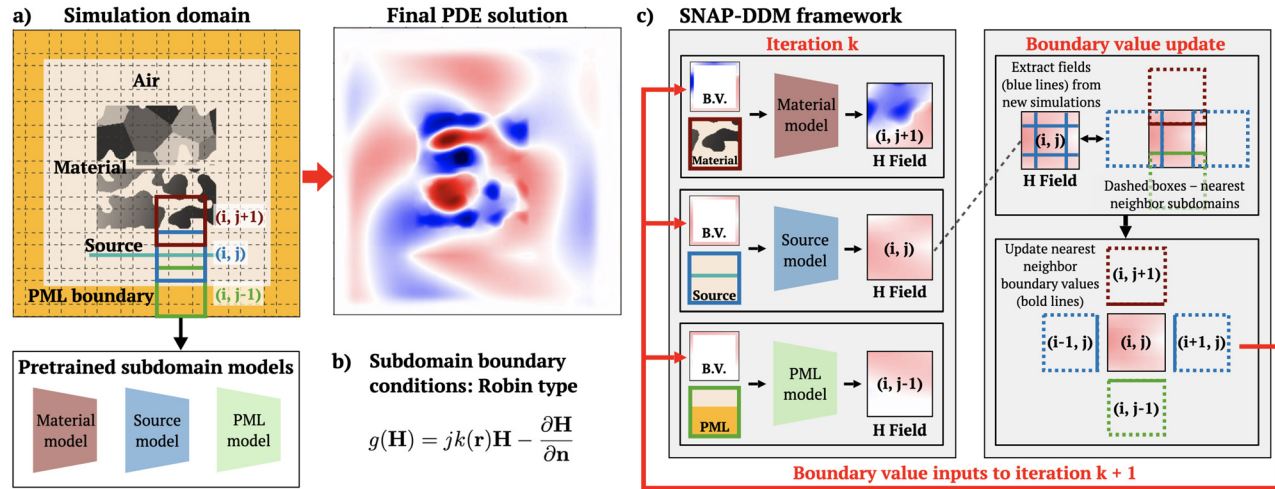
FIG. 12. SNAP-DDM framework with electromagnetics as a demonstration. (a) Shows the global simulation domain and corresponding H-field solution for a 2D electromagnetics problem featuring arbitrary sources, global boundary conditions, and freeform grayscale dielectric structures. The global domain is subdivided into overlapping subdomains parameterized by position (i, j). Three types of specialized neural operator models are trained to solve for three types of subdomain problems. (b) Expression for the Robin type boundary condition used in the specialized neural operator subdomain models. $k(\mathbf{r}) = 2\pi\sqrt{\varepsilon(\mathbf{r})}/\lambda$ is the wave vector in a medium with dielectric constant ε and $\mathbf{n}$ is the outward normal direction. (c) Flow chart of the iterative overlapping Schwarz method. In iteration k, electromagnetic fields in each subdomain are solved using the specialized neural operators, and the resulting fields are used to update the subdomain boundary value inputs for iteration $k + 1$. The "boundary value update" box shows how solved fields in the (i, j) subdomain are used to update the boundary value fields in nearest neighbor subdomains for subsequent iterations. **B.V.**: boundary value. **PML**: perfectly matched layers. Reproduced with permission from Mao et al. arXiv:2405.02351 (2024).

performance, this feature does not fall under the category of PHIL and will not be discussed in this review.

Section Summary: We close this section by emphasizing the necessity for the integration of diverse knowledge in PHIL approaches to maximize the explicit representations of physics in the model. In Fig. 13, we conceptualize the formalism of the most popular PHIL approaches, with a few example schematics shown to provide a clear overview of popular approaches. A standard DNN is shown for comparison in Fig. 13(a) as it an approach devoid of any physics knowledge and therefore a "black-box" model. In order to increase physics knowledge, in Fig. 13(b), we show incorporation of knowledge as inputs to a DNN. The Neural Operator is next highlighted in Fig. 13(c) and learns mappings between function spaces and therefore can be more interpretable than conventional DNNs. Finally, we show a representation of PINNs in Fig. 13(d), which embeds physical laws as differential equations directly into the loss function during training. The PINN therefore is trained to satisfy both observed data and underlying physical constraints. It is important to note that PINNs are solution-oriented and trained to solve a specific PDE, whereas Neural operator approaches aim to learn the underlying physical principles governing a system. Indeed, the integration of multiple types of physics into the PHIL framework demonstrates independent performance boosts from each. Although methods that use multiple points of integration have shown good performance and have overcome some hurdles, many of them are still data-driven. As such, they still face the curse of dimensionality and model scalability issues which must be overcome for further advance.

V. OUTLOOK

PHIL integrates physical knowledge into DNNs to solve both forward and inverse AEM problems with significantly higher data efficiency and enhanced interpretability. In this review, we have provided a PHIL taxonomy based on DL workflows, where we aimed to outline a clear picture of the process, systematically delineating the existing approaches and building a framework that can be readily applied to develop new future methods. We highlighted different integration approaches to provide diverse examples of PHIL in AEMs, which serve as the pivotal link between AEM physics and DNN models. Now that we have illustrated the substantial achievements of PHIL in AEM, we will discuss some open challenges and potential future directions.

A. Open challenges

1. Open-access data and benchmark

As advocated by previous reviews on DL for AEMs,[1,3,18] the lack of open access to data and code impedes progress, and the same applies to PHIL for AEMs. Open-access resources, such as ImageNET,[102] UCI data repository,[103] and PDEBench for scientific computing,[104] are crucial as they allow the standardization of the PHIL workflow, the simplification of architecture differences, the enhancement of model/knowledge transfer, and facilitate the benchmarking of PHIL approaches. Developing robust and generalizable PHIL models remains challenging without sufficient shared data and codes. Additionally, while all AEM data are derived from Maxwell's equations, the different methods used for data generation can introduce deviations. These discrepancies hinder the seamless integration of AEM data, complicating efforts to build cohesive and comprehensive models.

A critical step toward collaboratively advancing PHIL in AEM is the development of more generalized AEM physical representations
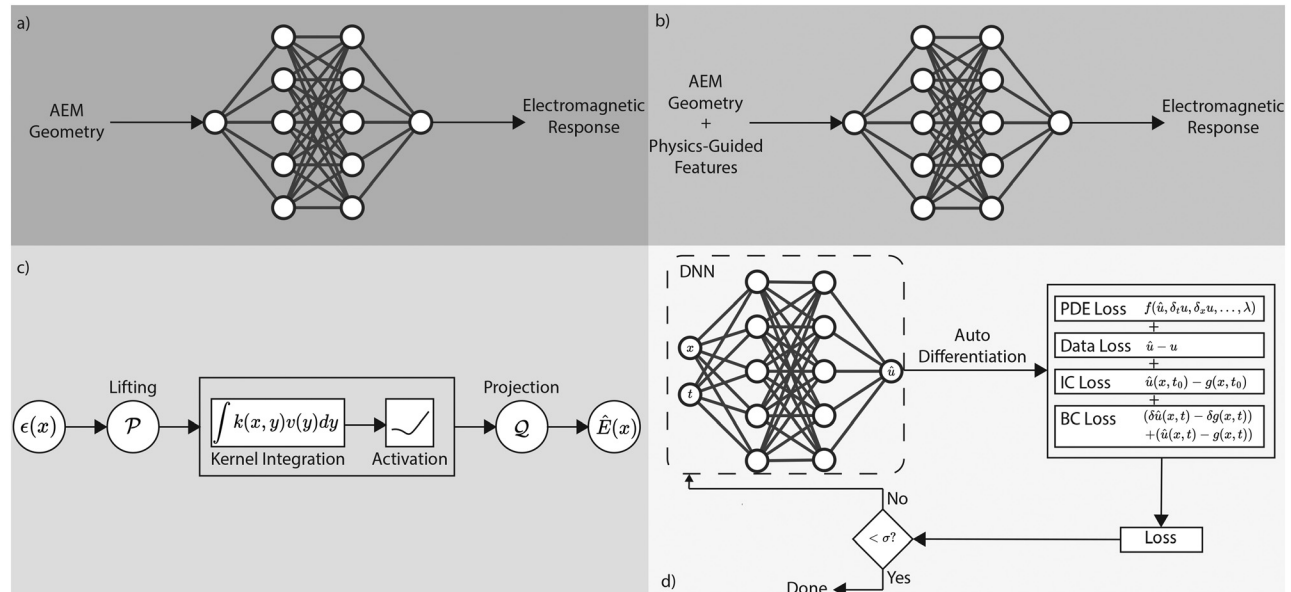
FIG. 13. (a) "Black-box" model—conventional DNN learns the mapping between pairs of input and output, typically AEM geometry and corresponding electromagnetic responses. (b) Feature engineered PHIL model—DNN takes more informative physical inputs such as polarization, geometry ratios, and dispersion relations other than just AEM geometry. (c) General representation of neural operator as an example of hypothesis set integration PHIL model. $\mathcal{P}$, $\mathcal{Q}$ are lifting and projection operators that are commonly nonlinear in the form of fully connected layers. The core of neural operator is the integral layer which includes both the kernel integration and nonlinear activation. We can treat FNO as a special type of neural operator. (d) General representation of PINN as an example of loss integration PHIL model. The DNN portion of the PINN learns the mapping from spatial coordinates $\mathbf{x}$ and time t to the surrogate solution $\hat{\mathbf{u}}$ of the PDE, assuming the PDE has both spatial and time derivative. With the help of auto differentiation, we can compute the PDE loss, where λ is PDE parameters. IC loss stands for initial condition loss at $t = t_0$. BC loss stands for boundary condition loss.

that harmonize the fusion of data from different generation methods. Currently, myriad physical representations are used, complicating PHIL models and limiting generalizability. A comprehensive benchmark is almost impossible for PHIL in AEM problems. The most straightforward option for the uniform representation of AEMs could be discretization in spatial coordinates, which is the same as that is used in CEMS. However, using spatial discretization could easily bring about scalability issues that lead us to the next open challenge.

