# Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems

JARED WILLARD*, University of Minnesota
XIAOWEI JIA*, University of Pittsburgh
SHAOMING XU, University of Minnesota
MICHAEL STEINBACH, University of Minnesota
VIPIN KUMAR, University of Minnesota

There is a growing consensus that solutions to complex science and engineering problems require novel methodologies that are able to integrate traditional physics-based modeling approaches with state-of-the-art machine learning (ML) techniques. This paper provides a structured overview of such techniques. Application-centric objective areas for which these approaches have been applied are summarized, and then classes of methodologies used to construct physics-guided ML models and hybrid physics-ML frameworks are described. We then provide a taxonomy of these existing techniques, which uncovers knowledge gaps and potential crossovers of methods between disciplines that can serve as ideas for future research.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computing methodologies** → **Machine learning**.

Additional Key Words and Phrases: physics-guided, neural networks, deep learning, physics-informed, theory-guided, hybrid, knowledge integration

## 1 INTRODUCTION

Machine learning (ML) models, which have already found tremendous success in commercial applications, are beginning to play an important role in advancing scientific discovery in environmental and engineering domains traditionally dominated by mechanistic (e.g. first principle) models [30, 124, 128, 141, 142, 157, 232, 283]. The use of ML models is particularly promising in scientific problems involving processes that are not completely understood, or where it is computationally infeasible to run mechanistic models at desired resolutions in space and time. However, the application of even state-of-the-art black box ML models has often met with limited success in scientific domains due to their large data requirements, inability to produce physically consistent results, and their lack of generalizability to out-of-sample scenarios [47, 166, 190]. Given that neither an ML-only nor a scientific knowledge-only approach can be considered sufficient for complex scientific and engineering applications, the research community is beginning to explore the continuum between mechanistic and ML models, where both scientific knowledge and ML are integrated in a synergistic manner [139, 141, 143]. This paradigm is fundamentally different from mainstream practices in the ML community for making use of domain-specific knowledge in feature

Authors' addresses: Jared Willard, willa099@umn.edu, University of Minnesota, Minneapolis, Minnesota, 55455; Xiaowei Jia, xiaowei@pitt.edu, University of Pittsburgh, Pittsburgh, Pennsylvania, 15260; Shaoming Xu, xu000114@umn.edu, University of Minnesota, Minneapolis, Minnesota, 55455; Michael Steinbach, stei0062@umn.edu, University of Minnesota, Minneapolis, Minnesota, 55455; Vipin Kumar, kumar001@umn.edu, University of Minnesota, Minneapolis, Minnesota, 55455.

engineering or post-processing, as it is focused on approaches that integrate scientific knowledge directly into the ML framework.

Even though the idea of integrating scientific principles and ML models has picked up momentum just in the last few years, there is already a vast amount of work on this topic. For instance, in all Web of Science databases (www.webofknowledge.com) a search for "physics-informed ML" alone shows the growth of publications from 2 in 2017, 8 in 2018, 27 in 2019, to 63 in 2020. Also, many workshops and symposiums have formed to focus on this field (e.g. [1–6, 21]). This work is being pursued in diverse disciplines (e.g., earth systems [232], climate science [87, 153, 207], turbulence modeling [37, 199, 295], computational physics [261], cyberphysical systems [223], material discovery [49, 222, 243], quantum chemistry [240, 245], biological sciences [8, 215, 306], and hydrology [275, 297]), and it is being performed in the context of diverse objectives specific to these applications. Early results in isolated and relatively simple scenarios are promising, and the expectations are rising for this paradigm to accelerate scientific discovery and help address some of the biggest challenges that are facing humanity in terms of climate [87], health [280], and food security [129].

The goal of this survey is to bring these exciting developments to the ML community, to make them aware of the progress that has been made, and the gaps and opportunities that exist for advancing research in this promising direction. We hope that this survey will also be valuable for scientists who are interested in exploring the use of ML to enhance modeling in their respective disciplines. Please note that work on this topic has been referred to by other names, such as "physics-guided ML," "physics-informed ML," or "physics-aware AI" although it covers many scientific disciplines. In this survey, we also use the terms "physics-guided" or "physics," which should be more generally interpreted as "scientific knowledge-guided" or "scientific knowledge".

The focus of this survey is on approaches that integrate scientific knowledge with ML for environmental and engineering systems where scientific knowledge is available as mechanistic models, theories, and laws (e.g. conservation of mass). This distinguishes our survey from other works that focus on more general knowledge integration into machine learning [7, 277] and works covering physics integration into ML in specific domains (e.g., cyber-physical systems [223], chemistry [205], medical imaging [175], fluid mechanics [46], climate and weather[146]). This survey creates a novel taxonomy that covers a wide array of physics-ML methodologies, application-centric objectives, and general computational objectives.

We organize the paper as follows. Section 2 describes different application-centric objectives that are being pursued through combinations of scientific knowledge and ML. Section 3 discusses novel ML loss functions, model initializations, architectures, and hybrid models that researchers are developing to achieve these objectives, as well as comparisons between them. Section 4 discusses the areas of current work as well as the possibilities for cross-fertilization between methods and application-centric objectives. Then, Section 5 contains concluding remarks. Table 3 categorizes the work surveyed in this paper by application-centric objective and methods for integrating scientific knowledge in ML according to the proposed taxonomy.

## 2 APPLICATION-CENTRIC OBJECTIVES OF PHYSICS-ML INTEGRATION

This section provides a brief overview of a set of application-centric objectives where couplings of ML and scientific modeling paradigms are being pursued for applications in environmental and engineering systems. In many of these applications, scientific knowledge is represented using a mechanistic model (also referred to as physical, process-based, or first principles models). This is shown in Figure 1 as part of an abstract representation of a generic scientific problem. For example, a model for lake temperature would have drivers $\vec{x}_t$, such as amount of sunlight, air temperature, and wind speed, with static parameters $\vec{s}$, such as lake depth and water clarity, which the model $F(\vec{x}_t, \vec{s})$ would use to predict the water temperature $\vec{y}_t$ at various depths in the lake. Such physical models typically have a notion of state (which is not depicted in Figure 1 for simplicity), and complex physical models can have multiple components that model various aspects of the system, e.g., components to model clouds or ocean
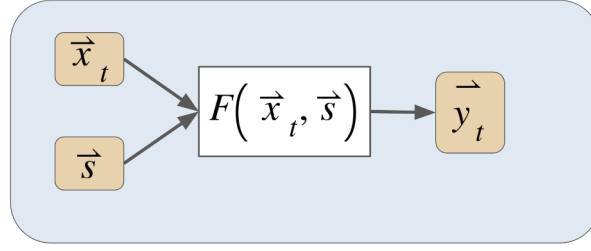
Fig. 1. A generic scientific problem in engineering, where $\vec{x}_t$ are the dynamic inputs in time, $\vec{s}$ is the set of static characteristics or parameters of the system, and $F()$ is the model producing target variable $\vec{y}_t$. $\vec{x}_t$ and $\vec{y}_t$ can also have spatial dimensions.

in a global climate model. The application-centric objectives described in this section describe different ways in which physics-ML integration can be used to address the imperfections of the mechanistic model $F()$, build a more resource efficient $F()$, or discover new knowledge such as $F()$. Below we describe how different application-centric objectives fit into each of these scenarios.

First, situations can arise in which a mechanistic model is inadequate to model a poorly understood process and a data-driven model could be leveraged to make better use of observations and possibly also process-based knowledge. Section 2.1 covers approaches like this, where physics-ML is being pursued to improve the effectiveness and predictive accuracy of $F(\vec{x}_t, \vec{s})$ with observations and scientific knowledge.

Often it is also desirable to improve the resource efficiency where physical models are too slow or at too coarse of a resolution. Section 2.2 on downscaling considers how physics-ML can produce high-resolution output variables faster than a physical model in situations spanning meteorology, climatology, and others. Similarly, if subprocesses of a larger mechanistic model are computationally intractable or inaccurate, an ML model can be used for faster or more accurate subprocess representation, as is covered in Section 2.3 on parameterization. More generally, the concept of reducing computational complexity of complex mechanistic or numerical models in the form of an ML-based or ML-assisted reduced order model is described in Section 2.4. Another objective seeking to improve resource efficiency with physics-ML is the solving of partial differential equations (PDEs) where the solution is represented with a data-driven model. This allows dynamical systems applications to bypass the often extreme computational complexity of using finite elements methods to solve complex systems of PDEs. As we can see, in many of these cases ML can be used as a more efficient *surrogate model* (also referred to as emulators) for an existing mechanistic or numerical approach.

Other objectives seek to discover new knowledge in the form of unobserved causal quantities or the symbolic representation of a process given only data. Compared to previous objectives where the goal is to produce accurate or efficient output variables, inverse modeling described in Section 2.6 flips the path shown in Figure 1 and instead seeks to find some causal static physical parameters within $\vec{s}$ given sufficient outputs. Also, Section 2.7 covers the discovering of governing equations from data, where ML has been used to go beyond traditional approaches and discover new explicit symbolic representations of phenomena.

Data generation objectives (Section 2.8) try to realistically reproduce the distribution of $\vec{x}_t$, $\vec{y}_t$, or $(\vec{x}_t, \vec{y}_t)$. Such synthetically generated data can be useful in the often data-limited situations present in engineering and environmental systems. UQ (Section 2.9) tries to learn the distribution of $\vec{y}_t$ in terms of the uncertainty of the other components of the model like the inputs, static parameters, and model state. UQ is necessary for accurate forecasting, which can also affect many of the other objectives.

Table 1 summarizes these objectives and their needs in terms of real world observations, synthetic samples (e.g. output from a mechanistic model), or knowledge of the governing equations of the system.

| Objective Name | Objective Description | Needs |
|---|---|---|
| 2.1 Improving Over Physical Model | Improved version of $F$ that better matches observations | Observations (synthetic samples can be used but not required) |
| 2.2 Downscaling | An approximate version of $F$ that provides high resolution output $Y_t$ given coarse resolution input $X_t$ | Synthetic samples at high resolution (can also make use of observations at high resolutions but this can be hard to obtain) |
| 2.3 Parameterization | Replace a component of $F$ when $F$ consists of interconnected component models | Synthetic samples (e.g. subprocess) |
| 2.4 Reduced-Order Models | Simpler version of $F$ that is more computationally efficient and possibly less accurate | Governing equations (e.g. high-fidelity, complex models) |
| 2.5 Forward Solving PDEs | Computationally efficient single pass solution of PDE | Governing equations (e.g. high-fidelity, complex models) |
| 2.6 Inverse Modeling | Discover static characteristics $S$ given $\vec{x}_t$ and $\vec{y}_t$ | Observations (synthetic samples can be used but not required) |
| 2.7 Discovering Governing Equations | Find governing equations that underlie $F$ | Observations |
| 2.8 Data Generation | Generate realistic synthetic samples without using $F$ | Synthetic data (or observations) |
| 2.9 Uncertainty Quantification | Estimate uncertainty on $\vec{y}_t$ given input $\vec{x}_t$ | Observations or synthetic samples |

Table 1. Application-centric objectives of using Physics-ML methods in terms of the generic scientific problem shown in Figure 1 and the needs to pursue each objective

## 2.1 Improving over state-of-the-art physical models

First-principle models are used extensively in a wide range of engineering and environmental applications. Even though these models are based on known physical laws, in most cases, they are necessarily approximations of reality due to incomplete knowledge of certain processes, which introduces bias. In addition, they often contain a large number of parameters whose values must be estimated with the help of limited observed data, degrading their performance further, especially due to heterogeneity in the underlying processes in both space and time. The limitations of physics-based models cut across discipline boundaries and are well known in the scientific community (e.g., see Gupta et al. [116] in the context of hydrology).

ML models have been shown to outperform physics-based models in many disciplines (e.g., materials science [148, 238, 285], applied physics [22, 127], aquatic sciences [132, 286], atmospheric science [206], biomedical science [264], computational biology [9]). A major reason for this success is that ML models (e.g., neural networks), given enough data, can find structure and patterns in problems where complexity prohibits the explicit programming of a system's exact physical nature. Given this ability to automatically extract complex relationships from data, ML models appear promising for scientific problems with physical processes that are not fully understood but have data of adequate quality and quantity is available. However, the black-box application of ML has met with limited

success in scientific domains due to a number of reasons [141]: (i) while state-of-the-art ML models are capable of capturing complex spatiotemporal relationships, they require far too much labeled data for training, which is rarely available in real application settings, (ii) ML models often produce scientifically inconsistent results; and (iii) ML models can only capture relationships in the available training data, and thus cannot generalize to out-of-sample scenarios (i.e., those not represented in the training data).

The key objective here is to combine elements of physics-based modeling with state-of-the-art ML models to leverage their complementary strengths. Such integrated physics-ML models are expected to better capture the dynamics of scientific systems and advance the understanding of underlying physical processes. An early attempt for combining ML with physics-based modeling in lake temperature dynamics [231]) has already demonstrated its potential for providing better prediction accuracy with a much smaller number of samples as well as generalizability in out-of-sample scenarios.