2. Scalability, multiphysics/multiscale

So far, we have seen the application of PHIL models to different AEM geometry representations, such as 1D geometrical parameters, 2D binary images, 2D discretization of material properties, and small 3D domains. These examples demonstrate successful PHIL implementation but involve relatively small physical spaces or discretization sizes, while the network sizes are already substantial in size. When larger scale problems and higher-resolution discretizations are needed, the scalability of DNNs becomes a roadblock. The required DNN size to handle large problems can quickly outgrow the number of parameters in the network, leading to impractical computational demands—the curse of dimensionality.[15,105]

This scalability issue is exacerbated in multiscale and multiphysics problems. For instance, accurately capturing the responses and coupling of metasurface unit cells in a large-area metasurface requires high-resolution discretization. The large-area fine-scaled discretization results in a DNN model size that grows exponentially to computationally prohibitive levels. This problem also exists in multiphysics

problems, where different discretizations might be needed for different physics (e.g., electromagnetics and thermodynamics). Moreover, a single model attempting to fit all these requirements could heavily overload the capabilities of the DNN.

A more feasible approach is to use individual DNNs to learn each specific physical phenomenon and then combine these models. This strategy, however, introduces additional model training requirements and complexities in integrating the different trained DNNs. Developing methods to efficiently combine these individual models while managing the increased computational load is crucial for advancing PHIL in large-scale and complex AEM applications. Ideally, a large model that leverages both large datasets[106] and explicit physics guidance would handle diverse geometry or even multiphysics problems comprehensively. However, the need for large, reliable scientific datasets in AEM has hindered the development of such models.[1,3] Nonetheless, some recent data-driven efforts aim to enhance data efficiency[107] and interpretability in AEM by integrating explicit physics knowledge, signifying progress in this domain.

3. Integration of physics into optimization

The integration of optimization techniques into PHIL for AEMs is an area that has yet to be fully explored. Currently, existing optimization methods such as Adam[108] and RMSprop[109] are used for PHIL without specific adaptations. Exploring how to integrate physical knowledge into the optimization framework could be crucial, which may involve creating new algorithms or modifying existing ones to align better with the physics governing AEMs.

4. Interpretability

Although we have seen successful integration of physics knowledge into DNNs, PHIL models remain fundamentally data/physics-driven black boxes. From feature engineering to loss integration, the enhanced interpretability mainly manifests as human understanding that the model operates within a physically meaningful hypothesis set or loss space guided by integrated physics. However, there is still a lack of insight into how these DNNs fit physics data. Despite the improvements, the underlying workings of PHIL models and their alignment with physics are not fully transparent, leaving room for further research to make these models more interpretable and to provide deeper insights into AEM physics.

5. PINN Implementation

PINNs are one of the most widely studied PHIL approaches in the literature, likely due to their theoretical need for little or no training data from simulation. However, the implementation of PINNs in practice typically comes with major challenges.[110,111] PINNs often exhibit prolonged training time[22,23]; for instance, addressing an inverse scattering problem in metamaterials can require 150,000 epochs to reach convergence.[8] Efforts to mitigate these issues are ongoing and currently a common topic in the PINN field.[112–114] Additionally, while data-free PINNs promise independence from empirical data, there is no guarantee of stable performance,[48,110,115] and performance often rises significantly from incorporating field observations. Despite being mesh-free and leveraging backpropagation,[22] the reliance on data combined with long convergence times diminishes their advantage over CEMS. Another point of consideration is the deterministic nature of Maxwell's equations (see Sec. VI A), which govern electromagnetic phenomena, in contrast to the more stochastic behaviors seen in fluid dynamics governed by Navier–Stokes equations. This deterministic factor makes the use of PINNs in electromagnetic contexts less critical. Furthermore, PINNs are tailored to specific geometries, requiring a unique, well-defined PDE for each field component within the solution space. Consequently, each new AEM geometry necessitates training a new PINN model, hindering the goal of reducing simulation times. Nonetheless, the introduction of PINNs is recent, and an expanding range of work is ongoing to transform PINNs into more effective techniques.[116–123]

B. Future directions

Despite the aforementioned open challenges of PHIL in AEMs, PHIL has successfully improved data efficiency and enhanced interpretability. Rather than interfering with future development, these challenges spotlight exciting opportunities to achieve streamlined AEM design and deepen our understanding of AEM physics in potentially new dimensions. We briefly list a few future directions for PHIL in AEM problems.

1. Purely physics-based learning

This review primarily focused on PHIL approaches that balance data and physics, bypassing purely data-driven deep learning, as many reviews[1,18,124] have covered this area. However, on the other end of the PHIL spectrum, purely physics model-based learning has received far less attention. In Ref. 59, the only purely physics model-based

approach listed in Table II, the model demonstrates orders-of-magnitude advantages over deep learning approaches in terms of accuracy, data requirements, and model size when applied to massively programmable complex media. Sol *et al.*[59] achieve frugal coherent wave control in metasurface-programmable complex media using two key components: (1) a compact physics-based forward model and (2) estimation/calibration of the forward model. The forward model is grounded in the coupled-mode theorem, offering a compact physical representation of the complex media without requiring a large number of parameters typical of deep learning models. While estimating the forward model still necessitates some data collection, the data requirement is orders of magnitude smaller than for deep learning methods. The success of this physics model-based approach suggests an exciting yet largely unexplored area within PHIL for AEMs.

2. Hybrid integration

Hybrid integration is likely the simplest and most effective way to improve current PHIL performance. The most straightforward approach combines existing methods, such as feature engineering, hypothesis set, and loss function integration. We can create more robust and accurate models by leveraging these techniques together. As highlighted in the open challenges, developing optimization-integrated PHIL is an exciting area. This approach could better align the gradient descent nature of DNNs with the second-order derivatives of Maxwell's equations, potentially leading to significant improvements in model accuracy and physical consistency.

3. Physics discovery

This review did not cover physics discovery in detail, as it is a less explored area in PHIL for AEMs. The closest example in the literature is an inverse scattering problem in the PINN example,[8] where the original PINN refers to this inverse problem as physics discovery, since the model can learn the PDE parameters through data.[21] However, physics discovery through PINN requires prior knowledge of the PDE in the domain. If we can address the interpretability challenge and develop transparent models that can learn AEM physics while providing easily accessible information, transparent PHIL could be applied to physics discovery. Such models can offer insights into higher dimensions, potentially unlocking new understandings and advancements in AEMs.

4. AEM foundational model

We envision a large PHIL model trained on an extensive AEM dataset as an AEM foundational model capable of accurately handling various types of AEM problems. While it might seem counterintuitive to combine PHIL, which can potentially operate data-free, with a large foundational model that demands enormous data, the rationale lies in the high data requirements of such a model. Any data efficiency achieved through PHIL can significantly mitigate the cost of data collection, which is particularly high for AEMs due to the expense of CEMS and experimental data. There have been some studies that have provided open-source data and benchmarks,[1,3,18] which are particularly useful in facilitating the development of such large models. There are ongoing initiatives[104,125] within the scientific computing community in this direction, and we believe PHIL is a perfect candidate as the

underlying architecture of an AEM foundational model. By leveraging these open-source resources and benchmarks, a robust, efficient, and accurate foundational model for AEMs can be realized, driving significant advancements in the field.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Ethics Approval

The research in this manuscript did not involve human or animal subjects.

Author Contributions

Y. Deng: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **K. Fan:** Conceptualization (equal); Investigation (equal); Writing – original draft (equal); Writing – review & editing (equal). **B. Jin:** Writing – original draft (equal). **J. Malof:** Conceptualization (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Resources (equal); Supervision (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **W. J. Padilla:** Conceptualization (equal); Funding acquisition (equal); Investigation (equal); Supervision (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

APPENDIX: PHYSICS OF ARTIFICIAL ELECTROMAGNETIC MATERIALS

PHIL is a fast growing field that harnesses the power of artificial intelligence to uncover and predict complex physical phenomena. To achieve this, PHIL integrates foundational theories that govern the behavior of the systems under study. AEMs are a vast and rapidly evolving field, and PHIL is poised to play a transformative role in designing, optimizing, and understanding these materials. The fundamental equations that govern classical fields are Maxwell's equations. Related equations derived from Maxwell's equations include temporal coupled mode theory, multipole expansion, and Green's functions. While this review does not explore these theories in their full depth, it summarizes them and highlights how they are incorporated into PHIL. This integration of physics and deep learning is paving the way for breakthroughs in understanding and manipulating AEMs.

1. Maxwell's equations

The fundamental equations governing the behavior of electromagnetic fields, including their interaction with matter, are given in differential form as

$$\nabla \cdot \mathbf{D} = \rho, \tag{A1a}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{A1b}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \tag{A1c}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}, \tag{A1d}$$

where $\mathbf{E}$ and $\mathbf{B}$ are the electric field strength and magnetic flux density, respectively, $\mathbf{J}$ and ρ are the electric current density and charge density, respectively, and $\mathbf{D}$ and $\mathbf{H}$ are auxiliary fields and termed the electric displacement field, and the magnetic field strength, respectively. Bold symbols are vectors, and all quantities are functions of space ($\mathbf{r}$) and time (t).

Although electromagnetism can be described solely through the electric field ($\mathbf{E}$) and magnetic field ($\mathbf{B}$), there is a useful alternative way using scalar (ϕ) and vector ($\mathbf{A}$) potentials. The scalar potential (ϕ) simplifies calculations involving electric fields, and the vector potential (A) helps with magnetic fields. The usefulness of the potentials becomes clear for complex arrangements where solving for ($\mathbf{E}$) and ($\mathbf{B}$) directly can be challenging. The potentials (ϕ) and (A) provide an alternative approach that can often lead to faster—and more elegant—solutions. These potentials, however, are not unique and there are many possible choices for given $\mathbf{E}$ and $\mathbf{B}$ fields. This freedom is termed "gauge choice" and allows us to choose a convenient form for the potential that simplifies calculations for a specific problem. A common form for the potentials is

$$\mathbf{B} = \nabla \times \mathbf{A}, \tag{A2}$$

$$\mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t} - \nabla \phi. \tag{A3}$$

When electromagnetic fields interact with matter, the material may be electrically or magnetically polarized, where the later is termed magnetization. The microscopic polarization of a dielectric material is denoted as $\mathbf{P}(\mathbf{r}, t)$, and the microscopic magnetization of a material as $\mathbf{M}(\mathbf{r}, t)$. The auxiliary fields ($\mathbf{D}, \mathbf{H}$) are related to the vector fields ($\mathbf{E}, \mathbf{B}$) through the constitutive relations given as

$$\mathbf{D}(\mathbf{r}, t) = \varepsilon_0 \mathbf{E}(\mathbf{r}, t) + \mathbf{P}(\mathbf{r}, t), \tag{A4}$$

$$\mathbf{H}(\mathbf{r}, t) = \mu_0^{-1} \mathbf{B}(\mathbf{r}, t) - \mathbf{M}(\mathbf{r}, t), \tag{A5}$$

where ε_0 and μ_0 are the permittivity and permeability of free-space, respectively. Maxwell's equations [Eq. (A1)] and the auxiliary fields [Eqs. (A4) and (A5)] do not depend on the specific details of a material and therefore are universally applicable to non-magnetoelectric materials. However, there are twelve components of $\mathbf{E}, \mathbf{B}, \mathbf{D},$ and $\mathbf{H}$, and therefore, these equations alone are insufficient to determine the fields uniquely. We must therefore assume a particular form of the materials under study which can reduce the number of unknowns. A more specific form of the constitutive relations establishes relationships between the auxiliary fields $\mathbf{D}$ and $\mathbf{H}$

and the fields **E** and **B**. These relationships reduce the number of unknowns but, as mentioned, necessitate assumptions about the properties of the medium.

The superposition principle indicates that the general time dependent behavior of a wave can be expressed as a superposition of waves at different frequencies through a Fourier transform. Therefore, it suffices to consider the properties of a wave at a single frequency ω—termed the time-harmonic form. If we assume our matter is isotropic and linear, then in the frequency domain we can describe it by a polarization and magnetization that takes the form

$$\mathbf{P}(r, \omega) = \varepsilon_0 \chi_e \mathbf{E}(r, \omega), \tag{A6}$$

$$\mathbf{M}(r, t) = \chi_m \mathbf{H}(r, \omega), \tag{A7}$$

where χ_e and χ_m are the electric and magnetic susceptibility, respectively. With these assumptions, and suppressing the explicit space and frequency dependencies, the auxiliary fields may be written in the frequency domain as

$$\mathbf{D} = \varepsilon_0 \mathbf{E} + \varepsilon_0 \chi_e \mathbf{E} = \varepsilon_0 \varepsilon_r \mathbf{E} = \varepsilon \mathbf{E}, \tag{A8}$$

$$\mathbf{B} = \mu_0 \mathbf{H} + \mu_0 \chi_m \mathbf{H} = \mu_0 \mu_r \mathbf{H} = \mu \mathbf{H}. \tag{A9}$$

2. Drude–Lorentz model

The Drude–Lorentz model describes the response of materials to electromagnetic waves. It is typically derived by considering bound electrons in a driven damped harmonic oscillator model—analogous to a mass-spring system with friction. However, the Drude–Lorentz model may also be derived from first principles[126] and therefore is more fundamental than it would appear in most presentations. In the context of this review, we explore the use of Drude–Lorentz oscillators as a modeling framework to capture the effective material properties of AEMs. Specifically, we make the assumption that relative permittivity ε_r and relative permeability μ_r within Eqs. (A8) and (A9) embody the distinctive characteristics and responses of AEMs by employing the concept of Drude–Lorentz oscillators, which take the specific form

$$\varepsilon_r(\omega) = \varepsilon_\infty + \sum_{i=1}^{N} \frac{\omega_{p,i}^2}{\omega_{0,i}^2 - \omega^2 - i\omega\gamma_i}, \tag{A10}$$

$$\mu_r(\omega) = \mu_\infty + \sum_{i=1}^{N} \frac{\omega_{mp,i}^2}{\omega_{m0,i}^2 - \omega^2 - i\omega\gamma_{m,i}}, \tag{A11}$$

where ε_∞ and μ_∞ represent the high-frequency behavior of the permittivity and permeability, respectively, $\omega_{p,i}$ is the plasma frequency, $\omega_{0,i}$ is the resonance frequency, and γ_i is the damping factor of the ith resonator. Similar parameters describe the relative permeability, and we add an "m" sub-script to denote these terms. The model is general for N oscillators, where we form a sum over the second term on the right of Eqs. (A10) and (A11) where our sum runs from $i = 1$ to N.

Often metamaterials are fashioned to utilize the two lowest order resonant modes, where one is an "even" mode described by the ε_r, and the other is an "odd" mode described by the μ_r. The Drude–Lorentz parameters shown in Eqs. (A10) and (A11) are functions of the geometry and material properties of the metamaterial, where ω_p is proportional to the filing fraction (areal density for

metasurfaces or volume density for metamaterials), γ is related to the material loss and scattering rate, and ω_0 is inversely proportional to the unwound length of the resonator.

There are several advantages of the Drude–Lorentz model in describing AEM response. One is that the model is first principles (as mentioned) and causal. Therefore, using Eqs. (A10) and (A11) to fit experimental scattering data, i.e., $\tilde{r}$ and/or $\tilde{t}$, yields important information about the physics of the material under study. Another benefit is that ε and μ are the only material parameters that enter directly in Maxwell's equations (A1). Therefore, being able to describe materials in terms of the Drude–Lorentz model provides insight and a firm connection to the theory of electrodynamics. Further, all other material parameters: impedance (Z), refractive index (n), wave vector (k), conductivity (σ), etc., are related to $\{\varepsilon, \mu\}$ by simple algebra. Therefore, another benefit of the Drude–Lorentz model lies in that the frequency-dependent scattering can be well formulated by a few featured parameters. In terms of PHIL, one benefit of the Drude–Lorentz model is that the quantities $\{\omega_p, \gamma, \omega_0\}$ represent a reduced parameter set which can accurately describe relatively complex frequency dependent scattering.[34] This compact parameter latent space is much more efficient for embedding knowledge into the neural network thereby enabling significantly faster and more efficient forward and inverse predictions.

3. Transfer matrix

The transfer matrix method expresses the relationship between waves at an interface between two different types of homogeneous matter including conductors and dielectrics. If we imagine our interface is vertical, then we can say, for example, that the transfer matrix expresses the waves on the right side of an interface to those on the left. The boundary conditions from Maxwell's equations (A1) are used to equate these two sets of waves and express the result as a system of equations and a matrix—the transfer matrix—which gives the relationship between the two sets of waves. A powerful result of the transfer matrix is that of the *multiplication law*, which permits matrices from a general number of interfaces, each spaced apart by some distance, to be written as a single matrix through matrix multiplication. The implication is that the study of relatively complex material multilayer systems can then be studied. We show a form for the reflection coefficient (r) and transmission coefficient (t) from the transfer matrix below but do not derive the general equations explicitly. Rather, we refer the interested reader to a text on the topic.[127]

We consider the case of a flat slab of material of thickness d embedded in vacuum. The reflection and transmission coefficients calculated from the transfer matrix method for this material are

$$r = \frac{\frac{i}{2}[Z_r^{-1} - Z_r]\sin(nk_0 d)}{\cos(nk_0 d) - \frac{i}{2}[Z_r^{-1} + Z_r]\sin(nk_0 d)}, \tag{A12}$$

$$t = \frac{1}{\cos(nk_0 d) - \frac{i}{2}[Z_r^{-1} + Z_r]\sin(nk_0 d)}, \tag{A13}$$

where $Z_r = Z/Z_0$ is the relative impedance, Z_0 is the impedance of free-space, k_0 is the free-space wave vector, and n is the index of refraction. A material which is modeled as Drude–Lorentz using