## 2.2 Downscaling

Complex mechanistic models are capable of capturing physical reality more precisely than simpler models, as they often involve more diverse components that account for a greater number of processes at finer spatial or temporal resolution. However, given the computation cost and modeling complexity, many models are run at a resolution that is coarser than what is required to precisely capture underlying physical processes. For example, cloud-resolving models (CRM) need to run at sub-kilometer horizontal resolution to be able to effectively represent boundary-layer eddies and low clouds, which are crucial for the modeling of Earth's energy balance and the cloud–radiation feedback [230]. However, it is not feasible to run global climate models at such fine resolutions even with the most powerful computers expected to be available in the near future.

Downscaling techniques have been widely used as a solution to capture physical variables that need to be modeled at a finer resolution. In general, the downscaling techniques can be divided into two categories: statistical downscaling and dynamical downscaling. Statistical downscaling refers to the use of empirical models to predict finer-resolution variables from coarser-resolution variables. Such a mapping across different resolutions can involve complex non-linear relationships that cannot be precisely modeled by traditional empirical models. Recently, artificial neural networks have shown a lot of promise for this problem, given their ability to model non-linear relationships [249, 273]. In contrast, dynamical downscaling makes use of high-resolution regional simulations to dynamically simulate relevant physical processes at regional or local scales of interest. Due to the substantial time cost of running such complex models, there is an increasing interest in using ML models as surrogate models (a model approximating simulation-driven input–output data) to predict target variables at a higher resolution [103, 254].

Although the state-of-the-art ML methods can be used in both statistical and dynamical downscaling, it remains a challenge to ensure that the learned ML component is consistent with established physical laws and can improve the overall simulation performance.

## 2.3 Parameterization

Complex physics-based models (e.g., for simulating phenomena in climate, weather, turbulence modeling, hydrology) often use an approach known as *parameterization* to account for missing physics. In parameterization (note that this term has a different meaning when used in mathematics and geometry), specific complex dynamical processes are replaced by simplified physical approximations that are represented as static parameters. A common way to estimate the value of these parameters is to use grid search over the space of combinations of parameter values that lead to the best match with observations. This procedure is referred to as parameter calibration. The failure to correctly parameterize can make the model less robust, and errors that result from imperfect parameterization can also feed into other components of the entire physics-based model and deteriorate the modeling

of important physical processes. Another way, that is being considered increasingly, is to replace processes that are too complex to be physically represented in the model by a simplified dynamic or statistical/ML process. This makes it possible to learn new parameterizations directly from observations and/or high-resolution model simulation using ML methods. Already, ML-based parameterizations have shown success in geology [52, 109], atmospheric science [40, 103], and hydrology [29]. A major benefit of ML-based parameterizations is the reduction of computation time compared to traditional physics-based simulations. In chemistry, Hansen et al. [120] find that parameterizations of atomic energy using ML take seconds compared to multiple days for more standard quantum-calculation calculations, and Behler et al. [27] find that neural networks can vastly improve the efficiency of finding potential energy surfaces of molecules.

Most of the existing work uses standard black box ML models for parameterization, but there is an increasing interest in integrating physics in the ML models [32], as it has the potential to make them more robust and generalizable to unseen scenarios as well as reduce the number of training samples needed for training.

## 2.4 Reduced-Order Models

Reduced-Order Models (ROMs) are computationally inexpensive representations of more complex models. Usually, constructing ROMs involves dimensionality reduction that attempts to capture the most important dynamical characteristics of often large, high-fidelity simulations and models of physical systems (e.g., in fluid dynamics [164]). This can also be viewed as a controlled loss of accuracy. A common way to do this is to project the governing equations of a system onto a linear subspace of the original state space using a method such as principal components analysis or dynamic mode decomposition [221]. However, limiting the dynamics to a lower-dimensional subspace inherently limits the accuracy of any ROM.

ML is beginning to assist in constructing ROMs for increased accuracy and reduced computational cost in several ways. One approach is to build an ML-based surrogate model for full-order models [56, 147], where the ML model can be considered a ROM. Other ways include building an ML-based surrogate model of an already built ROM by another dimensionality reduction method [295] or building an ML model to mimic the dimensionality reduction mapping from a full-order model to a reduced-order model [199]. ML and ROMs can also be combined by using the ML model to learn the residual between a ROM and observational data [278]. ML models have the potential to greatly augment the capabilities of ROMs because of their typically quick forward execution speed and ability to leverage data to model high dimensional phenomena.

One area of recent focus of ML-based ROMs is in approximating the dominant modes of the Koopman (or composition) operator, as a method of dimensionality reduction. The Koopman operator is an infinite-dimension linear operator that encodes the temporal evolution of the system state through nonlinear dynamics [42]. This allows linear analysis methods to be applied to nonlinear systems and enables the inference of properties of dynamical systems that are too complex to express using traditional analysis techniques. Applications span many disciplines, including fluid dynamics [250], oceanography [107], molecular dynamics [289], and many others. Though dynamic mode decomposition [197] is the most common technique for approximating the Koopman operator, many recent approaches have been made to approximate Koopman operator embeddings with deep learning models that outperform existing methods [170, 185, 191, 200, 209, 210, 260, 284, 307]. Adding physics-based knowledge to the learning of the Koopman operator has the potential to augment generalizability and interpretability, which current ML methods in this area tend to lack [210].

Traditional methods for ROMs often lack robustness with respect to parameter changes within the systems they are representing [11], or are not cost-effective enough when trying to predict complex dynamical systems (e.g., multiscale in space and time). Thus, incorporating principles from physics-based models could potentially reduce the search space to enable more robust training of ROMs, and also allow the model to be trained with less data in many scenarios.

## 2.5 Forward Solving Partial Differential Equations

In many physical systems, governing equations are known, but direct numerical solutions of partial differential equations (PDEs) using common methods, such as the Finite Elements Method or the Finite Difference Method [93], are prohibitively expensive. In such cases, traditional methods are not ideal or sometimes even possible. A common technique is to use an ML model as a surrogate for the solution to reduce computation time [75, 159]. In particular, NN solvers can reduce the high computational demands of traditional numerical methods into a single forward-pass of a NN. Notably, solutions obtained via NNs are also naturally differentiable and have a closed analytic form that can be transferred to any subsequent calculations, a feature not found in more traditional solving methods [159]. Especially with the recent advancement of computational power, neural network models have shown success in approximating solutions across different kinds of physics-based PDEs [15, 149, 233], including the difficult quantum many-body problem [51] and many-electron Schrodinger equation [119]. As a step further, deep neural networks models have shown success in approximating solutions across high dimensional physics-based PDEs previously considered unsuitable for approximation by ML [118, 252]. However, slow convergence in training, limited applicability to many complex systems, and reduced accuracy due to unawareness of physical laws can prove problematic. More recently, Li et al. [171] have defined a neural Fourier operator which allows a neural network to learn and solve an entire family of PDEs by learning the mapping from any functional parametric dependence to the solution in Fourier space.

## 2.6 Inverse Modeling

The forward modeling of a physical system uses the physical parameters of the system (e.g., mass, temperature, charge, physical dimensions or structure) to predict the next state of the system or its effects (outputs). In contrast, inverse modeling uses the (possibly noisy) output of a system to infer the intrinsic physical parameters or inputs. Inverse problems often stand out as important in physics-based modeling communities because they can potentially shed light on valuable information that cannot be observed directly. One example is the use of x-ray images from a CT scan to create a 3D image reflecting the structure of part of a person's body [183]. This can be viewed as a computer vision problem, where, given many training datasets of x-ray scans of the body at different capture angles, a model can be trained to inversely reconstruct textures and 3D shapes of different organs or other areas of interest. Allowing for better reconstruction while reducing scan time could potentially increase patient satisfaction and reduce overall medical costs. Though there are many inverse modeling scenarios, in this work we focus on intrinsic physical parameters found in a mechanistic modeling scenario for engineering and environmental systems.

Often, the solution of an inverse problem can be computationally expensive due to the potentially millions of forward model evaluations needed for estimator evaluation or characterization of posterior distributions of physical parameters [96]. ML-based surrogate models (in addition to other methods such as reduced-order models) are becoming a realistic choice since they can model high-dimensional phenomena with lots of data and execute much faster than most physical simulations.

Inverse problems are traditionally solved using regularized regression techniques. Data-driven methods have seen success in inverse problems in remote sensing of surface properties [69], hydrology [106], photonics [218], and medical imaging [183], among many others. Recently, novel algorithms using deep learning and neural networks have been applied to inverse problems. While still in their infancy, these techniques exhibit strong performance for applications such as computerized tomography [57, 193, 263], seismic processing [272], or various sparse data problems.

There is also increasing interest in the inverse design of materials using ML, where desired target properties of materials are used as input to the model to identify atomic or microscale structures that exhibit such properties [156, 172, 222, 243]. Physics-based constraints and stopping conditions based on material properties can be used to guide the optimization process [172, 262]. These constraints and similar physics-guided techniques have the

potential to alleviate noted challenges in inverse modeling, particularly in scenarios with a small sample size and a paucity of ground-truth labels [142]. The integration of prior physical knowledge is common in current approaches to the inverse problem, and its integration into ML-based inverse models has the potential to improve data efficiency and increase its ability to solve ill-posed inverse problems.

## 2.7 Discovering Governing Equations

When the governing equations of a dynamical system are known explicitly, they allow for more robust forecasting, control, and the opportunity for analysis of system stability and bifurcations through increased interpretability [235]. Furthermore, if a mathematical model accurately describes the processes governing the observed data, it therefore can generalize to data outside of the training domain. However, in many disciplines (e.g., neuroscience, cell biology, ecology, epidemiology) dynamical systems have no formal analytic descriptions. Often in these cases, data is abundant, but the underlying governing equations remain elusive. In this section, we discuss equation discovery systems that do not assume the structure of the desired equation (as in Section 2.5), but rather explore a large space of possibly nonlinear mathematical terms.

Advances in ML for the discovery of these governing equations has become an active research area with rich potential to integrate principles from applied mathematics and physics with modern ML methods. Early works on the data-driven discovery of physical laws relied on heuristics and expert guidance and were focused on rediscovering known, non-differential, laws in different scientific disciplines from artificial data [105, 162, 163, 168]. This was later expanded to include real-world data and differential equations in ecological applications [83]. Recently, general and robust data-driven discovery of potentially unknown governing equations has been pioneered by [38, 244], where they apply symbolic regression to differences between computed derivatives and analytic derivatives to determine underlying dynamical systems. More recently, works have used sparse regression built on a dictionary of functions and partial derivatives to construct governing equations [43, 220, 234]. Lagergren et al. [160] expand on this by using ANNs to construct the dictionary of functions. These sparse identification techniques are based on the principle of Occam's Razor, where the goal is that only a few equation terms be used to describe any given nonlinear system.

## 2.8 Data Generation

Data generation approaches are useful for creating virtual simulations of scientific data under specific conditions. For example, these techniques can be used to generate potential chemical compounds with desired characteristics (e.g., serving as catalysts or having a specific crystal structure). Traditional physics-based approaches for generating data often rely on running physical simulations or conducting physical experiments, which tend to be very time consuming. Also, these approaches are restricted by what can be produced by physics-based models. Hence, there is an increasing interest in generative ML approaches that learn data distributions in unsupervised settings and thus have the potential to generate novel data beyond what could be produced by traditional approaches. Generative ML models have found tremendous success in areas such as speech recognition and generation [208], image generation [73], and natural language processing [114]. These models have been at the forefront of unsupervised learning in recent years, mostly due to their efficiency in understanding unlabeled data. The idea behind generative models is to capture the inner probabilistic distribution in order to generate similar data. With the recent advances in deep learning, new generative models, such as the generative adversarial network (GAN) and variational autoencoder (VAE), have been developed. These models have shown much better performance in learning non-linear relationships to extract representative latent embeddings from observation data. Hence the data generated from the latent embeddings are more similar to true data distribution. In particular, the adversarial component of GAN consists of a two-part framework with a generative network and discriminative network, where the generative

network's objective is to generate fake data to fool the discriminative network, while the discriminative network attempts to determine true data from fake data.

In the scientific domain, GANs can generate data like the data generated by the physics-based models. Using GANs often incurs certain benefits, including reduced computation time and a better reproduction of complex phenomenon, given the ability of GANs to represent nonlinear relationships. For example, Farimani et al. [91] have shown that Conditional Generative Adversarial Networks (cGAN) can be trained to simulate heat conduction and fluid flow purely based on observations without using knowledge of the underlying governing equations. Such approaches that use generative models have been shown to significantly accelerate the process of generating new data samples.

However, a well-known issue of GANs is that they incur dramatically high sample complexity. Therefore, a growing area of research is to engineer GANs that can leverage prior knowledge of physics in terms of physical laws and invariance properties. For example, GAN-based models for simulating turbulent flows can be further improved by incorporating physical constraints, e.g., conservation laws [308] and energy spectrum [291], in the loss function. Cang et al. [49] also imposed a physics-based morphology constraint on a VAE-based generative model used for simulating artificial material samples. The physics-based constraint forces the generated artificial samples to have the same morphology distribution as the authentic ones and thus greatly reduces the large material design space.