Eqs. (A10) and (A11) may then use the transfer matrix Eqs. (A12) and (A13) by using the relationship between material parameters given by

$$Z_r = \sqrt{\frac{\mu_r}{\varepsilon_r}} \qquad (A14)$$

and

$$n = \sqrt{\varepsilon_r \mu_r}. \qquad (A15)$$

We also highlight that for optically thin (long wavelength limit) materials of thickness d embedded in vacuum, where $k_0 d \ll 1$, the transmission and reflection coefficients given by the transfer matrix Eqs. (A12) and (A13) can be simplified as

$$r = \frac{ik_0 d}{2}(\varepsilon_r - \mu_r), \qquad (A16)$$

$$t = 1 + \frac{ik_0 d}{2}(\varepsilon_r + \mu_r). \qquad (A17)$$

4. Temporal coupled mode theory

Temporal coupled-mode theory (TCMT) is a powerful tool used in AEMs, permitting the analysis and design of general resonant systems. Modern TCMT permits the additional investigation of many nonlinear interactions as well. TCMT serves as a versatile framework, particularly useful for describing idealized systems subjected to weak disturbances, akin to time-dependent perturbation theory. TCMT has become a valuable tool for investigating phenomena arising from the coupling of resonant systems, including Fano resonances and optical switching.[45,128,129] Here, we focus on a specific form of TCMT that describes coupled resonating systems through parameters representing resonant frequencies and coupling constants, without delving into the intricacies of the particular system. In this section, we will provide a concise overview of the application of TCMT to artificial electromagnetic materials. A general form of TCMT describing resonant cavities interacting with external ports is given as

$$\frac{da}{dt} = (-i\boldsymbol{\omega}_0 - \boldsymbol{\gamma}_r - \boldsymbol{\gamma}_i)a + \mathbf{K}^T s_{\text{in}}, \qquad (A18a)$$

$$s_{\text{out}} = \mathbf{C}s_{\text{in}} + \mathbf{F}a, \qquad (A18b)$$

where a is a complex vector representing the amplitude of the resonant mode with a time-harmonic dependence of $\exp(-i\omega t)$, matrices $\boldsymbol{\omega}_0$, $\boldsymbol{\gamma}_r$, and $\boldsymbol{\gamma}_i$ are the resonant frequencies, radiative decay rate, and intrinsic decay rate of the modes. Matrices $\mathbf{K}$ and $\mathbf{F}$ correspond to the input and output coupling of resonant modes to the ports, respectively. s_{in} is a vector with elements that represent inputs from each port where $|s_{\text{in}}|^2$ is equal to the input power, with similar terms for the outputs s_{out}. The matrix $\mathbf{C}$ represents a direct non-resonant pathway from port to port. Generally, for a lossless and reciprocal system, the matrix $\mathbf{C}$ is considered unitary and symmetric.

The scattering matrix gives the relationship between the inputs to systems and the resulting outputs.[130] For a two port system, the scattering matrix is

$$S = \frac{s_{out}}{s_{in}} = \begin{bmatrix} \tilde{s}_{11} & \tilde{s}_{12} \\ \tilde{s}_{21} & \tilde{s}_{22} \end{bmatrix}. \qquad (A19)$$

If we assume a reciprocal symmetric system with two modes of opposite symmetry (mode 1 = even, mode 2 = odd), then from Eq. (A18) we find a scattering matrix of

$$S = \left\{ I - \frac{\gamma_1}{i(\omega_1 - \omega) + \gamma_1 + \delta_1} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \\ - \frac{\gamma_2}{i(\omega_2 - \omega) + \gamma_2 + \delta_2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \right\} \begin{bmatrix} \tilde{r}_0 & \tilde{t}_0 \\ \tilde{t}_0 & \tilde{r}_0 \end{bmatrix} \qquad (A20)$$

where the terms $\tilde{r}_0$ and $\tilde{t}_0$, etc., are from the $\mathbf{C}$ matrix and equivalent to the ε_∞ and μ_∞ terms of the Drude–Lorentz model.

One metamaterial system that is well described by the TCMT Eq. (A18) is that of an absorber. The absorption (A) is given by

$$A = 1 - S^\dagger S. \qquad (A21)$$

To obtain unity absorption we see that in Eq. (A20) we must have $\omega_1 = \omega_2 \equiv \omega_0$ i.e., the even and odd resonators must be degenerate. We also find that if each of these modes is critically coupled $\delta = \gamma$, then we can achieve perfect absorption $A(\omega_0) = 1$, where the scattering matrix is null $S = [0, 0; 0, 0]$.

Like the Drude–Lorentz model presented in Subsection 2 of Appendix, an advantage of TCMT lies in that only a few parameters are needed to describe the frequency-dependent scattering. This compact parameter TCMT set can therefore also be used to efficiently embed knowledge into the deep learning workflow.

5. Multipole expansion

The interaction of electromagnetic fields with the unit-cell of an AEM induces a polarization current described by $\mathbf{J}_P = \partial \mathbf{P}/\partial t$. The induced $\mathbf{J}_P$ may then generate scattered E and H fields. $\mathbf{J}_P$ can be decomposed into various moments in Cartesian coordinates and, depending on the complexity of the polarization current, may consist of several non-zero terms including: electric dipole moment p, magnetic dipole moment m, electric quadruple tensor Q, magnetic quadruple tensor M, toroidal dipole moment T, etc. We do not derive or enumerate all of these terms and refer the interested reader to published work on this topic.[38,131,132] Here, we list only the dipole moments

$$p = \frac{i}{\omega} \int_V \mathbf{J}_P dV, \qquad (A22)$$

$$m = \frac{1}{2} \int_V \mathbf{r} \times \mathbf{J}_P dV, \qquad (A23)$$

where $\int_V (\cdot)dV$ is the volume integral, and r is the position vector.

If we have that our unit-cell is sub-wavelength and independent of polarization, we can assume a $\hat{x}$ polarized field ($E_{0,x}$) and $+\hat{z}$ propagation ($e^{ik_0 z}$), which may lead to several non-zero multipoles. These multipoles scatter in both the forward (E^+) and backward (E^-) direction, given by

$$E^+ = E_{0,x} e^{ik_0 z} + \frac{ik_0}{2A\varepsilon_0} e^{ik_0 z}\left[p_x + \frac{1}{c}m_y + \cdots\right], \qquad (A24)$$

$$E^- = e^{ik_0 z} \frac{ik_0}{2A\varepsilon_0} e^{ik_0 z} \left[p_x - \frac{1}{c} m_y + \cdots \right], \qquad (A25)$$

where A is the area of the unit cell.

If we take our unit-cell to be at $z = 0$ and ignore higher order multipoles, we may then calculate the reflection coefficient $r = E^-/E_{0,x}$ and transmission coefficient $t = E^+/E_{0,x}$

$$r = \frac{ik_0}{2\varepsilon_0 A} \left(p_x - \frac{1}{c} m_y \right), \qquad (A26)$$

$$t = 1 + \frac{ik_0}{2\varepsilon_0 A} \left(p_x + \frac{1}{c} m_y \right). \qquad (A27)$$

The multipole expansion technique is usually paired with computational electromagnetic simulations to determine the frequency (or wavelength) dependent $\mathbf{J}_P(\omega)$. Then, from $\mathbf{J}_P(\omega)$ the frequency or wavelength dependence of the multipoles may be calculated using Eqs. (A22) and (A23) and then r and t from Eqs. (A26) and (A27). This method—although computationally demanding—can reveal the intricate connection between the geometric properties and the scattering response of photonic structures. Consequently, optimizing an AEM to achieve a specific scattered state typically relies on parameter tuning and optimization, often driven by designer expertise. Furthermore, determination of the needed scatterer geometry in inverse design problems is a challenging task, as it is frequently ill-posed. To address this challenge, complex algorithms or approximations are often required. Notably, in recent AEM studies, there has been a surge in the development of various deep learning inverse techniques.[133] These methods aim to establish the underlying relationships between multipole scattering phenomena and the geometric features of AEMs, providing innovative solutions for inverse design and optimization.

6. Effective current sheets

The electromagnetic response of planar metamaterials and metasurfaces may be described by electric and magnetic surface current densities. Here, the surface current densities describe the response to both the conduction and polarization currents induced by external E and H fields. Past work has shown that an impedance boundary condition can be used to describe planar arrays of small particles.[134–136] This approach to planar metasurfaces assumes the existence of effective electric ($\mathbf{K}_e$) and magnetic ($\mathbf{K}_m$) current sheets, given as

$$\mathbf{K}_e = -i\omega \frac{\mathbf{p}}{A}, \qquad (A28)$$

$$\mathbf{K}_m = -i\omega \mu_0 \frac{\mathbf{m}}{A}, \qquad (A29)$$

where $\mathbf{p}$ and $\mathbf{m}$ are the effective electric and magnetic dipole moments, respectively, and A is the unit-cell area.[57]

We next consider a metamaterial which consists of two infinite parallel sheets, separated by a distance d, which supports $\mathbf{K}_e$ and $\mathbf{K}_m$ current sheets. The distance d is much smaller than the free-space wavelength of an incident electromagnetic wave, which travels parallel to the surface normal of the sheets. Therefore, we assume an

optically thin (long-wavelength limit) metamaterial. We assume the electric field is polarized in the $\hat{x}$ direction, the magnetic field in the $\hat{y}$ direction, and therefore the wave is propagating in the $\hat{z}$ direction.

The magnetic field strength (H) and the surface current density (K) have the same units, and therefore, the electric field can be written as $E = ZH = ZK_e$. Thus, the reflected and transmitted electric fields can be written as[136]

$$E_r = -\frac{1}{2} Z_0 \mathbf{K}_e - \frac{1}{2} \hat{z} \times \mathbf{K}_m, \qquad (A30)$$

$$E_t = E_i - \frac{1}{2} Z_0 \mathbf{K}_e + \frac{1}{2} \hat{z} \times \mathbf{K}_m. \qquad (A31)$$

Therefore, we can find the reflection coefficient ($r = E_r/E_i$) and transmission coefficient ($t = E_t/E_i$) by assuming a normalized field intensity (E_i) as[10]

$$r = \frac{ik_0}{2\varepsilon_0 A} \left(p_x - \frac{1}{c} m_y \right), \qquad (A32)$$

$$t = 1 + \frac{ik_0}{2\varepsilon_0 A} \left(p_x + \frac{1}{c} m_y \right), \qquad (A33)$$

where p_x is the effective electric dipole in the $\hat{x}$ direction, and m_y is the effective magnetic dipole in the $\hat{y}$ direction. Since the electric polarization $\mathbf{P}$ in Eq. (A6) is the number of dipoles per unit volume, and the magnetization $\mathbf{M}$ in Eq. (A7) is the number of magnetic dipoles per unit volume, we see that $p_x \propto \varepsilon_r$ and $m_y \propto \mu_r$. Indeed, upon inspection, we find a similar form for the reflection coefficient shown in Eq. (A6) to that in Eq. (A32), and in the transmission coefficient displayed in Eq. (A17) compared to Eq. (A33). Results for the reflection and transmission coefficients derived in the *multipole expansion method* in Sec. VI E with only dipole contributions is identical to those of the *effective current sheet* method.

7. Model fitting examples

To demonstrate the accuracy of some of the physical models described in Sec. VI, and to highlight their benefit for AEM studies, we provide an example of their use. We use both the Drude–Lorentz [Eqs. (A10) and (A11)] and TCMT Eq. (A20) models to fit the frequency dependent absorption spectra of an all-dielectric metasurface absorber.

The all-dielectric metasurface absorber we chose to study consists of an infinite square array of cylindrical silicon resonators, each having a radius of $r = 61.9\,\mu m$, a height of $h = 51.5\,\mu m$, and a unit-cell periodicity of $p = 217.1\,\mu m$. The metasurface geometry is optimized to achieve a peak absorption at a frequency near 1 THz, shown as the solid black curve in Fig. 14(a). In Fig. 14, we show simulated spectra as the solid black curves, and Drude–Lorentz fits as the dashed red curves. For the Drude–Lorentz model, we use a total of two oscillators (on in ε_r and one in μ_r) to fit the spectra. Each oscillator has three parameters: ω_0, ω_p, and γ, with all oscillators sharing common values for ε_∞ and μ_∞. The relative permittivity and permeability derived from the Lorentz model [Eqs. (A10) and (A11)] are then used to calculate n and Z_r from Eqs. (A14) and (A15), respectively, then input into the transfer matrix Eqs. (A12) and (A13) to obtain the spectra. In Figs. 14(b) and 14(c), we show
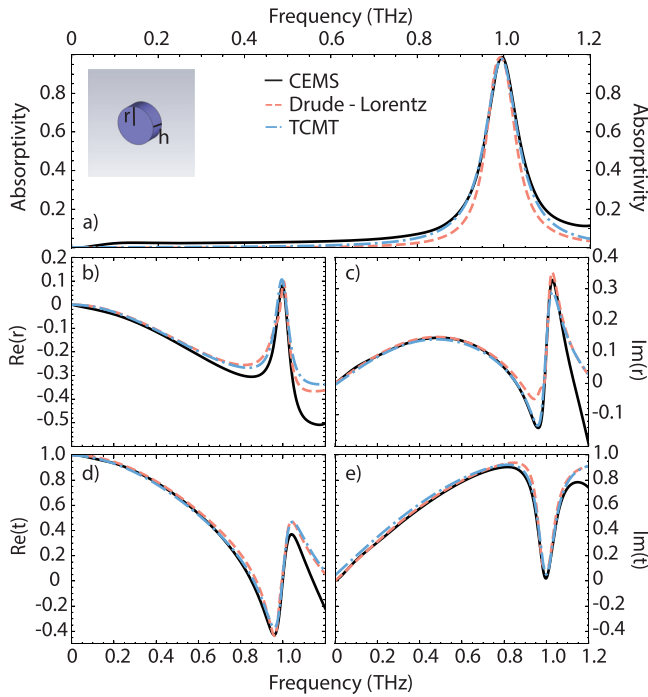
FIG. 14. Spectra fit of Drude–Lorentz (dashed red curve) and TCMT (dashed blue curve) models to CEMS (black curve) for (a) absorptivity, with the geometry of all dielectric metasurface shown in inset, (b) real part of the reflection coefficient, (c) imaginary part of the reflection coefficient, (d) real part of the transmission, and (e) imaginary part of the transmission of an all-dielectric metasurface absorber with an absorption peak near 1 THz.

TABLE IV. Table of Lorentz and TCMT parameters for fitting of the real and imaginary parts of the reflection and transmission coefficients shown in Fig. 14. The bottom row shows the error in each of these model fits, as described in the text.

Drude–Lorentz		TCMT		-
Parameter	Value	Parameter	Value	Units
ε_∞	1.86	ε_∞	1.85	$\cdots$
μ_∞	0.96	μ_∞	0.95	$\cdots$
$\omega_{0,1}$	0.99	ω_{01}	1.00	$2\pi \cdot \text{THz}$
$\omega_{p,1}$	0.256	γ_1	0.025	$2\pi \cdot \text{THz}$
γ_1	0.035	δ_1	0.021	$2\pi \cdot \text{THz}$
$\omega_{m0,1}$	0.997	ω_{02}	0.99	$2\pi \cdot \text{THz}$
ω_{mp1}	0.277	γ_2	0.020	$2\pi \cdot \text{THz}$
$\gamma_{m,1}$	0.031	δ_2	0.025	$2\pi \cdot \text{THz}$
$\cdots$	$\cdots$	β_1	0	Rad
$\cdots$	$\cdots$	β_2	π	Rad
$\mathscr{L}$	$3.12e^{-3}$	$\mathscr{L}$	$3.33e^{-3}$	MSE

the Drude–Lorentz fits to $\text{Re}(r)$ and $\text{Im}(r)$, respectively, and Figs. 14(d) and 14(e) show the Drude–Lorentz fits to $\text{Re}(t)$ and $\text{Im}(t)$, respectively.

Similarly, we use two resonators in the TCMT model [Eq. (A20)] to fit the same spectra. Fits to the spectra are shown in Fig. 14 as the dashed blue curves. The non-resonant matrix **C** contains the terms r_0 and t_0 which are the background reflection coefficient and background transmission coefficient, respectively. We calculate these terms using Eqs. (A12) and (A13), respectively, where we use $\varepsilon_r = \varepsilon_\infty$ and $\mu_r = \mu_\infty$ in Eqs. (A14) and (A15). The matrix Ω is defined as $\omega \mathbf{I}$, Ω_0 as $\omega_0 \mathbf{I}$, and Γ as $(\gamma_r + \gamma_i)\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. We assume anti-symmetry for odd/even modes at the same resonance frequency in TCMT. The matrices **F** and **K** in Eq. (A18) can be derived from **C**, Ω, Ω_0, and Γ. We fit the model to the spectra of the absorber using nonlinear least squares fitting, and the resulting parameter values are shown in Table IV. We report the final fitting MSE on the concatenated array, $RT = [\text{Re}(\tilde{r}), \text{Im}(\tilde{r}), \text{Re}(\tilde{t}), \text{Im}(\tilde{t})]$, and the MSE is defined as $\mathscr{L} = \overline{|\Delta RT|^2}$.

8. Synopsis

The various models are often used in relevant studies and presented in Sec. VI, and each has its strong points for understanding of the physics underlying the nature of artificial electromagnetic

materials. Although all of these models are derived from Maxwell's equations (11), each makes various approximations, which is most often the atomically thin or long-wavelength limit assumption used in the effective current sheet and multipole expansion models.