## 2.9 Uncertainty Quantification

Uncertainty quantification (UQ) is of great importance in many areas of computational science (e.g., climate modeling [74], fluid flow [65], systems engineering [217], among many others). At its core, UQ requires an accurate characterization of the entire distribution $p(y|x)$, where $y$ is the response and $x$ is the covariate of interest, rather than just making a point prediction $y = f(x)$. This makes it possible to characterize all quantiles and skews in the distribution, which allows for analysis such as examining how close predictions are to being unacceptable, or sensitivity analysis of input features.

Applying UQ tasks to physics-based models using traditional methods such as Monte Carlo (MC) is usually infeasible due to the very large number of forward model evaluations needed to obtain convergent statistics. In the physics-based modeling community, a common technique is to perform model reduction (described in Section 2.4) or create an ML surrogate model, in order to increase model evaluation speed since ML models often execute much faster [99, 189, 269]. With a similar goal, the ML community has often employed Gaussian Processes as the main technique for quantifying uncertainty in simulating physical processes [34, 229], but neither Gaussian Processes nor reduced models scale well to higher dimensions or larger datasets (Gaussian Processes scale as $O(N^3)$ with $N$ data points).

Consequently, there is an effort to fit deep learning models, which have exhibited countless successes across disciplines, as a surrogate for numerical simulations in order to achieve faster model evaluations for UQ that have greater scalability than Gaussian Processes [269]. However, since artificial neural networks do not have UQ naturally built into them, variations have been developed. One such modification uses a probabilistic drop-out strategy in which neurons are periodically "turned off" as a type of Bayesian approximation to estimate uncertainty [98]. There are also Bayesian variants of neural networks that consist of distributions of weights and biases [186, 314, 319], but these suffer from high computation times and high dependence on reliable priors. Another method uses an ensemble of neural networks to create a distribution from which uncertainty is quantified [161].

The integration of physics knowledge into ML for UQ has the potential to allow for a better characterization of uncertainty. For example, ML surrogate models run the risk of producing physically inconsistent predictions, and incorporating elements of physics could help with this issue. Also, note that the reduced data needs of ML due to

constraints for adherence to known physical laws could alleviate some of the high computational cost of Bayesian neural networks for UQ.

## 3 PHYSICS-ML METHODS

Given the diversity of forms in which scientific knowledge is represented in different disciplines and applications, researchers have developed a large variety of methods for integrating physical principles into ML models. This section categorizes them into the following four classes; (i) physics-guided loss function, (ii) physics-guided initialization, (iii) physics-guided design of architecture, and (iv) hybrid modeling.

Choosing between different classes of methods for a given problem can depend on many factors including the availability and performance of existing mechanistic models, and also the general computational objectives that need to be addressed. The general computational objectives of physics-ML methods described throughout this section, as opposed to traditional ML methods, can be placed into three categories. First, *prediction performance* defined as better matching between predicted and observed values can be improved in a variety of ways including improved generalizability to out-of-sample scenarios, improved general accuracy, or forcing solutions to be physically consistent (e.g. obeying known physics-based governing equations). Second, *sample efficiency* can be improved by reducing the number of observations required for adequate performance or reducing the overall search space. The third general computational objective is *interpretability*, where often traditional ML models are a "black box" and the incorporation of scientific knowledge can shine light on physical meanings, interpretations, and processes within the ML framework. Even though computational objectives can be categorized within these categories, there is also overlap between them. For example, forcing models to be physically consistent can effectively reduce the solution search space. Improved sample efficiency can also lead to improved prediction performance by getting more value out of each observation. We end this section with a summary and detailed discussion comparing different kinds of methods, their requirements, and the general computational objectives achieved.

### 3.1 Physics-Guided Loss Function

Scientific problems often exhibit a high degree of complexity due to relationships between many physical variables varying across space and time at different scales. Standard ML models can fail to capture such relationships directly from data, especially when provided with limited observation data. This is one reason for their failure to generalize to scenarios not encountered in training data. Researchers are beginning to incorporate physical knowledge into loss functions to help ML models capture generalizable dynamic patterns consistent with established physical laws.

One of the most common techniques to make ML models consistent with physical laws is to incorporate physical constraints into the loss function of ML models as follows [141],

$$\text{Loss} = \text{Loss}_{\text{TRN}}(Y_{\text{true}}, Y_{\text{pred}}) + \lambda R(W) + \gamma \text{Loss}_{\text{PHY}}(Y_{\text{pred}}) \tag{1}$$

where the training loss $\text{Loss}_{\text{TRN}}$ measures a supervised error (e.g., RMSE or cross-entropy) between true labels $Y_{\text{true}}$ and predicted labels $Y_{\text{pred}}$, and $\lambda$ is a hyper-parameter to control the weight of model complexity loss $R(W)$. The first two terms are the standard loss of ML models. The addition of physics-based loss $\text{Loss}_{\text{PHY}}$ aims to ensure consistency with physical laws and it is weighted by a hyper-parameter $\gamma$, where $\gamma$ is determined alongside other ML hyperparameters using validation data or a nested cross validation setup. A comprehensive guide to implementing physics-based loss functions can be found in Ebert-Uphoff et al. [84].

Steering ML predictions towards physically consistent outputs has numerous benefits. First, this provides the possibility to ensure the consistency with physical laws and therefore reduce the solution search space of ML models. Second, the regularization by physical constraints allows the model to learn even with unlabeled data, as the computation of physics-based loss $\text{Loss}_{\text{PHY}}$ does not require observation data. Third, ML models which follow desired physical properties are more likely to be generalizable to out-of-sample scenarios relative to basic ML models[133, 231]. It is important to note, however, that the physics-guided loss function does not "guarantee" either

**Loss Function:**
$$\text{Loss}_{\text{TRN}}(y_{\text{PRED}}, y_{\text{TRUE}}) + \lambda * R(w) + \gamma * \text{Loss}_{\text{PHYS}}(y_{\text{PRED}}, U)$$
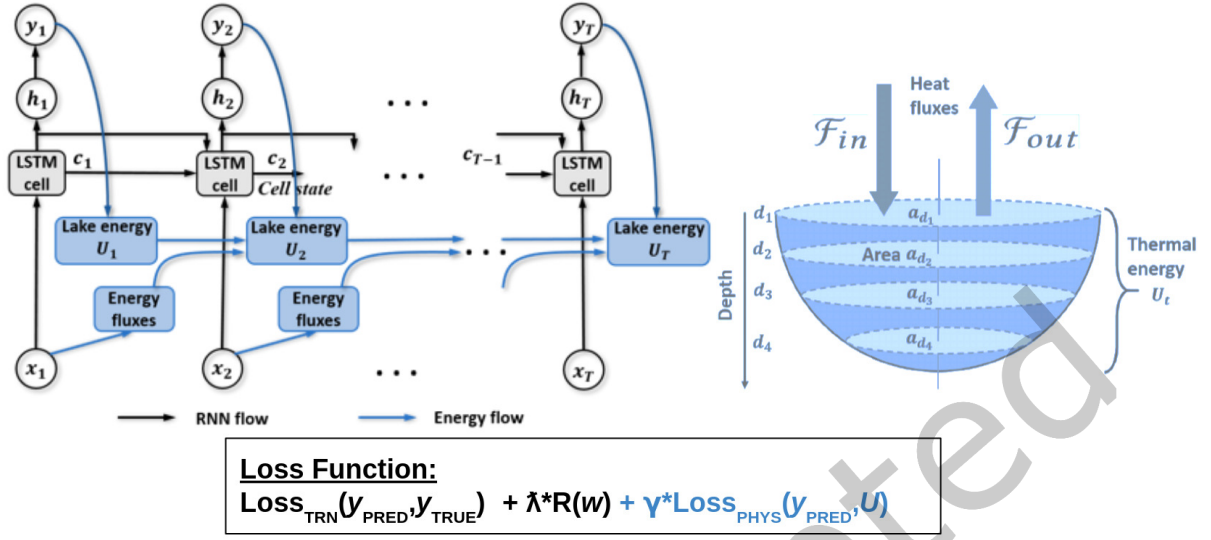
Fig. 2. The Physics-Guided Recurrent Neural Network (PGRNN) model demonstrated in Jia et al. [132] is an example of a physics-guided loss function allowing physical knowledge to be incorporated into the ML model. They include the standard RNN flow (black arrows) and the energy flow (blue arrows) in the recurrent process. Here $U_T$ represents the thermal energy of the lake at time $T$, and both the energy output and temperature output $y_T$ are used in calculating the loss function value. This enables the PGRNN to predict lake temperature without violating energy constraints. A detailed description of the loss function equation (Equation 1) can be found in Section 3.1.

physical consistency or generalizability as it is fundamentally a weak constraint. Loss function terms corresponding to physical constraints are applicable across many different types of ML frameworks. In addition, this method is extensively used across all application-centric objectives listed in Section 2. In the following paragraphs, we demonstrate the use of physics-based loss functions for different objectives described in Section 2.

*Replacing or improving over physical models.* Incorporation of physics-based loss has shown great success in improving prediction ability of ML models. In the context of lake temperature modeling, Karpatne et al. [140] includes a physics-based penalty that ensures that predictions of denser water are at lower depths than predictions of less dense water, a known monotonic relationship.

Jia et al. [130] and Read et al. [231] further extended this work to capture even more complex and general physical relationships that happen on a temporal scale. Specifically, they use a physics-based penalty for energy conservation in the loss function to ensure the lake thermal energy gain across time is consistent with the net thermodynamic fluxes in and out of the lake. A diagram of this model is shown in Figure 2. Note that the recurrent structure contains additional nodes (shown in blue) to represent physical variables (lake energy, etc) that are computed using purely physics-based equations. These are needed to incorporate energy conservation in the loss function. Similar structure can be used to model other physical laws such as mass conservation, etc. Qualitative mathematical properties of dynamical systems modeling have also shown promise in informing loss functions to improve prediction beyond that of the physics model. Erichson et al. [86] penalize autoencoders based on physically meaningful stability measures in dynamical systems to improve prediction of fluid flow and sea surface temperature. They showed an improved mapping of past states to future states for both modeling scenarios in addition to improving generalizability to new data.

*Solving PDEs.* Another strand of work that involves loss function alterations is solving PDEs for dynamical systems modeling, in which adherence to the governing equations is enforced in the loss function. In Raissi et al. [228], this concept is developed and shown to create data-efficient spatiotemporal function approximators to both solve and find parameters of basic PDEs like Burgers Equation or Schrodinger Equation. Going beyond a simple feed-forward network, Zhu et al. [318] propose an encoder-decoder model for predicting transient PDEs with governing PDE constraints. Geneva et al. [102] extended this approach to deep auto-regressive dense encoder-decoders with a Bayesian framework using stochastic weight averaging to quantify uncertainty.

*Discovering Governing Equations.* Physics-based loss function terms have also been used in the discovery of governing equations. Loiseau et al. [178] used constrained least squares [110] to incorporate energy-preserving nonlinearities or to enforce symmetries in the identified equations for the equation learning process described in Section 2.7. Though these loss functions are mostly seen in common variants of NNs, they are also be seen in architectures such as echo state networks. Doan et al. [76] found that integrating the physics-based loss from the governing equations in a Lorenz system, a commonly studied system in dynamical systems, strongly improves the echo state network's time-accurate prediction of the system and also reduces convergence time.

*Inverse modeling.* For applications in vortex induced vibrations, Raissi et al. [224] pose the inverse modeling problem of predicting the lift and drag forces of a system given sparse data about its velocity field. Kahana et al. [136] uses a loss function term pertaining to the physical consistency of the time evolution of waves for the inverse problem of identifying the location of an underwater obstacle from acoustic measurements. In both cases, the addition of physics-based loss terms made results more accurate and more robust to out-of-sample scenarios.

*Parameterization.* While ML has been used for parameterization, adding physics-based loss terms can further benefit this process by ensuring physically consistent outputs. Zhang et al. [310] parameterize atomic energy for molecular dynamics using a NN with a loss function that takes into account atomic force, atomic energy, and terms relating kinetic and potential energy. Furthermore, in climate modeling, Beucler *et al.* show that enforcing energy conservation laws improves prediction when emulating cloud processes [31, 32].

*Downscaling.* Super resolution and downscaling frameworks have also begun to incorporate physics-based constraints. Jiang et al. [134] use PDE-based constraints for super resolution problems in computational fluid dynamics where they are able to more efficiently recover physical quantities of interest. Bode et al. [37] use similar constraint ideas in building generative adversarial networks for super resolution in turbulence modeling in combustion scenarios, where they find improved generalization capability and extrapolation due to the constraints.

*Uncertainty quantification.* In Yang et al. [303] and Yang et al. [304], physics-based loss is implemented in a deep probabilistic generative model for uncertainty quantification based on adherence to the structure imposed by PDEs. To accomplish this, they construct probabilistic representations of the system states, and use an adversarial inference procedure to train using a physics-based loss function that enforces adherence to the governing laws. This is expanded in Zhu et al. [318], where a physics-informed encoder-decoder network is defined in conjunction with a conditional flow-based generative model for similar purposes. A similar loss function modification is performed in other works [102, 144, 299], but for the purpose of solving high dimensional stochastic PDEs with uncertainty propagation. In these cases, physics-guided constraints provide effective regularization for training deep generative models to serve as surrogates of physical systems where the cost of acquiring data is high and the data sets are small [304].