Although the Drude–Lorentz model is not a fundamental law of physics such as Newton's laws or Maxwell's equations, it can be considered a *governing equation* that describes the behavior of materials under the influence of electromagnetic radiation. The Drude–Lorentz model is causal and based on classical mechanics and electromagnetism, and it involves differential equations that relate to the electric field, polarization, and current density. It can be used to predict the optical properties of materials, such as refractive index, absorption coefficient, and reflection and transmission coefficients. However, it is an approximation, as it simplifies the complex interactions within materials to the two parameters ε and μ and therefore assumes homogeneous matter. None-the-less, the Drude–Lorentz model Eqs. (A10) and (A11) provides a valuable framework for understanding many optical phenomena. When combined with the transfer matrix method Eqs. (A12) and (A13), it can predict scattering without a long-wavelength assumption, unlike other models. Further, a modified Drude–Lorentz form of equations that takes into account spatial dispersion[137] can describe the response of AEMs with material parameters of $\tilde{\varepsilon}(\omega, k)$, $\tilde{\mu}(\omega, k)$ and is an excellent fit to both metallic-based metamaterials[138] and all-dielectric metasurfaces.[34]

Likewise, temporal coupled mode theory has predictive properties for the electromagnetic response of materials and is also based on a firm mathematical foundation and Maxwell's equations.[36] Further, TCMT obeys the conservation of energy, reciprocity, and causality and has been successfully applied to a wide range of optical systems, demonstrating its robustness and predictive power.[139] A benefit afforded by TCMT is understanding of the symmetry of resonances in AEM systems and the importance of radiative and material loss damping rates, as well as the degeneracy of modes.[45] This can be important for understanding of perfect absorbers[140,141] and coherent perfect absorbers.[142,143]

The multipole expansion method is useful for understanding of the relevance of various moments (dipole, quadrupole, etc.) from which from which the scattering of electromagnetic waves by small particles and sub-wavelength unit cells of AEMs is possible. Such an understanding can help to determine the contribution of multipoles to the scattering and absorption cross sections and for the design of AEM systems to achieve directional scattering through constructive or destructive interference in preferred directions, similar to the Kerker conditions.[144] However, the number of exactly solvable problems with the multipole expansion method is limited, and thus one usually resorts to computational electromagnetic simulations to determine the polarization current density ($\mathbf{J}_P$), which then permits the calculation of the frequency (or wavelength) dependence of multipoles supported by the AEM system under study. The geometry of the AEM or particle may then be tuned to realize various scattered states including the First and second Kerker conditions,[144] invisibility[145] (or antireflective),[146] and perfect absorption.[147] Another result of the multipole expansion method is the description of the reflection coefficient [Eq. (A26)] and transmission coefficient [Eq. (A27)], which is similar to the results from the transfer matrix method in the long-wavelength limit Eqs. (A16) and (A17), and which is identical to the effective current sheet method Eqs. (A32) and (A33).

REFERENCES

[1] O. Khatib, S. Ren, J. Malof, and W. J. Padilla, "Deep learning the electromagnetic properties of metamaterials—A comprehensive review," Adv. Funct. Mater. **31**, 2101748 (2021).

[2] S. So, T. Badloe, J. Noh, J. Bravo-Abad, and J. Rho, "Deep learning enabled inverse design in nanophotonics," Nanophotonics **9**, 1041–1057 (2020).

[3] J. Jiang, M. Chen, and J. A. Fan, "Deep neural networks for the evaluation and design of photonic devices," Nat. Rev. Mater. **6**, 679–700 (2020).

[4] P. R. Wiecha, A. Arbouet, C. Girard, and O. L. Muskens, "Deep learning in nano-photonics: Inverse design and beyond," Photonics Res. **9**, B182 (2021).

[5] 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).

[6] W. Ma, F. Cheng, and Y. Liu, "Deep-learning-enabled on-demand design of chiral metamaterials," ACS Nano **12**, 6326–6334 (2018).

[7] Z. Liu, D. Zhu, S. P. Rodrigues, K.-T. Lee, and W. Cai, "Generative model for the inverse design of metasurfaces," Nano Lett. **18**, 6570–6576 (2018).

[8] Y. Chen, L. Lu, G. E. Karniadakis, and L. Dal Negro, "Physics-informed neural networks for inverse problems in nano-optics and metamaterials," Opt. Express **28**, 11618 (2020).

[9] 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).

[10] W. J. Padilla and K. Fan, *Metamaterial Electromagnetic Wave Absorbers* (Springer Nature, 2022).

[11] P. Moitra, B. A. Slovick, W. li, I. I. Kravchencko, D. P. Briggs, S. Krishnamurthy, and J. Valentine, "Large-scale all-dielectric metamaterial perfect reflectors," ACS Photonics **2**, 692–698 (2015).

[12] H. Chen, C. T. Chan, and P. Sheng, "Transformation optics and metamaterials," Nat. Mater. **9**, 387–396 (2010).

[13] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from Data* (AMLBook, New York, NY, 2012), Vol. 4.

[14] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," Proc. IEEE **111**, 465–499 (2023).

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, London, UK, 2016).

[16] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with alphafold," Nature **596**, 583–589 (2021).

[17] G. B. Goh, N. O. Hodas, and A. Vishnu, "Deep learning for computational chemistry," J. Comput. Chem. **38**, 1291–1307 (2017).

[18] W. Ma, Z. Liu, Z. A. Kudyshev, A. Boltasseva, W. Cai, and Y. Liu, "Deep learning for the design of photonic structures," Nat. Photonics **15**, 77–90 (2021).

[19] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," Commun. ACM **64**, 107–115 (2021).

[20] R. Shwartz-Ziv and N. Tishby, "Opening black box deep neural networks via information" (2017).

[21] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Comput. Phys. **378**, 686–707 (2019).

[22] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," Nat. Rev. Phys. **3**, 422–440 (2021).

[23] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics–informed neural networks: Where we are and what's next," J. Sci. Comput. **92**, 88 (2022).

[24] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?': Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16* (ACM, 2016).

[25] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," Nat. Mach. Intell. **1**, 206–215 (2019).

[26] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via deeponet based on the universal approximation theorem of operators," Nat. Mach. Intell. **3**, 218–229 (2021a).

[27] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier neural operator for parametric partial differential equations," in Proceedings of the International Conference on Learning Representations (2021).

[28] W. Ji, J. Chang, H.-X. Xu, J. R. Gao, S. Gröblacher, H. P. Urbach, and A. J. L. Adam, "Recent advances in metasurface design and quantum optics applications with machine learning, physics-informed neural networks, and topology optimization methods," Light. Sci. Appl. **12**, 169 (2023).

[29] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," Proc. Natl. Acad. Sci. U. S. A. **113**, 3932–3937 (2016).

[30] S.-M. Udrescu and M. Tegmark, "AI Feynman: A physics-inspired method for symbolic regression," Sci. Adv. **6**, eaay2631 (2020).

[31] Z. Chen, A. Ogren, C. Daraio, L. C. Brinson, and C. Rudin, "How to see hidden patterns in metamaterials with interpretable machine learning," Extreme Mech. Lett. **57**, 101895 (2022a).

[32] G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman, and D. Shih, "Machine learning in the search for new fundamental physics," Nat. Rev. Phys. **4**, 399–412 (2022).

[33] J. Jiang and J. A. Fan, "Global optimization of dielectric metasurfaces using a physics-driven neural network," Nano Lett. **19**, 5366–5372 (2019a).

[34] O. Khatib, S. Ren, J. Malof, and W. J. Padilla, "Learning the physics of all-dielectric metamaterials with deep Lorentz neural networks," Adv. Opt. Mater. **10**, 2200097 (2022).

[35] A. Sommerfeld, *Electrodynamics: Lectures on Theoretical Physics* (Academic Press, San Diego, CA, 2013).

[36] H. A. Haus, *Waves and Fields in Optoelectronics*, Prentice-Hall series in solid state physical electronics (Prentice Hall, Old Tappan, NJ, 1983).

[37] J. D. Jackson, *Classical Electrodynamics, International Adaptation*, 3rd ed. (John Wiley & Sons, Nashville, TN, 2021).

[38] A. Zangwill, *Modern Electrodynamics* (Cambridge University Press, Cambridge, UK, 2012).

[39] C. A. Balanis, *Advanced Engineering Electromagnetics*, 2nd ed. (John Wiley & Sons, Chichester, UK, 2012).

22 March 2025 09:36:26

[40]P. Markos and C. M. Soukoulis, *Wave Propagation* (Princeton University Press, Princeton, NJ, 2008).

[41]N. Engheta and R. W. Ziolkowski, eds., *Metamaterials* (John Wiley & Sons, Chichester, UK, 2006).

[42]B. E. A. Saleh and M. C. Teich, *Fundamentals of Photonics*, 3rd ed., Wiley Series in Pure and Applied Optics (John Wiley & Sons, Nashville, TN, 2019).

[43]B. T. Draine and P. J. Flatau, "Discrete-dipole approximation for scattering calculations," J. Opt. Soc. Am. A **11**, 1491 (1994).

[44]H. Haus and W. Huang, "Coupled-mode theory," Proc. IEEE **79**, 1505–1518 (1991).

[45]S. Fan, W. Suh, and J. D. Joannopoulos, "Temporal coupled-mode theory for the Fano resonance in optical resonators," J. Opt. Soc. Am. A **20**, 569 (2003).

[46]Y. Chen and L. Dal Negro, "Physics-informed neural networks for imaging and parameter retrieval of photonic nanostructures from near-field data," APL Photonics **7**, 010802 (2022).

[47]M. Chen, R. Lupoiu, C. Mao, D.-H. Huang, J. Jiang, P. Lalanne, and J. A. Fan, "High speed simulation and freeform optimization of nanophotonic devices with physics-augmented deep learning," ACS Photonics **9**, 3110–3123 (2022b).

[48]M. Zhelyeznyakov, J. Fröch, A. Wirth-Singh, J. Noh, S. Rho, S. Brunton, and A. Majumdar, "Large area optimization of meta-lens via data-free machine learning," Commun. Eng. **2**, 60 (2023).

[49]E. Khoram, Z. Wu, Y. Qu, M. Zhou, and Z. Yu, "Graph neural networks for metasurface modeling," ACS Photonics **10**, 892–899 (2022).

[50]L. Kuhn, T. Repän, and C. Rockstuhl, "Exploiting graph neural networks to perform finite-difference time-domain based optical simulations," APL Photonics **8**, 036109 (2023).