Another direction for encoding physics knowledge into ML UQ applications is to create a physics-guided Bayesian NN. This is explored in Yang et al. [300], where they use a Bayesian NN, which naturally encodes uncertainty, as a surrogate for a PDE solution. Additionally, they add a PDE constraint for the governing laws of the

system to serve as a prior for the Bayesian net, allowing for more accurate predictions in situations with significant noise due to the physics-based regularization.

*Generative models.* In recent years, GANs have been used to efficiently generate solutions to PDEs and there is interest in using physics knowledge to improve them. Yang et al. [301] showed GANs with loss functions based on PDEs can be used to solve stochastic elliptic PDEs in up to 30 dimensions. In a similar vein, Wu et al. [292] showed that physics-based loss functions in GANs can lower the amount of data and training time needed to converge on solutions of turbulence PDEs, while Shah et al. [248] saw similar results in the generation of microstructures satisfying certain physical properties in computational materials science.

## 3.2 Physics-Guided Initialization

Since many ML models require an initial choice of model parameters before training, researchers have explored different ways to physically inform a model starting state. For example, in NNs, weights are often initialized according to a random distribution prior to training. Poor initialization can cause models to anchor in local minima, which is especially true for deep neural networks. However, if physical or other contextual knowledge is used to help inform the initialization of the weights, model training can be accelerated and may require fewer training samples [132]. One way to inform the initialization to assist in model training and escaping local minima is to use an ML technique known as *transfer learning*. In transfer learning, a model is *pre-trained* on a related task prior to being fine-tuned with limited training data to fit the desired task. The pre-trained model serves as an informed initial state that ideally is closer to the desired parameters for the desired task than random initialization. One way to achieve this is to use the physics-based model's simulated data to pre-train the ML model. This is similar to the common application of pre-training in computer vision, where CNNs are often pre-trained with very large image datasets before being fine-tuned on images from the task at hand [259].

Jia *et al.* use this strategy in the context of modeling lake temperature dynamics [130, 132]. They pre-train their Physics-Guided Recurrent Neural Network (PGRNN) models for lake temperature modeling on simulated data generated from a physics-based model and fine tune the NN with little observed data. They showed that pre-training, even using data from a physical model with an incorrect set of parameters, can still significantly reduce the training data needed for a quality model. In addition, Read et al. [231] demonstrated that models using both physics-guided initialization and a physics-guided loss function are able to generalize better to unseen scenarios than traditional physics-based models. In this case, physics-guided initialization allows the model to have a physically-consistent starting state prior to seeing any observations.

Another application can be seen in robotics, where images from robotics simulations have been shown to be sufficient without any real-world data for the task of object localization [267], while reducing data requirements by a factor of 50 for object grasping [39]. Then, in autonomous vehicle training, Shah et al. [247] showed that pre-training the driving algorithm in a simulator built on a video game physics engine can drastically lessen data needs. More generally, we see that simulation-based pre-training of applications allows for significantly less expensive data collection than is possible with physical robots.

Physics-guided model initialization has also been employed in chemical process modeling [180, 181, 298]. Yan et al. [298] use Gaussian process regression for process modeling that has been transferred and adapted from a similar task. To adapt the transferred model, they used scale-bias correcting functions optimized through maximum likelihood estimation of parameters. Furthermore, Gaussian process models come equipped with uncertainty quantification which is also informed through initialization. A similar transfer and adapt approach is seen in Lu et al. [180], but for an ensemble of NNs transferred from related tasks. In both studies, the similarity metrics used to find similar systems are defined by considering various common process characteristics and behaviors.

Physics-guided initialization can also be done using a self-supervised learning method, which has been widely used in computer vision and natural language processing. In the self-supervised setting, deep neural networks

learn discriminative representations using pseudo labels created from pre-defined pretext tasks. These pretext tasks are designed to extract complex patterns related to our target prediction task. For example, the pretext task can be defined to predict intermediate physical variables that play an important role in underlying processes. This approach can make use of a physics-based model to simulate these intermediate physical variables, which can then be used to pre-train ML models by adding supervision on hidden layers. As an illustration of this approach, Jia et al. [133] have shown promising results for modeling temperature and flow in river networks by using upstream water variables simulated by a physics-based PRMS-SNTemp model [265] to pre-train hidden variables in a graph neural network.

## 3.3 Physics-Guided Design of Architecture

Although the physics-based loss and initialization in the previous sections help constrain the search space of ML models during training, the ML architecture is often still a black box. In particular, they do not encode physical consistency or other desired physical properties into the ML architecture. A recent research direction has been to construct new ML architectures that can make use of the specific characteristics of the problem being solved. Furthermore, incorporating physics-based guidance into architecture design has the added bonus of making the previously black box algorithm more interpretable, a desirable but typically missing feature of ML models used in physical modeling. In the following paragraphs, we discuss several contexts in which physics-guided ML architectures have been used. Much of the work in this section is focused largely on neural network architectures. The modular and flexible nature of NNs in particular makes them prime candidates for architecture modification. For example, domain knowledge can be used to specify node connections that capture physics-based dependencies among variables. We also include subsections on multi-task learning and structures of Gaussian processes to show how task interrelationships or informed prior distributions can inform ML models.

*Intermediate Physical Variables.* One way to embed physical principles into NN design is to ascribe physical meaning for certain neurons in the NN. It is also possible to declare physically relevant variables explicitly. In lake temperature modeling, Daw et al. [68] incorporate a physical intermediate variable as part of a monotonicity-preserving structure in the LSTM architecture. This model produces physically consistent predictions in addition to appending a dropout layer to quantify uncertainty. Muralidlar et al. [204] used a similar approach to insert physics-constrained variables as the intermediate variables in the convolutional neural network (CNN) architecture which achieved significant improvement over state-of-the-art physics-based models on the problem of predicting drag force on particle suspensions in moving fluids.

An additional benefit of adding physically relevant intermediate variables in an ML architecture is that they can help extract physically meaningful hidden representation that can be interpreted by domain scientists. This is particularly valuable, as standard deep learning models are limited in their interpretability since they can only extract abstract hidden variables using highly complex connected structure. This is further exacerbated given the randomness involved in the optimization process.

Another related approach is to fix one or more weights within the NN to physically meaningful values or parameters and make them non-modifiable during training. A recent approach is seen in geophysics where researchers use NNs for the waveform inversion modeling to find subsurface parameters from seismic wave data. In Sun et al. [256], they assign most of the parameters within a network to mimic seismic wave propagation during forward propagation of the NN, with weights corresponding to values in known governing equations. They show this leads to more robust training in addition a more interpretable NN with meaningful intermediate variables.

*Encoding invariances and symmetries.* In physics, there is a deep connection between symmetries and invariant quantities of a system and its dynamics. For example, Noether's Law, a common paradigm in physics, demonstrates a mapping between conserved quantities of a system and the system's symmetries (e.g. translational symmetry

can be shown to correspond to the conservation of momentum within a system). Therefore, if an ML model is created that is translation-invariant, the conservation of momentum becomes more likely and the model's prediction becomes more robust and generalizable.

State-of-the-art deep learning architectures already encode certain types of invariance. For example, RNNs encode temporal invariance and CNNs can implicitly encode spatial translation, rotation, and scale invariance. In the same way, scientific modeling tasks may require other invariances based on physical laws. In turbulence modeling and fluid dynamics, Ling et al [173] define a *tensor basis neural network* to embed rotational invariance into a NN for improved prediction accuracy. This solves a key problem in ML models for turbulence modeling because without rotational invariance, the model evaluated on identical flows with axes defined in other directions could yield different predictions. This work alters the NN architecture by adding a higher-order multiplicative layer that ensures the predictions lie on a rotationally invariant tensor basis. In a molecular dynamics application, Anderson et al. [12] show that a rotationally covariant NN architecture can learn the behavior and properties of complex many-body physical systems.

In a general setting, Wang et al. [281] show how spatiotemporal models can be made more generalizable by incorporating symmetries into deep NNs. More specifically, they demonstrated the encoding of translational symmetries, rotational symmetries, scale invariances, and uniform motion into NNs using customized convolutional layers in CNNs that enforce desired invariance properties. They also provided theoretical guarantees of invariance properties across the different designs and showed additional to significant increases in generalization performance.

Incorporating symmetries, by informing the structure of the solution space, also has the potential to reduce the search space of an ML algorithm. This is important in the application of discovering governing equations, where the space of mathematical terms and operators is exponentially large. Though in its infancy, physics-informed architectures for discovering governing equations are beginning to be investigated by researchers. In Section 2.7, symbolic regression is mentioned as an approach that has shown success. However, given the massive search space of mathematical operators, analytic functions, constants, and state variables, the problem can quickly become NP-hard. Udrescu et al. [270] designs a recursive multidimensional version of symbolic regression that uses a NN in conjunction with techniques from physics to narrow the search space. Their idea is to use NNs to discover hidden signs of "simplicity", such as symmetry or separability in the training data, which enables breaking the massive search space into smaller ones with fewer variables to be determined.

In the context of molecular dynamics applications, a number of researchers [28, 310] have used a NN per individual atom to calculate each atom's contribution to the total energy. Then, to ensure the energy invariance with respect to the possibility of interchanging two atoms, the structure of each NN and the values of each network's weight parameters are constrained to be identical for atoms of the same element. More recently, novel deep learning architectures have been proposed for fundamental invariances in chemistry. Schutt et al. [245] proposes continuous-filter convolutional (cfconv) layers for CNNs to allow for modeling objects with arbitrary positions such as atoms in molecules, in contrast to objects described by Cartesian-gridded data such as images. Furthermore, their architecture uses atom-wise layers that incorporate inter-atomic distances that enabled the model to respect quantum-chemical constraints such as rotationally invariant energy predictions as well as energy-conserving force predictions. As we can see, because molecular dynamics often ascribes importance to different important geometric properties of molecules (e.g. rotation), network architectures dealing with invariances can be effective for improving performance and robustness of ML models.

Architecture modifications incorporating symmetry are also seen extensively in dynamical systems research involving differential equations. In a pioneering work by Ruthotto et al [236], three variations of CNNs are proposed to improve classifiers for images. Each variation uses mathematical theories to guide the design of the CNN based on fundamental properties of PDEs. Multiple types of modifications are made, including adding symmetry layers to guarantee the stability expressed by the PDEs and layers that convert inputs to kinematic eigenvalues that satisfy certain physical properties. They define a parabolic CNN inspired by anisotropic filtering, a hyperbolic CNN based

on Hamiltonian systems, and a second order hyperbolic CNN. Hyperbolic CNNs were found to preserve the energy in the system as intended, which set them apart from parabolic CNNs that smooth the output data, reducing the energy. Furthermore, though solving PDEs with neural networks has traditionally focused on learning on Euclidean spaces, recently Li et al. [171] proposed a new architecture which includes "Fourier neural operators" to generalize this to functional spaces. They showed it achieves greater accuracy compared to previous ML-based solvers and also can solve entire families of PDEs instead of just one. There is a vast amount of other work using physics-guided architecture towards solving PDEs and other PDE-related applications as well which are not included in this survey (e.g. see ICLR workshop on deep learning for differential equations ([5]))

A recent direction also relating to conserved or invariant quantities is the incorporation of the Hamiltonian operator into NNs [64, 112, 268, 317]. The Hamiltonian operator in physics is the primary tool for modeling the time evolution of systems with conserved quantities, but until recently the formalism had not been integrated with NNs. Greydanus et al. [112] design a NN architecture that naturally learns and respects energy conservation and other invariance laws in simple mass-spring or pendulum systems. They accomplish this through predicting the Hamiltonian of the system and re-integrating instead of predicting the state of physical systems themselves. This is taken a step further in Toth et al. [268], where they show that not only can NNs learn the Hamiltonian, but also the abstract phase space (assumed to be known in Greydanus et al. [112]) to more effectively model expressive densities in similar physical systems and also extend more generally to other problems in physics. Recently, the Hamiltonian-parameterized NNs above have also been expanded into NN architectures that perform additional differential equation-based integration steps based on the derivatives approximated by the Hamiltonian network [61].

*Encoding other domain-specific physical knowledge.* Various other domain-specific physical information can be encoded into architecture that doesn't exactly correspond to known invariances but provides meaningful structure to the optimization process depending on the task at hand. This can take place in many ways, including using domain-informed convolutions for CNNs, additional domain-informed discriminators in GANs, or structures informed by the physical characteristics of the problem. For example, Sadoughi et al. [239] prepend a CNN with a Fast Fourier Transform layer and a physics-guided convolution layer based on known physical information pertaining to fault detection of rolling element bearings. A similar approach is used in Sturmfels et al. [255], but the added beginning layer instead serves to segment different areas of the brain for domain guidance in neuroimaging tasks. In the context of generative models, Xie et al. [296] introduce tempoGAN, which augments a general adversarial network with an additional discriminator network along with additional loss function terms that preserve temporal coherence in the generation of physics-based simulations of fluid flow. This type of approach, though found mostly in NN models, has been extended to non-NN models in Baseman et al. [24], where they introduce a physics-guided Markov Random Field that encodes spatial and physical properties of computer memory devices into the corresponding probabilistic dependencies.

Fan et al. [89] define new architectures to solve the inverse problem of electrical impedance tomography, where the goal is to determine the electrical conductivity distribution of an unknown medium from electrical measurements along its boundary. They define new NN layers based on a linear approximation of both the forward and inverse maps relying on the nonstandard form of the wavelet decomposition [33].

Architecture modifications are also seen in dynamical systems research encoding principles from differential equations. Chen et al. [58] develop a continuous depth NN based on the Residual Network [122] for solving ordinary differential equations. They change the traditionally discretized neuron layer depths into continuous equivalents such that hidden states can be parameterized by differential equations in continuous time. This allows for increased computational efficiency due to the simplification of the backpropagation step of training, and also creates a more scalable normalizing flow, an architectural component for solving PDEs. This is done by by parameterizing the *derivative* of the hidden states of the NN as opposed to the states themselves. Then, in a similar application, Chang

et al. [53] uses principles from the stability properties of differential equations in dynamical systems modeling to guide the design of the gating mechanism and activation functions in an RNN.

Currently, human experts have manually developed the majority of domain knowledge-encoded employed architectures, which can be a time-intensive and error-prone process. Because of this, there is increasing interest in automated neural architecture search methods [20, 85, 126]. A young but promising direction in ML architecture design is to embed prior physical knowledge into neural architecture searches. Ba et al. [18] adds physically meaningful input nodes and physical operations between nodes to the neural architecture search space to enable the search algorithm to discover more ideal physics-guided ML architectures.

*Auxiliary Task in Multi-Task Learning.* Domain knowledge can be incorporated into ML architecture as auxiliary tasks in a multi-task learning framework. Multi-task learning allows for multiple learning tasks to be solved at the same time, ideally while exploiting commonalities and differences across tasks. This can result in improved learning efficiency and predictions for one or more of the tasks. Therefore, another way to implement physics-based learning constraints is to use an auxiliary task in a multi-task learning framework. Here, an example of an auxiliary task in a multi-task framework might be related to ensuring physically consistent solutions in addition to accurate predictions. The promise of such an approach was demonstrated for a computer vision task by integrating auxiliary information (e.g. pose estimation) for facial landmark detection [315]. In this paradigm, a task-constrained loss function can be formulated to allow errors of related tasks to be back-propagated jointly to improve model generalization. Early work in a computational chemistry application showed that a NN could be trained to predict energy by constructing a loss function that had penalties for both inaccuracy *and* inaccurate energy derivatives with respect to time as determined by the surrounding energy force field [219]. In particle physics, De Oliveira et al. [72] uses an additional task for the discriminator network in a generative adversarial network (GAN) to satisfy certain properties of particle interaction for the production of jet images of particle energy.

*Physics-guided Gaussian process regression.* Gaussian process regression (GPR) [287] is a nonparametric, Bayesian approach to regression that is increasingly being used in ML applications. GPR has several benefits, including working well on small amounts of data and enabling uncertainty measurements on predictions. In GPR, first a Gaussian process prior must be assumed in the form of a mean function and a matrix-valued kernel or covariance function. One way to incorporate physical knowledge in GPR is to encode differential equations into the kernel [258]. This is a key feature in Latent Force Models which attempt to use equations in the physical model of the system to inform the learning from data [10, 182]. Alvarez et al. [10] draw inspiration from similar applications in bioinformatics [101, 165], which showed an increase in predictive ability in computational biology, motion capture, and geostatistics datasets. More recently, Glielmo et al. [108] propose a vectorial GPR that encodes physical knowledge in the matrix-valued kernel function. They show rotation and reflection symmetry of the interatomic force between atoms can be encoded in the Gaussian process with specific invariance-preserving covariant kernels. Furthermore, Raissi et al. [225] show that the covariance function can explicitly encode the underlying physical laws expressed by differential equations in order to solve PDEs and learn with smaller datasets.

## 3.4 Hybrid Physics-ML Models

Contrary to previous sections where the focus has been on augmenting ML models specifically, numerous approaches combine physics-based models with ML models where both are operating simultaneously. We call these Hybrid Physics-ML models. In the context of Figure 1, hybrid models can be viewed as replacing mechanistic model $F()$ with a new model in which $F()$ and an ML model are working together, or a subcomponent of $F()$ is replaced with ML. Hence, such methods are also referred to as ML-enhanced physical models by some researchers [100].

*3.4.1 Residual modeling.* The oldest and most common approach for directly addressing the imperfection of physics-based models in the scientific community is residual modeling, where an ML model (usually linear regression) learns to predict the errors, or residuals, made by a physics-based model [94, 266]. A visualization is shown in Figure 3. The key concept is to learn biases of the physical model (relative to observations) and use it to correct the physical model's predictions. However, one key limitation in residual modeling is its inability to enforce physics-based constraints (like in Section 3.1) because such approaches model the errors instead of the physical quantities in scientific problems.
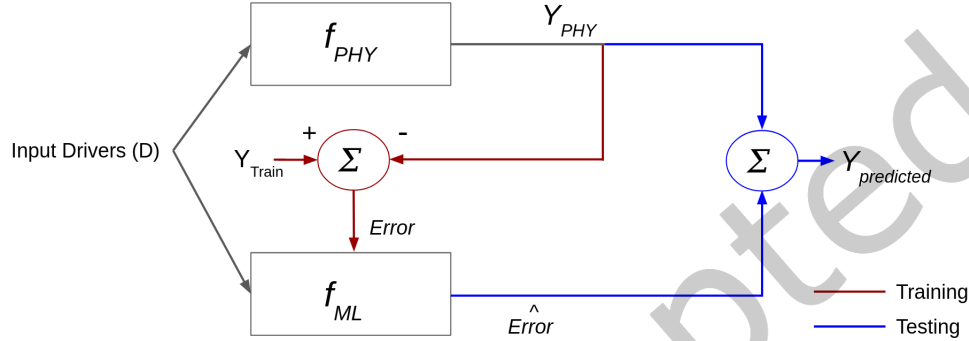


Fig. 3. An illustration of the concept of residual modeling where an ML model $f_{ML}$ is trained to model the error made by the physics-based model $f_{PHY}$. Final predictions are then the sum of the predictions made by $f_{PHY}$ and the residual modeled by $f_{ML}$. Processes shown in red and blue are training and testing respectively. Figure adapted from [94].

Recently, a key area in which residual modeling has been applied is in reduced order models (ROMs) of dynamical systems (described in Section 2.4). After reducing model complexity to create a ROM, an ML model can be used to model the residual due to the truncation. ROM methods were created in response to the problem of many detailed simulations being too expensive to be used in various engineering tasks including design optimization and real-time decision support. In San et al. [241, 242], a simple NN used to model the error due to the model reduction is shown to sharply reduce high error regions when applied to known differential equations. Also, in Wan et al. [278], an RNN is used to model the residual between a ROM for prediction of extreme weather events and the available data projected to a reduced-order space.

As another example, in Kani et al. [138] a physics-driven "deep residual recurrent neural network (DR-RNN)" is proposed to find the residual minimiser of numerically discretized PDEs. Their architecture involves a stacked RNN embedded with the dynamical structure of the PDEs such that each layer of the RNN solves a layer of the residual equations. They showed that DR-RNN sharply reduces both computational cost and time discretization error of the reduced order modeling framework. Finally in Blakseth et al. [36], a feed-forward neural network is used generate a corrective source term that augments the discretized governing equation of a physics-based model for improved prediction performance. This is a more advanced residual model since the ML model is modifying the governing equation itself instead of just the output.

*3.4.2 Output of physical model as input to ML model.* In recent years many other hybrid physics-ML models have been created that extend beyond residual modeling. Another straightforward method to combine physics-based and ML models is to feed the output of a physics-based model as input to an ML model. Karpatne et al [140] showed that using the output of a physics-based model as one feature in an ML model along with inputs used to drive the physics-based model for lake temperature modeling can improve predictions. Visualization of this method

is shown in Figure 4. As we discuss below, there are multiple other ways of constructing a hybrid model, including replacing part of a larger physical model, or weighting predictions from different modeling paradigms depending on context.
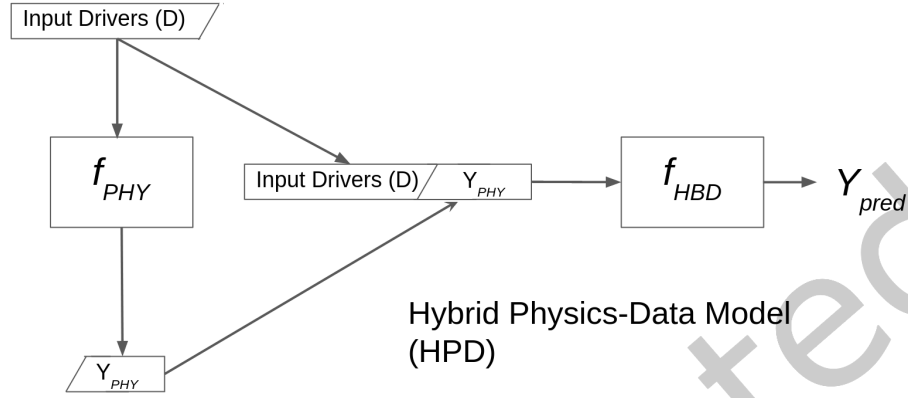


Fig. 4. Diagram of a hybrid physics-ML model which accepts the output of a physical model as input to an ML model (Figure adapted from Karpatne et al. [140]). In the diagram, the physics-based model converts the input drivers $D$ to simulated outputs $Y_{PHY}$. Then, the hybrid physics-ML model $f_{HPD}$ jointly uses the input drivers $D$ and the simulated outputs $Y_{PHY}$ to make the final prediction $Y_{pred}$

*3.4.3 Replacing part of a physical model with ML.* In one variant of hybrid physics-ML models, ML models are used to replace one or more components of a physics-based model or to predict an intermediate quantity that is poorly modeled using physics. For example, to improve predictions of the discrepancy of Reynolds-Averaged Navier–Stokes (RANS) solvers in fluid dynamics, Parish et al. [212] propose a NN to estimate variables in the turbulence closure model to account for missing physics. They show this correction to traditional turbulence models results in convincing improvements of predictions. In Hamilton et al. [117], a subset of the mechanistic model's equations are replaced with data-driven nonparametric methods to improve prediction beyond the baseline process model. In Zhang et al. [312], a physics-based architecture for power system state estimation embeds a deep learning model in place of traditional predicting and optimization techniques. To do this, they substitute NN layers into an unrolled version of an existing solution framework which drastically reduced the overall computational cost due to the fast forward evaluation property of NNs, but kept information of the underlying physical models of power grids and of physical constraints.

ML models for parameterization (see Section 2.3) can also be viewed as a type of hybrid modeling. The vast majority of these efforts use black box ML models [52, 109, 207, 230], but some of these parameterization models can use more sophisticated physics-guided versions of ML as we mentioned in Section 3.1.

*3.4.4 Combining predictions from both physical model and ML model.* In another class of hybrid frameworks, the overall prediction is a combination of predictions from a physical model and an ML model, where the weights depend on prediction circumstance. For example, long-range interactions (e.g. gravity) can often be more easily modeled by classical physics equations than more stochastic short-range interactions (quantum mechanics) that are better modeled using data-driven alternatives. Hybrid frameworks like this have been used to adaptively combine ML predictions for short-range processes and physics model predictions for long-range processes for applications in chemical reactivity [305] and seismic activity prediction [211]. Estimator quality at a given time and location can also be used to determine whether a prediction comes from the physical model or the ML model,

which was shown in Chen et al. [60] for air pollution estimation and in Vlachas et al. [276] for dynamical system forecasting more generally. In the context of solving PDEs, Malek et al [188] showcases a hybrid NN and traditional optimization technique to find the closed analytical form of the solution of a PDE. In this hybrid solver, there exist two terms, a term described by the NN and a term described by traditional optimization techniques.

*3.4.5 ML informing or augmenting physics-model for inverse modeling.* Moreover, in inverse modeling, there is a growing use of hybrid models that first use physics-based models to perform the direct inversion, then use deep learning models to further refine the inverse model's predictions. Multiple works have shown an effective application for this in computed tomography (CT) reconstructions [45, 135]. Another common technique in inverse modeling of images (e.g. medical imaging, particle physics imaging), is the use of CNNs as *deep image priors* [271]. To simultaneously exploit data and prior knowledge, Senouf et al. [246] embed a CNN that serves as the image prior for a physics-based forward model for MRIs.