[51]Y. Tang, J. Fan, X. Li, J. Ma, M. Qi, C. Yu, and W. Gao, "Physics-informed recurrent neural network for time dynamics in optical resonances," Nat. Comput. Sci. **2**, 169–178 (2022).

[52]P. N. Prasad, *Nanophotonics* (Wiley-Interscience, New York, NY, 2004).

[53]S. Maier, *Plasmonics: Fundamentals and Applications* (Springer, New York, NY, 2007).

[54]C. Simovski and S. Tretyakov, *An Introduction to Metamaterials and Nanophotonics* (Cambridge University Press, Cambridge, UK, 2020).

[55]Y. Song, D. Wang, J. Qin, J. Li, H. Ye, Z. Zhang, X. Chen, M. Zhang, and A. C. Boucouvalas, "Physical information-embedded deep learning for forward prediction and inverse design of nanophotonic devices," J. Lightwave Technol. **39**, 6498–6508 (2021).

[56]P. Naseri, S. Pearson, Z. Wang, and S. V. Hum, "A combined machine-learning/optimization-based approach for inverse design of nonuniform bianisotropic metasurfaces," IEEE Trans. Antennas Propag. **70**, 5105–5119 (2022).

[57]S. Pearson, P. Naseri, and S. V. Hum, "A beam-splitting bianisotropic metasurface designed by optimization and machine learning," IEEE Open J. Antennas Propag. **3**, 798–811 (2022).

[58]S. Noureen, I. H. Syed, S. Ijaz, A. A. Abdellatif, H. Cabrera, M. Zubair, Y. Massoud, and M. Q. Mehmood, "Physics-driven tandem inverse design neural network for efficient optimization of UV–Vis meta-devices," Appl. Surf. Sci. Adv. **18**, 100503 (2023).

[59]J. Sol, H. Prod'homme, L. Le Magoarou, and P. del Hougne, "Experimentally realized physical-model-based frugal wave control in metasurface-programmable complex media," Nat. Commun. **15**, 2841 (2024).

[60]A.-P. Blanchard-Dionne and O. J. F. Martin, "Teaching optics to a machine learning network," Opt. Lett. **45**, 2922–2925 (2020).

[61]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**, 329–338 (2020).

[62]C. Majorel, C. Girard, A. Arbouet, O. L. Muskens, and P. R. Wiecha, "Deep learning enabled strategies for modeling of complex aperiodic plasmonic metasurfaces of arbitrary size," ACS Photonics **9**, 575–585 (2022).

[63]S. Li, Z. Liu, S. Fu, Y. Wang, and F. Xu, "Intelligent beamforming via physics-inspired neural networks on programmable metasurface," IEEE Trans. Antennas Propag. **70**, 4589–4599 (2022).

[64]B. Liang, D. Xu, N. Yu, Y. Xu, X. Ma, Q. Liu, M. S. Asif, R. Yan, and M. Liu, "Physics-guided neural-network-based inverse design of a photonic–plasmonic nanodevice for superfocusing," ACS Appl. Mater. Interfaces **14**, 27397–27404 (2022).

[65]Y. Hu, Y. Jin, X. Wu, and J. Chen, "A theory-guided deep neural network for time domain electromagnetic simulation and inversion using a differentiable programming platform," IEEE Trans. Antennas Propag. **70**, 767–772 (2022).

[66]Y. Augenstein, T. Repän, and C. Rockstuhl, "Neural operator-based surrogate solver for free-form electromagnetic inverse design," ACS Photonics **10**, 1547–1557 (2023).

[67]J. Zhang, J. W. You, F. Feng, W. Na, Z. C. Lou, Q.-J. Zhang, and T. J. Cui, "Physics-driven machine-learning approach incorporating temporal coupled mode theory for intelligent design of metasurfaces," IEEE Trans. Microwave Theory Tech. **71**, 2875–2887 (2023).

[68]Y. Xu, J. Yang, K. Fan, S. Wang, J. Wu, C. Zhang, D. Zhan, W. J. Padilla, B. Jin, J. Chen, and P. Wu, "Physics-informed inverse design of programmable metasurfaces," Adv. Sci. **11**, 2406878 (2024).

[69]Z. Fang and J. Zhan, "Deep physical informed neural networks for metamaterial design," IEEE Access **8**, 24506–24513 (2020).

[70]L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson, "Physics-informed neural networks with hard constraints for inverse design," SIAM J. Sci. Comput. **43**, B1105–B1132 (2021b).

[71]J. Lim and D. Psaltis, "MaxwellNet: Physics-driven deep neural network training based on Maxwell's equations," APL Photonics **7**, 011301 (2022).

[72]C. Dove, J. Boondicharern, and L. Waller, "Physics-guided training of neural electromagnetic wave simulators with time-reversal consistency," in *NeurIPS 2023 Workshop on Deep Learning and Inverse Problems* (2023).

[73]C. Gigli, A. Saba, A. B. Ayoub, and D. Psaltis, "Predicting nonlinear optical scattering with physics-driven neural networks," APL Photonics **8**, 026105 (2023).

[74]Y.-D. Hu, X.-H. Wang, H. Zhou, L. Wang, and B.-Z. Wang, "A more general electromagnetic inverse scattering method based on physics-informed neural network," IEEE Trans. Geosci. Remote Sens. **61**, 1–9 (2023).

[75]V. Medvedev, A. Erdmann, and A. Rosskopf, "Modeling of near-and far-field diffraction from EUV absorbers using physics-informed neural networks," in *2023 Photonics & Electromagnetics Research Symposium (PIERS)* (IEEE, 2023).

[76]A. Scheinker and R. Pokharel, "Physics-constrained 3D convolutional neural networks for electrodynamics," APL Mach. Learn. **1**, 026109 (2023).

[77]S. Sarkar, A. Ji, Z. Jermain, R. Lipton, M. Brongersma, K. Dayal, and H. Y. Noh, "Physics-informed machine learning for inverse design of optical metamaterials," Adv. Photonics Res. **4**, 2300158 (2023).

[78]C. Park, S. Kim, A. W. Jung, J. Park, D. Seo, Y. Kim, C. Park, C. Y. Park, and M. S. Jang, "Sample-efficient inverse design of freeform nanophotonic devices with physics-informed reinforcement learning," Nanophotonics **13**, 1483 (2024).

[79]Y. Jeong, J. Jo, T. Lee, and J. Yoo, "Combined analysis of thermofluids and electromagnetism using physics-informed neural networks," Eng. Appl. Artif. Intell. **133**, 108216 (2024).

[80]R. Peng, J. Dong, J. Malof, W. J. Padilla, and V. Tarokh, "Deep generalized green's functions," arXiv:2306.02925 (2023).

[81]Y.-Q. Pan, R. Wang, and B.-Z. Wang, "Physics-informed neural networks with embedded analytical models: Inverse design of multilayer dielectric-loaded rectangular waveguide devices," IEEE Trans. Microwave Theory Tech. **72**, 3993 (2024).

[82]C. Mao, R. Lupoiu, T. Dai, M. Chen, and J. A. Fan, "Towards general neural surrogate solvers with specialized neural accelerators," arXiv:2405.02351 (2024).

[83]X. Liu, K. Fan, I. V. Shadrivov, and W. J. Padilla, "Experimental realization of a terahertz all-dielectric metasurface absorber," Opt. Express **25**, 191 (2017).

[84]Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE **86**, 2278–2324 (1998).

[85]A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017).

[86]S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput. **9**, 1735–1780 (1997).

[87]A. Anandkumar, K. Azizzadenesheli, K. Bhattacharya, N. Kovachki, Z. Li, B. Liu, and A. Stuart, "Neural operator: Graph kernel network for partial

[88]differential equations," in *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations* (2019).

[88]J. Jiang and J. A. Fan, "Simulator-based training of generative neural networks for the inverse design of metasurfaces," Nanophotonics **9**, 1059–1069 (2019b).

[89]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).

[90]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 meta-surface design," ACS Photonics **6**, 3196–3207 (2019).

[91]Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," Comput. Methods Appl. Mech. Eng. **360**, 112789 (2020).

[92]S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for fluid mechanics: A review," Acta Mech. Sin. **37**, 1727–1738 (2021a).

[93]S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks for heat transfer problems," J. Heat Transfer **143**, 060801 (2021b).

[94]R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox, "Projection-based model reduction: Formulations for physics-based machine learning," Comput. Fluids **179**, 704–717 (2019).

[95]P. Holl, N. Thuerey, and V. Koltun, "Learning to control PDEs with differentiable physics," in *Proceedings of the International Conference on Learning Representations* (2020).

[96]L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick, R. Ramamurthy, J. Garcke, C. Bauckhage, and J. Schuecker, "Informed machine learning—A taxonomy and survey of integrating prior knowledge into learning systems," IEEE Trans. Knowl. Data Eng. **35**, 614–633 (2021).

[97]X. Jin, S. Cai, H. Li, and G. E. Karniadakis, "NSFnets (Navier–Stokes flow nets): Physics-informed neural networks for the incompressible Navier–Stokes equations," J. Comput. Phys. **426**, 109951 (2021).

[98]J. Yu, L. Lu, X. Meng, and G. E. Karniadakis, "Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems," Comput. Methods Appl. Mech. Eng. **393**, 114823 (2022).