## 3.5 Requirements and Benefits from Different Physics-ML Methodologies

| Physics-ML Method | Requirements | Possible Benefits |
|---|---|---|
| Loss Function | Known physical relationship (e.g. physical laws, PDEs) | Physical consistency, Improved generalization, Reduced observations required, Improved accuracy |
| Initialization | Synthetic data from mechanistic model available during training | Reduced observations required Improved accuracy |
| Architecture | Intermediate physical variables/processes, *or* hard constraints (e.g. symmetries), *or* task interrelationships, *or* informed prior distributions | Interpretability, Physical consistency, Improved generalization, Reduced solution search space, Improved accuracy |
| Hybrid | Operational mechanistic model available during run time | Improved accuracy |

Table 2. Summary of requirements and possible benefits from different physics-ML methodologies. The left column corresponds to the four types of methods described earlier in Section 3

Methodologies for integrating scientific knowledge in ML described in this section encompass the vast majority of work on this topic. Table 2 summarizes these by listing the requirements needed for different types of methods and the corresponding possible benefits. As we can see, depending on the context of the problem or available resources, different methods can be optimal. Hybrid methods like residual modeling are the simplest case, as they require no process-based knowledge beyond an operational mechanistic model to be used during run time. Physics-guided loss functions require additional domain expertise to determine what terms to add to the loss function, and ML cross-validation techniques are also recommended to weight the different loss function terms. Many of the foundational works on physics-guided loss functions also include open source code that could be adapted to new applications (e.g. Raissi et al. [226], Read et al. [231], Wang et al. [282]). For physics-guided initialization, domain expertise can be used to determine the most relevant synthetic data for the application, but the ML can remain process-agnostic. Physics-guided architecture is often the most complex approach, where

both domain and ML expertise is needed, for example, to customize neural networks by establishing physically meaningful connections and nodes. Note that there can also be multiple Physics-ML method options for a given computational benefit. For example, incorporating physical consistency into ML models can be done through weak constraints as in a loss function, hard constraints through new architectures, or indirectly through physically consistent training data from a mechanistic model simulation.

Note that for a given application-centric objective, only some of these methods may be applicable. For example, hybrid methods will not be suitable for solving PDEs since the goal of reduced computational complexity cannot be reached if the existing solver is still needed to produce the output ($y_t$ in Figure 1). Also in the case of discovering governing equations, there often isn't a known physical model to compare to for either creating a residual model or hybrid approach. Data generation applications also do not make sense for residual modeling since the purpose is to simulate a data distribution rather than improve on a physical model.

Many of the physics-ML methods can also be combined. For example, a physics-guided loss function, physics-guided architecture, and physics-guided initialization could all be applied to an ML model. We saw in Section 3.1 that Jia et al. [130] and Read et al. [231] in particular combined physics-guided loss functions with physics-guided initialization. Also, Karpatne et al [140] combined a physics-guided loss function with a hybrid physics-ML framework. More recently, Jia et al [133] combine physics-guided initialization and physics-guided architecture.

An overall goal of physics-ML methods presented in this section is to address resource efficiency issues (i.e., the ability to solve problems with less computational resources in the context of objectives defined in Section 2) while maintaining high predictive performance, sample efficiency, and interpretability relative to traditional ML approaches. For example, physics-ML methods for solving PDEs (Section 2.5) are likely to be more computationally efficient than direct numerical approaches and more physically consistent than traditional ML approaches. As another example, for the objective of downscaling (Section 2.2), physics-ML methods can be expected to provide high resolution $y_t$ but at much smaller computational cost than possible via traditional mechanistic models and provide much better quality output while using fewer training samples relative to traditional ML approaches. Another major utility of physics-ML methods is to reduce the overall solution search space, which has a direct impact on sample efficiency (i.e. reduced number of observations required) and the amount of computation time taken for model training. For example, physics-ML methods for discovering governing equations can be expected to work with much fewer observations and take less computation time relative to traditional ML methods.

## 4 AREAS OF CURRENT WORK AND POSSIBILITIES FOR CROSS-FERTILIZATION

Table 3 provides a systematic organization and taxonomy of the application-centric objectives and methods of existing physics-based ML applications. This table provides a convenient organization for papers discussed in this survey and other papers that could not be discussed because of space limitations. Importantly, analysis of works within our taxonomy uncovers knowledge gaps and potential crossovers of methods between disciplines that can serve as ideas for future research.

Indeed, there are a myriad of opportunities for taking ideas across applications, objectives, and methods, as well as bringing them back to the traditional ML discipline. For example, the physics-guided NN approaches developed for aquatic sciences [121, 132] can be used in any application where an imperfect mechanistic model is available. Raissi et al. [224] take physics-guided loss function methods to solve PDEs and extend them to inverse modeling problems in fluid dynamics. Furthermore, Daw et al. [68] develop a monotonicity-preserving architecture for lake temperature based on previous work done using loss function terms and a hybrid physics-ML architecture [140]. As another example Jia et al. [133] use physics to inform the propagation of knowledge in a graph neural network. Future research along this direction may shed light on the interpretability of hidden variables in graph neural network models and also how to build dynamic graph structures based on physics.

From Table 3, it is easy to see that several boxes are rather sparse or entirely empty, many of which represent opportunities for future work. For example, ML models for parameterization are increasingly being used in domains such as climate science and weather forecasting [153], all of which can benefit from the integration of physical principles. Furthermore, principles from super-resolution frameworks, originally developed in the context of computer vision applications, are beginning to be applied to downscaling to create higher resolution climate predictions [273]. However, most of these do not incorporate physics (e.g., through additional loss function terms or an informed design of architecture). The fact that nearly all of the other methods (columns) except for hybrid modeling are applicable to this task shows that there is tremendous scope for new exploration, where research pursued in these columns in the context of other objectives can be applied for this objective. We also see a lot of opportunities for new research on physics-guided initialization, where, for example, an ML algorithm could be pre-trained for inverse modeling.

Not all promising research directions are covered by Table 3 and the previous discussion. For instance, one promising direction is to forecast future events using ML and continuously update model states by incorporating ideas from data assimilation [137]. An instance of this is pursued by Dua et al. [80] to build an ML model that predicts the parameters of a physical model using past time series data as an input. Another instance of this approach is seen in epidemiology, where Magri et al. [187] used a NN for data assimilation to predict a parameter vector that captures the time evolution of a COVID-19 epidemiological model. Such approaches can benefit from the ideas in Section 3 (e.g., physics-based loss, intermediate physics variables, etc.). Another direction is to combine scientific knowledge and machine learning to better inform human decisions on environmental or engineering systems. For example, by using anticipated water temperature, one may build new reinforcement learning algorithms to dynamically decide when and how much water to release from reservoirs to a river network [131]. Similarly, such techniques for decision making can be used for automated control in the power plant.

## 5  CONCLUDING REMARKS

Given the current deluge of sensor data and advances in ML methods, we envision that the merging of principles from ML and physics will play an invaluable role in the future of scientific modeling to address the pressing environmental and physical modeling problems facing society. The application-centric objectives defined in Section 2 span the primary communities and disciplines that have both contributed to and benefit from physics-ML integration in a significant way. We believe these categories both provide perspective on the different ways of viewing the physics-ML integration methodologies in Section 3 for different purposes and also allow for coverage of a variety of disciplines that have been pursuing these ideas mostly-independently in recent years. Researchers working in one of these objectives can see how their methods fit within the taxonomy and relate them to how they are being used in other objectives. Our hope is that this survey will accelerate the cross-pollination of ideas among these diverse research communities.

The discussion and structure provided in this survey also serve to benefit the ML community, where, for example, techniques for adding physical constraints to loss functions can be used to enforce fairness in predictive models, or realism for data generated by GANs. Further, novel architecture designs can enable new ways to incorporate prior domain information (beyond what is usually done using Bayesian frameworks) and can lead to better interpretability.

This survey focuses primarily on improving the modeling of engineering and environmental systems that are traditionally solved using mechanistic modeling. However, the general ideas discussed here for integrating scientific knowledge in ML have wider applicability, and such research is already being pursued in many other contexts. For example, there are several interesting works in system control which often involves reinforcement learning techniques (e.g., combining model predictive control with Gaussian processes in robotics [13], informed priors for neuroscience modeling [104], physics-based reward functions in computational chemistry [63], and fluidic feedback control from a cylinder [152]). Other examples include identifying features of interest in the output of

Table 3.  Table of literature classified by objective and method

| | Physics-Guided Loss Function (3.1) | Physics-Guided Initialization (3.2) | Physics-Guided Architecture (3.3) | Hybrid Model (3.4) | |
| --- | --- | --- | --- | --- | --- |
| | | | | Residual (3.4.1) | Other (3.4.2-3.4.5) |
| Improve or replace physical model (2.1) | [266] [10] [219] [158] [140] [182] [203] [76] [86] [130] [174] [204] [231] [313] [121] [125] [187] [92] [306] | [180] [298] [184] [267] [39] [247] [133] [130] [231] [298] | [173] [66] [24] [255] [12] [17] [203] [204] [213] [239] [125] [281] [214] [176] [249] [310] [309] [311] [245] [78] | [266] [297] [279] [241] [278] [293] [174] [19] | [109] [113] [240] [117] [140] [253] [60] [78] [179] [211] [305] [312] [276] [302] [111] [187] [55] [36] |
| Parameterization (2.2) | [310] [32] [31] [306] | | [28] [32] [312] | | |
| Downscaling (2.3) | [37] [134] | | [201] [274] | | |
| Reduced Order Models (2.4) | [209] [16] [167] | | [138] [86] [210] | [138] [241] [242] [278] [115] [210] | [67] |
| Solve PDEs (2.5) | [226] [251] [301] [303] [70] [196] [228] [248] [318] [82] [102] [144] [292] [216] [195] [23] [90] [320] [154] | | [58] [236] [53] [70] [192] [252] [26] [61] [149] [88] [299] [216] [195] [225] [145] [198] [171] [44] | | [188] |
| Inverse modeling (2.6) | [224] [136] | | [35] [89] [256] [150] | [123] | [212] [117] [135] [257] [48] [45] [79] [246] |
| Discover Governing Equations (2.7) | [227] [178] | | [270] [177] [62] [169] | | |
| Data Generation (2.8) | [72] [49] [37] [316] [151] [292] [308] | | [59] [296] [248] | | |
| Uncertainty Quantification (2.9) | [290] [303] [288] [304] [318] [102] | [298] [184] | [68] [300] [288] [274] | | [77] |

computational simulations of physics-based models (e.g., high-impact weather predictions [194], segmentation of climate models [202], and tracking phenomena from climate model data [97, 237]). There is also recent work on encoding domain knowledge in geometric deep learning [41, 50] that is finding increasing use in computational chemistry [71, 81, 95], physics [25, 54], hydrology [133], geostatistics [14, 294], and neuroscience [155]. We expect that there will be a lot of potential for cross-over of ideas amongst these different efforts that will greatly expedite research in this nascent field.

## REFERENCES

[1] 2019. *ICERM Workshop on Scientific Machine Learning*. https://icerm.brown.edu/events/ht19-1-sml/
[2] 2020. *1st Workshop on Knowledge Guided Machine Learning (KGML): A Framework for Accelerating Scientific Discovery*. https://sites.google.com/umn.edu/kgml/workshop
[3] 2020. *AAAI Symposium on Physics-Guided AI*. https://sites.google.com/vt.edu/pgai-aaai-20
[4] 2020. *IGARS 2020 Symposium on Incorporating Physics into Deep Learning*. https://igarss2020.org/Papers/ViewSession_MS.asp?Sessionid=1016
[5] 2020. *International Conference on Learning Representations 2020 Workshop on Integration of Deep Neural Models and Differential Equations*. https://openreview.net/group?id=ICLR.cc/2020/Workshop/DeepDiffEq

[6] 2021. *AAAI Spring Symposium on Combining Artificial Intelligence and Machine Learning with Physics Sciences*. https://sites.google.com/view/aaai-mlps

[7] S Aditya, Y Yang, and C Baral. 2019. Integrating knowledge and reasoning in image understanding. *arXiv:1906.09954* (2019).

[8] M Alber et al. 2019. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ Digit. Med.* 2, 1 (2019), 1–11.

[9] B Alipanahi et al. 2015. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology* 33, 8 (2015), 831–838.

[10] M. Alvarez et al. 2009. Latent force models. In *Artificial Intelligence and Statistics*. 9–16.

[11] D Amsallem and C Farhat. 2008. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA journal* 46, 7 (2008), 1803–1813.

[12] B Anderson, TS Hy, and R Kondor. 2019. Cormorant: Covariant molecular neural networks. In *Adv. Neural Inf. Process. Syst.* 14510–14519.

[13] O Andersson, F Heintz, and P Doherty. 2015. Model-based reinforcement learning in continuous environments using real-time constrained optimization. In *AAAI*.

[14] G Appleby, L Liu, and L Liu. 2020. Kriging Convolutional Networks.. In *AAAI*. 3187–3194.

[15] L Arsenault et al. 2014. Machine learning for many-body physics: The case of the Anderson impurity model. *Phys. Rev. B* (2014).

[16] O Azencot et al. 2020. Forecasting Sequential Data using Consistent Koopman Autoencoders. *arXiv:2003.02236* (2020).

[17] Y Ba et al. 2019. Physics-based neural networks for shape from polarization. *arXiv:1903.10210* (2019).

[18] Y Ba, G Zhao, and A Kadambi. 2019. Blending diverse physical priors with neural networks. *arXiv:1910.00201* (2019).

[19] M Bahari, I Nejjar, and A Alahi. 2021. Injecting knowledge in data-driven vehicle trajectory predictors. *Transp. Res. Part C Emerg.* 128 (2021), 103010.

[20] B Baker et al. 2016. Designing neural network architectures using reinforcement learning. *arXiv:1611.02167* (2016).

[21] N Baker et al. 2019. *Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence.* Technical Report. USDOE Office of Science.

[22] P Baldi et al. 2016. Jet substructure classification in high-energy physics with deep neural networks. *Phys. Rev. D* 93, 9 (2016), 094034.

[23] Y Bar-Sinai et al. 2019. Learning data-driven discretizations for partial differential equations. *PNAS* 116, 31 (2019), 15344–15349.

[24] E Baseman et al. 2018. Physics-Informed Machine Learning for DRAM Error Modeling. In *IEEE DFT*.

[25] P Battaglia et al. 2016. Interaction networks for learning about objects, relations and physics. In *Adv. Neural Inf. Process. Syst.* 4502–4510.

[26] C Beck, E Weinan, and A Jentzen. 2019. Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *J Nonlinear Sci.* 29, 4 (2019), 1563–1619.

[27] J Behler. 2011. Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *Phys. Chem* 13, 40 (2011), 17930–17955.

[28] J Behler and M Parrinello. 2007. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* 98, 14 (2007), 146401.

[29] A Bennett and B Nijssen. 2020. Deep learned process parameterizations provide better representations of turbulent heat fluxes in hydrologic models. *Water Resour. Res.* (2020), e2020WR029328.

[30] Karianne J Bergen et al. 2019. Machine learning for data-driven discovery in solid Earth geoscience. *Science* 363, 6433 (2019), eaau0323.

[31] P Beucler et al. 2019. Achieving conservation of energy in neural network emulators for climate modeling. *arXiv:1906.06622* (2019).

[32] T Beucler et al. 2019. Enforcing Analytic Constraints in Neural-Networks Emulating Physical Systems. *arXiv:1909.00912* (2019).

[33] G Beylkin, R Coifman, and V Rokhlin. 1991. Fast wavelet transforms and numerical algorithms I. *Commun. Pure Appl. Math.* 44, 2 (1991), 141–183.

[34] I Bilionis and N Zabaras. 2012. Multi-output local Gaussian process regression: Applications to uncertainty quantification. *J. Comput. Phys* (2012).

[35] R Biswas et al. 2019. Prestack and poststack inversion using a physics-guided convolutional neural network. *Interpretation* 7, 3 (2019), SE161–SE174.

[36] SS Blakseth et al. 2022. Deep neural network enabled corrective source term approach to hybrid analysis and modeling. *Neural Netw* 146 (2022), 181–199.

[37] M Bode et al. 2021. Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows. *Proc Combust Inst.* (2021).

[38] J Bongard and H Lipson. 2007. Automated reverse engineering of nonlinear dynamical systems. *PNAS* (2007). https://doi.org/10.1073/pnas.0609476104

[39] K others Bousmalis. 2018. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *IEEE ICRA*. IEEE, 4243–4250.

[40] ND Brenowitz and CS Bretherton. 2018. Prognostic validation of a neural network unified physics parameterization. *Geophys. Res. Lett.* 45, 12 (2018), 6289–6298.

[41] MM Bronstein et al. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Process. Mag.* 34, 4 (2017), 18–42.

[42] S Brunton et al. 2017. Koopman operator theory: Past, present, and future. *APS* (2017), L27–004.

[43] S Brunton, J Proctor, and J Kutz. 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *PNAS* (2016).

[44] J Bu and A Karpatne. 2021. Quadratic residual networks: A new class of neural networks for solving forward and inverse problems in physics involving pdes. In *SDM*. SIAM, 675–683.

[45] T A Bubba et al. 2019. Learning the invisible: A hybrid deep learning-shearlet framework for limited angle computed tomography. *Inverse Probl.* (2019).

[46] S Cai et al. 2021. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *arXiv:2105.09506* (2021).

[47] PM Caldwell et al. 2014. Statistical significance of climate sensitivity predictors obtained by data mining. *Geophys. Res. Let.* 41, 5 (2014), 1803–1808.

[48] G Camps-Valls et al. 2018. Physics-aware Gaussian processes in remote sensing. *Appl. Soft Comput.* 68 (2018), 69–82.

[49] R Cang, H Li, H Yao, Y Jiao, and Y Ren. 2018. Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model. *Computational Materials Science* 150 (2018), 212–221.

[50] W Cao et al. 2020. A comprehensive survey on geometric deep learning. *IEEE Access* 8 (2020), 35929–35949.

[51] G Carleo and M Troyer. 2017. Solving the quantum many-body problem with artificial neural networks. *Science* 355, 6325 (2017), 602–606.

[52] S Chan and A Elsheikh. 2017. Parametrization and generation of geological models with generative adversarial networks. *arXiv:1708.01810* (2017).

[53] B Chang et al. 2019. AntisymmetricRNN: A dynamical system view on recurrent neural networks. *arXiv:1902.09689* (2019).

[54] MB Chang et al. 2016. A compositional object-based approach to learning physical dynamics. *arXiv:1612.00341* (2016).

[55] MA Chao et al. 2022. Fusing physics-based and deep learning models for prognostics. *Reliab. Eng. Syst. Saf.* 217 (2022), 107961.

[56] G Chen et al. 2012. Support-vector-machine-based reduced-order model for limit cycle oscillation prediction of nonlinear aeroelastic system. *Math. Probl. Eng* 2012 (2012).

[57] H Chen et al. 2017. Low-dose CT with a residual encoder-decoder convolutional neural network. *IEEE T-MI* (2017).

[58] TQ Chen et al. 2018. Neural ordinary differential equations. In *NIPS*.

[59] W Chen and M Fuge. 2018. BezierGAN: Automatic Generation of Smooth Curves from Interpretable Low-Dimensional Parameters. *arXiv:1808.08871* (2018).

[60] X Chen et al. 2018. Pga: Physics guided and adaptive approach for mobile fine-grained air pollution estimation. In *Ubicomp*.

[61] Z Chen et al. 2019. Symplectic Recurrent Neural Networks. *arXiv:1909.13334* (2019).

[62] Z Chen, Y Liu, and H Sun. 2020. Deep learning of physical laws from scarce data. *arXiv:2005.03448* (2020).

[63] Y Cho et al. 2019. Physics-guided reinforcement learning for 3D molecular structures. In *NeurIPS*.

[64] A Choudhary et al. 2019. Physics enhanced neural networks predict order and chaos. *arXiv:1912.01958* (2019).

[65] M Christie, V Demyanov, and D Erbas. 2006. Uncertainty quantification for porous media flows. *J. Comput. Phys* (2006).

[66] TS Cohen et al. 2018. Spherical cnns. *arXiv:1801.10130* (2018).

[67] T Daniel et al. 2020. Model order reduction assisted by deep neural networks (ROM-net). *Adv. Model. Simul. Eng.* 7 (2020), 1–27.

[68] A Daw et al. 2019. Physics-Guided Architecture (PGA) of Neural Networks for Quantifying Uncertainty in Lake Temperature Modeling. *arXiv:1911.02682* (2019).

[69] MS Dawson et al. 1992. Inversion of surface parameters using fast learning neural networks. *IGARSS Symposium* (1992).

[70] E de Bezenac, A Pajot, and P Gallinari. 2019. Deep learning for physical processes: Incorporating prior scientific knowledge. *J. Stat. Mech.: Theory Exp* 2019, 12 (2019), 124009.

[71] Nicola De Cao and Thomas Kipf. 2018. MolGAN: An implicit generative model for small molecular graphs. *arXiv:1805.11973* (2018).

[72] L de Oliveira, M Paganini, and B Nachman. 2017. Learning particle physics by example: location-aware generative adversarial networks for physics synthesis. *Comput. Softw. Big Sci* (2017).

[73] EL Denton et al. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*.

[74] C Deser, A Phillips, V Bourdette, and H Teng. 2012. Uncertainty in climate change projections: the role of internal variability. *Climate dynamics* (2012).

[75] MWMG Dissanayake and N Phan-Thien. 1994. Neural-network-based approximations for solving partial differential equations. *Commun. Numer. Methods Eng.* 10, 3 (1994), 195–201.

[76] NAK Doan, W Polifke, and L Magri. 2019. Physics-informed echo state networks for chaotic systems forecasting. In *ICCS*. Springer.

[77] B Dong, Z Li, SMM Rahman, and R Vega. 2016. A hybrid model approach for forecasting future residential electricity consumption. *Energy Build.* (2016).

[78] Arinan D Dourado and Felipe Viana. 2020. Physics-Informed Neural Networks for Bias Compensation in Corrosion-Fatigue. In *AIAA Forum*. 1149.

[79] JE Downton and DP Hampson. 2019. Use of theory-guided neural networks to perform seismic inversion. In *GeoSoftware, CGG*.

[80] V Dua. 2011. An artificial neural network approximation based decomposition approach for parameter estimation of system of ordinary differential equations. *Comput. Chem. Eng.* 35, 3 (2011), 545–553.

[81] DK Duvenaud et al. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *Adv. Neural Inf. Process. Syst.* 2224–2232.

[82] V Dwivedi and B Srinivasan. 2020. Solution of Biharmonic Equation in Complicated Geometries with Physics Informed Extreme Learning Machine. *J. Comput. Inf. Sci. Eng.* (2020), 1–10.

[83] S Džeroski et al. 1999. Equation discovery with ecological applications. In *Machine learning methods for ecological applications*. Springer, 185–207.

[84] I Ebert-Uphoff et al. 2021. CIRA Guide to Custom Loss Functions for Neural Networks in Environmental Sciences–Version 1. *arXiv:2106.09757* (2021).

[85] T Elsken, JH Metzen, and F Hutter. 2018. Neural architecture search: A survey. *arXiv:1808.05377* (2018).

[86] NB Erichson, M Muehlebach, and MW Mahoney. 2019. Physics-informed autoencoders for lyapunov-stable fluid flow prediction. *arXiv:1905.10866* (2019).

[87] JH Faghmous and V Kumar. 2014. A big data guide to understanding climate change: The case for theory-guided data science. *Big data* 2, 3 (2014), 155–163.

[88] Y Fan et al. 2019. A multiscale neural network based on hierarchical matrices. *Multiscale Model Simul.* 17, 4 (2019), 1189–1213.

[89] Y Fan and L Ying. 2020. Solving electrical impedance tomography with deep learning. *J. Comput. Phys* 404 (2020), 109119.

[90] Z Fang. 2021. A High-Efficient Hybrid Physics-Informed Neural Networks Based on Convolutional Neural Network. *IEEE Trans Neural Netw Learn Syst.* (2021).

[91] AB Farimani, J Gomes, and VS Pande. 2017. Deep learning the physics of transport phenomena. *arXiv:1709.02432* (2017).

[92] F Fioretto, TWK Mak, and T Van Hentenryck. 2020. Predicting AC optimal power flows: Combining deep learning and lagrangian dual methods. In *AAAI*, Vol. 34. 630–637.

[93] J Fish and T Belytschko. 2007. *A first course in finite elements*. Wiley.

[94] U Forssell and P Lindskog. 1997. Combining Semi-Physical and Neural Network Modeling: An Example ofIts Usefulness. *IFAC Proceedings Volumes* 30, 11 (1997), 767–770.

[95] Alex Fout et al. 2017. Protein interface prediction using graph convolutional networks. In *Adv. Neural Inf. Process. Syst.* 6530–6539.

[96] M Frangos et al. 2010. Surrogate and reduced-order modeling: A comparison of approaches for large-scale statistical inverse problems. *Large-scale inverse problems and quantification of uncertainty* 123149 (2010).

[97] DJ Gagne et al. 2017. Storm-based probabilistic hail forecasting with machine learning applied to convection-allowing ensembles. *Weather Forecast.* 32, 5 (2017), 1819–1840.

[98] Y Gal and Z Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*.

[99] D Galbally et al. 2010. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *Int. J. Numer. Methods Fluids* (2010).

[100] AR Ganguly et al. 2014. Toward enhanced understanding and projections of climate extremes using physics-guided data mining techniques. *Nonlinear Process. Geophys* 21, 4 (2014), 777–795.

[101] P Gao et al. 2008. Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics* 24, 16 (2008), i70–i75.

[102] N Geneva and N Zabaras. 2020. Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. *J. Comput. Phys* (2020).

[103] P Gentine et al. 2018. Could machine learning break the convection parameterization deadlock? *Geophys. Res. Lett* (2018).

[104] SJ Gershman. 2016. Empirical priors for reinforcement learning models. *J. Math. Psychol.* 71 (2016), 1–6.

[105] D Gerwin. 1974. Information processing, data inferences, and scientific generalization. *Behavioral Science* 19, 5 (1974), 314–325.

[106] H Ghorbanidehno et al. 2020. Recent developments in fast and scalable inverse modeling and data assimilation methods in hydrology. *J. Hydrol.* (2020), 125266.

[107] D Giannakis, J Slawinska, and Z Zhao. 2015. Spatiotemporal feature extraction with data-driven Koopman operators. In *Feature Extraction: Modern Questions and Challenges*. 103–115.