[99]M. Rasht-Behesht, C. Huber, K. Shukla, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for wave propagation and full waveform inversions," J. Geophys. Res. Solid Earth **127**, e2021JB023120, https://doi.org/10.1029/2021JB023120 (2022).

[100]B. F. Smith, "Domain decomposition methods for partial differential equations," in *Parallel Numerical Algorithms* (Springer, Netherlands, 1997), pp. 225–243.

[101]V. Dolean, P. Jolivet, and F. Nataf, *An Introduction to Domain Decomposition Methods* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015).

[102]J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2009).

[103]M. Kelly, R. Longjohn, and K. Nottingham, "The UCI machine learning repository," see https://archive.ics.uci.edu.

[104]M. Takamoto, T. Praditia, R. Leiteritz, D. MacKinlay, F. Alesiani, D. Pflüger, and M. Niepert, "PDEBENCH: An extensive benchmark for scientific machine learning," in *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 35, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Curran Associates, Inc., 2022), pp. 1596–1611.

[105]J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou, "Deep learning scaling is predictable, empirically," arXiv:1712.00409 [cs.LG] (2017).

[106]J. Jiang, R. Lupoiu, E. W. Wang, D. Sell, J. Paul Hugonin, P. Lalanne, and J. A. Fan, "MetaNet: A new paradigm for data sharing in photonics research," Opt. Express **28**, 13670 (2020).

[107]R. Pestourie, Y. Mroueh, T. V. Nguyen, P. Das, and S. G. Johnson, "Active learning of deep surrogates for PDEs: Application to metasurface design," npj Comput. Mater. **6**, 164 (2020).

[108]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, edited by Y. Bengio and Y. LeCun (2015).

[109]S. Ruder, "An overview of gradient descent optimization algorithms," arXiv:1609.04747 (2016).

[110]A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney, "Characterizing possible failure modes in physics-informed neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 34, edited by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (Curran Associates, Inc., 2021), pp. 26548–26560.

[111]S. Wang, X. Yu, and P. Perdikaris, "When and why PINNs fail to train: A neural tangent kernel perspective," J. Comput. Phys. **449**, 110768 (2022).

[112]Y. Shin, "On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs," Commun. Comput. Phys. **28**, 2042–2074 (2020).

[113]S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient flow pathologies in physics-informed neural networks," SIAM J. Sci. Comput. **43**, A3055–A3081 (2021).

[114]K. Shukla, A. D. Jagtap, and G. E. Karniadakis, "Parallel physics-informed neural networks via domain decomposition," J. Comput. Phys. **447**, 110683 (2021).

[115]P.-Y. Chuang and L. A. Barba, "Experience report of physics-informed neural networks in fluid simulations: Pitfalls and frustration," arXiv:2205.14249 (2022).

[116]D. Zhang, L. Lu, L. Guo, and G. E. Karniadakis, "Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems," J. Comput. Phys. **397**, 108850 (2019).

[117]A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, "Adaptive activation functions accelerate convergence in deep and physics-informed neural networks," J. Comput. Phys. **404**, 109136 (2020).

[118]A. D. Jagtap and G. Karniadakis, "Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations," Comm. Comput. Phys. **28**, 2002–2041 (2020).

[119]L. Yang, D. Zhang, and G. E. Karniadakis, "Physics-informed generative adversarial networks for stochastic differential equations," SIAM J. Sci. Comput. **42**, A292–A317 (2020).

[120]X. Meng, Z. Li, D. Zhang, and G. E. Karniadakis, "PPINN: Parareal physics-informed neural network for time-dependent PDEs," Comput. Methods Appl. Mech. Eng. **370**, 113250 (2020).

[121]G. Pang, M. D'Elia, M. Parks, and G. E. Karniadakis, "nPINNs: Nonlocal physics-informed neural networks for a parametrized nonlocal universal Laplacian operator. Algorithms and applications," J. Comput. Phys. **422**, 109760 (2020).

[122]E. Kharazmi, Z. Zhang, and G. E. Karniadakis, "hp-VPINNs: Variational physics-informed neural networks with domain decomposition," Comput. Methods Appl. Mech. Eng. **374**, 113547 (2021).

[123]L. Yang, X. Meng, and G. E. Karniadakis, "B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data," J. Comput. Phys. **425**, 109913 (2021).

[124]J. Jiang and J. A. Fan, "Multiobjective and categorical global optimization of photonic structures based on ResNet generative neural networks," Nanophotonics **10**, 361–369 (2020).

[125]M. McCabe, B. R.-S. Blancard, L. Parker, R. Ohana, M. Cranmer, A. Bietti, M. Eickenberg, S. Golkar, G. Krawezik, F. Lanusse, M. Pettee, T. Tesileanu, K. Cho, and S. Ho, "Multiple physics pretraining for physical surrogate models," in *NeurIPS 2023 AI for Science Workshop* (2023).

[126]W. J. Padilla and K. Fan, "Transparent phase dielectric metasurfaces," in *All-Dielectric Nanophotonics* (Elsevier, 2024), pp. 287–328.

[127]P. Markos and C. M. Soukoulis, *Wave Propagation* (Princeton University Press, Princeton, NJ, 2008).

[128]M. F. Yanik, S. Fan, M. Soljačić, and J. D. Joannopoulos, "All-optical transistor action with bistable switching in a photonic crystal cross-waveguide geometry," Opt. Lett. **28**, 2506 (2003).

[129]W. Suh, Z. Wang, and S. Fan, "Temporal coupled-mode theory and the presence of non-orthogonal modes in lossless multimode cavities," IEEE J. Quantum Electron. **40**, 1511–1518 (2004).

22 March 2025 09:36:26

[130]D. M. Pozar, *Microwave Engineering*, 4th ed. (John Wiley & Sons, Chichester, UK, 2011).

[131]P. Grahn, A. Shevchenko, and M. Kaivola, "Electromagnetic multipole theory for optical nanomaterials," New J. Phys. **14**, 093033 (2012).

[132]P. D. Terekhov, K. V. Baryshnikova, Y. A. Artemyev, A. Karabchevsky, A. S. Shalin, and A. B. Evlyukhin, "Multipolar response of nonspherical silicon nanoparticles in the visible and near-infrared spectral ranges," Phys. Rev. B **96**, 035443 (2017).

[133]S. Ren, A. Mahendra, O. Khatib, Y. Deng, W. J. Padilla, and J. M. Malof, "Inverse deep learning methods and benchmarks for artificial electromagnetic material design," Nanoscale **14**, 3958–3969 (2022).

[134]S. Tretyakov, A. Viitanen, S. Maslovski, and I. Saarela, "Impedance boundary conditions for regular dense arrays of dipole scatterers," IEEE Trans. Antennas Propag. **51**, 2073–2078 (2003).

[135]S. Tretyakov, S. Maslovski, and P. Belov, "An analytical model of metamaterials based on loaded wire dipoles," IEEE Trans. Antennas Propag. **51**, 2652–2658 (2003).

[136]Y. Ra'di, C. Simovski, and S. Tretyakov, "Thin perfect absorbers for electromagnetic waves: Theory, design, and realizations," Phys. Rev. Appl. **3**, 037001 (2015).

[137]D. Smith and S. Schultz, P. Markos, and C. M. Soukoulis, "Determination of effective permittivity and permeability of metamaterials from reflection and transmission coefficients," Phys. Rev. B **65**, 195104 (2002).

[138]D. R. Smith, "Analytic expressions for the constitutive parameters of magneto-electric metamaterials," Phys. Rev. E **81**, 036605 (2010).

[139]T. Christopoulos, O. Tsilipakos, and E. E. Kriezis, "Temporal coupled-mode theory in nonlinear resonant photonics: From basic principles to contemporary systems with 2D materials, dispersion, loss, and gain," J. Appl. Phys. **136**, 011101 (2024).

[140]J. R. Piper, V. Liu, and S. Fan, "Total absorption by degenerate critical coupling," Appl. Phys. Lett. **104**, 251110 (2014).

[141]X. Ming, X. Liu, L. Sun, and W. J. Padilla, "Degenerate critical coupling in all-dielectric metasurface absorbers," Opt. Express **25**, 24658–24669 (2017).

[142]C. Wang, W. R. Sweeney, A. D. Stone, and L. Yang, "Coherent perfect absorption at an exceptional point," Science **373**, 1261–1265 (2021).

[143]J. Y. Suen, K. Fan, and W. J. Padilla, "A zero-rank, maximum nullity perfect electromagnetic wave absorber," Adv. Opt. Mater. **7**, 1801632 (2019).

[144]M. Kerker, *The Scattering of Light and Other Electromagnetic Radiation: Physical Chemistry: A Series of Monographs* (Academic Press, 2013), Vol. 16.

[145]M. V. Rybin, D. S. Filonov, P. A. Belov, Y. S. Kivshar, and M. F. Limonov, "Switching from visibility to invisibility via Fano resonances: Theory and experiment," Sci. Rep. **5**, 8774 (2015).

[146]H.-T. Chen, J. Zhou, J. F. O'Hara, F. Chen, A. K. Azad, and A. J. Taylor, "Antireflection coating using metamaterials and identification of its mechanism," Phys. Rev. Lett. **105**, 073901 (2010).

[147]K. Fan, I. V. Shadrivov, A. E. Miroshnichenko, and W. J. Padilla, "Infrared all-dielectric Kerker metasurfaces," Opt. Express **29**, 10518–10526 (2021).

22 March 2025 09:36:26