[108] A Glielmo, P Sollich, and A De Vita. 2017. Accurate interatomic force fields via machine learning with covariant kernels. *Phys. Rev. B* 95, 21 (2017), 214302.

[109] EB Goldstein, G Coco, AB Murray, and MO Green. 2014. Data-driven components in a model of inner-shelf sorted bedforms: a new hybrid model. *Earth Surf. Dyn.* (2014).

[110] GH Golub and CF Van Loan. 2012. *Matrix computations*. Vol. 3. JHU press.

[111] NP Greis et al. 2020. Physics-guided machine learning for self-aware machining. In *AAAI Symposium – AI and Manufacturing*.

[112] S Greydanus, M Dzamba, and J Yosinski. 2019. Hamiltonian neural networks. In *NIPS*.

[113] A Grover et al. 2015. A deep hybrid model for weather forecasting. In *ACM SIGKDD*. 379–386.

[114] J Guo et al. 2018. Long text generation via adversarial training with leaked information. In *AAAI*.

[115] M Guo and JS Hesthaven. 2019. Data-driven reduced order modeling for time-dependent problems. *Comput. Methods Appl. Mech. Eng.* 345 (2019), 75–99.

[116] HV Gupta et al. 2014. Debates—The future of hydrological sciences: A (common) path forward? Using models and data to learn: A systems theoretic perspective on the future of hydrological science. *Water Resour. Res.* (2014).

[117] F Hamilton et al. 2017. Hybrid modeling and prediction of dynamical systems. *PLOS Comput. Biol* (2017).

[118] J Han, A Jentzen, and E Weinan. 2018. Solving high-dimensional partial differential equations using deep learning. *PNAS* (2018).

[119] J Han, L Zhang, and E Weinan. 2019. Solving many-electron Schrödinger equation using deep neural networks. *J. Comput. Phys* 399 (2019), 108929.

[120] K Hansen et al. 2013. Assessment and validation of machine learning methods for predicting molecular atomization energies. *J. Chem. Theory Comput* 9, 8 (2013), 3404–3419.

[121] PC Hanson et al. 2020. Predicting lake surface water phosphorus dynamics using process-guided machine learning. *Ecol Modell* 430 (2020), 109136.

[122] K He, X Zhang, S Ren, and J Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

[123] JR Holland et al. 2019. Field inversion and machine learning with embedded neural networks: Physics-consistent neural network training. In *AIAA Forum*. 3200.

[124] WW Hsieh. 2009. *Machine learning methods in the environmental sciences: Neural networks and kernels*. Cambridge university press.

[125] X Hu et al. 2020. Physics-Guided Deep Neural Networks for PowerFlow Analysis. *arXiv:2002.00097* (2020).

[126] F Hutter, L Kotthoff, and J Vanschoren. 2019. *Automated machine learning: methods, systems, challenges*. Springer Nature.

[127] G Ibarra-Berastegi et al. 2015. Short-term forecasting of the wave energy flux: Analogues, random forests, and physics-based models. *Ocean Eng.* 104 (2015), 530–539.

[128] Z Ivezić et al. 2019. *Statistics, data mining, and machine learning in astronomy: a practical Python guide for the analysis of survey data*. Princeton University Press.

[129] X Jia et al. 2019. Bringing automated, remote-sensed, machine learning methods to monitoring crop landscapes at scale. *Agric Econ* 50 (2019), 41–50.

[130] X Jia et al. 2019. Physics Guided RNNs for Modeling Dynamical Systems: A Case Study in Simulating Lake Temperature Profiles. In *SDM*. SIAM.

[131] X Jia et al. 2021. Graph-based reinforcement learning for active learning in real time: An application in modeling river networks. In *SDM*. SIAM, 621–629.

[132] X Jia et al. 2021. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ACM/IMS J. of Data Sci.* 2, 3 (2021), 1–26.

[133] X Jia et al. 2021. Physics-Guided Recurrent Graph Model for Predicting Flow and Temperature in River Networks. In *SDM*. SIAM, 612–620.

[134] CM Jiang et al. 2020. Meshfreeflownet: A physics-constrained deep continuous space-time super-resolution framework. *arXiv:2005.01463* (2020).

[135] KH Jin et al. 2017. Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Process* (2017).

[136] A Kahana et al. 2020. Obstacle segmentation based on the wave equation and deep learning. *J. Comput. Phys* (2020), 109458.

[137] E Kalnay. 2003. *Atmospheric modeling, data assimilation and predictability*. Cambridge university press.

[138] J Kani and A Elsheikh. 2017. DR-RNN: A deep residual recurrent neural network for model reduction. *arXiv:1709.00939* (2017).

[139] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. 2021. Physics-informed machine learning. *Nature Reviews Physics* 3, 6 (2021), 422–440.

[140] A Karpatne et al. 2017. Physics-guided neural networks (PGNN): An application in lake temperature modeling. *arXiv:1710.11431* (2017).

[141] A Karpatne et al. 2017. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE TKDE* (2017).

[142] A Karpatne et al. 2018. Machine learning for the geosciences: Challenges and opportunities. *IEEE TKDE* (2018).

[143] A Karpatne, N Ramakrishnan, and V Kumar (Eds.). 2022. *Knowledge-guided machine learning: Accelerating discovery using scientific knowledge and data*. CRC Press.

[144] S Karumuri et al. 2020. Simulator-free solution of high-dimensional stochastic elliptic partial differential equations using deep neural networks. *J. Comput. Phys* (2020).

[145] K Kashinath et al. 2020. Enforcing physical constraints in cnns through differentiable pde layer. In *ICLR Workshop on Integration of Deep Neural Models and Differential Equations*.

[146] K Kashinath et al. 2021. Physics-informed machine learning: case studies for weather and climate modelling. *Philos. Trans. R. Soc. A* 379, 2194 (2021), 20200093.

[147] MF Kasim et al. 2020. Building high accuracy emulators for scientific simulations with deep neural architecture search. *arXiv e-prints* (2020), arXiv–2001.

[148] SK Kauwe et al. 2018. Machine learning prediction of heat capacity for solid inorganics. *Integr. Mater. Manuf. Innov.* 7, 2 (2018), 43–51.

[149] Y Khoo, J Lu, and L Ying. 2019. Solving for high-dimensional committor functions using artificial neural networks. *Res. Math. Sci* 6, 1 (2019), 1.

[150] Y Khoo and L Ying. 2019. SwitchNet: a neural network model for forward and inverse scattering problems. *SIAM J Sci Comput* 41, 5 (2019), A3182–A3201.

[151] B Kim et al. 2019. Deep fluids: A generative network for parameterized fluid simulations. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 59–70.

[152] H Koizumi, S Tsutsumi, and E Shima. 2018. Feedback control of Karman vortex shedding from a cylinder using deep reinforcement learning. In *Flow Control Conference*. 3691.

[153] VM Krasnopolsky and MS Fox-Rabinovitz. 2006. Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Netw* 19, 2 (2006), 122–134.

[154] A Krishnapriyan et al. 2021. Characterizing possible failure modes in physics-informed neural networks. *Adv. Neural Inf. Process. Syst.* 34 (2021).

[155] SI Ktena et al. 2017. Distance metric learning using graph convolutional networks: Application to functional brain networks. In *MICCAI*. Springer, 469–477.

[156] S Kumar et al. 2020. Inverse-designed spinodoid metamaterials. *Npj Comput. Mater.* 6, 1 (2020), 1–10.

[157] NJ Kutz. 2017. Deep learning in fluid dynamics. *J. Fluid Mech.* 814 (2017), 1–4.

[158] L Ladickỳ et al. 2015. Data-driven fluid simulations using regression forests. *ACM Trans. Graph* 34, 6 (2015), 1–9.

[159] IE Lagaris, A Likas, and DI Fotiadis. 1998. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw. Learn* (1998).

[160] JH Lagergren et al. 2020. Learning partial differential equations for biological transport models from noisy spatio-temporal data. *Proc. R. Soc A* 476, 2234 (2020), 20190800.

[161] B Lakshminarayanan, A Pritzel, and C Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*.

[162] P Langley. 1981. Data-driven discovery of physical laws. *Cogn. Sci.* 5, 1 (1981), 31–54.

[163] P Langley, GL Bradshaw, and HA Simon. 1983. Rediscovering chemistry with the BACON system. In *Machine learning*. Springer, 307–329.

[164] T Lassila et al. 2014. Model order reduction in fluid dynamics: challenges and perspectives. In *Reduced Order Methods for modeling and computational reduction*. Springer, 235–273.

[165] ND Lawrence, G Sanguinetti, and M Rattray. 2007. Modelling transcriptional regulation using Gaussian processes. In *Adv. Neural Inf. Process. Syst.* 785–792.

[166] D Lazer et al. 2014. The parable of Google Flu: traps in big data analysis. *Science* (2014).

[167] K Lee and KT Carlberg. 2020. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J. Comput. Phys.* 404 (2020), 108973.

[168] DB Lenat. 1983. The role of heuristics in learning by discovery: Three case studies. In *Machine learning*. Springer, 243–306.

[169] H Li and Y Weng. 2021. Physical Equation Discovery Using Physics-Consistent Neural Network (PCNN) Under Incomplete Observability. In *ACM SIGKDD*. 925–933.

[170] Q Li et al. 2017. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. *Chaos* 27, 10 (2017), 103111.

[171] Z Li et al. 2020. Fourier neural operator for parametric partial differential equations. *arXiv:2010.08895* (2020).

[172] TW Liao and G Li. 2020. Metaheuristic-based inverse design of materials–A survey. *J Materiomics* (2020).

[173] J Ling, A Kurzawski, and J Templeton. 2016. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech* (2016).

[174] D Liu and Y Wang. 2019. Multi-Fidelity Physics-Constrained Neural Network and Its Application in Materials Modeling. *J. Mech. Des.* 141, 12 (2019).

[175] L Liu et al. 2021. Anatomy-aided deep learning for medical image segmentation: a review. *Phys. Med. Biol* (2021).

[176] Y Liu et al. 2020. Multiresolution Convolutional Autoencoders. *arXiv:2004.04946* (2020).

[177] Z Liu and M Tegmark. 2021. Machine Learning Conservation Laws from Trajectories. *Phys. Rev. Lett.* 126, 18 (2021), 180604.

[178] JC Loiseau and SL Brunton. 2018. Constrained sparse Galerkin regression. *J. Fluid Mech.* 838 (2018), 42–67.

[179] Y Long, X She, and S Mukhopadhyay. 2018. HybridNet: integrating model-based and data-driven learning to predict evolution of dynamical systems. *arXiv:1806.07439* (2018).

[180] Junde Lu and Furong Gao. 2008. Model migration with inclusive similarity for development of a new process model. *Ind. Eng. Chem. Res* 47, 23 (2008), 9508–9516.

[181] J Lu, K Yao, and F Gao. 2009. Process similarity and developing new process models through migration. *AIChE journal* 55, 9 (2009), 2318–2328.

[182] D Luengo, M Campos-Taberner, and G Camps-Valls. 2016. Latent force models for earth observation time series prediction. In *IEEE Int. Workshop Mach. Learn.)*. IEEE, 1–6.

[183] S Lunz, O Öktem, and CB Schönlieb. 2018. Adversarial regularizers in inverse problems. In *NIPS*.

[184] L Luo, Y Yao, and F Gao. 2015. Bayesian improved model migration methodology for fast process modeling by incorporating prior information. *Chem. Eng. Sci.* 134 (2015), 23–35.

[185] B Lusch, NJ Kutz, and SL Brunton. 2018. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* 9, 1 (2018), 4950.

[186] D JC MacKay. 1992. A practical Bayesian framework for backpropagation networks. *Neural computation* (1992).

[187] L Magri and NAK Doan. 2020. First-principles machine learning modelling of COVID-19. *arXiv:2004.09478* (2020).

[188] A Malek and R Shekari Beidokhti. 2006. Numerical solution for high order differential equations using a hybrid neural network—optimization method. *Appl. Math. Comput.* 183, 1 (2006), 260–271.

[189] A Manzoni, S Pagani, and T Lassila. 2016. Accurate solution of Bayesian inverse uncertainty quantification problems combining reduced basis methods and reduction error models. *SIAM/ASA JUQ* (2016).

[190] G Marcus and E Davis. 2014. Eight (no, Nine!) Problems with big data. *The New York Times* 6, 04 (2014), 2014.

[191] A Mardt et al. 2018. VAMPnets for deep learning of molecular kinetics. *Nat. Commun.* 9, 1 (2018), 1–11.

[192] Marios Mattheakis et al. 2019. Physical symmetries embedded in neural networks. *arXiv:1904.08991* (2019).

[193] MT McCann, KH Jin, and M Unser. 2017. A review of convolutional neural networks for inverse problems in imaging. *arXiv:1710.04011* (2017).

[194] A McGovern et al. 2017. Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bull. Amer. Meteor.* 98, 10 (2017), 2073–2090.

[195] Xuhui Meng and George Em Karniadakis. 2020. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. *J. Comput. Phys* 401 (2020), 109020.

[196] X Meng, Z Li, D Zhang, and GE Karniadakis. 2019. Ppinn: Parareal physics-informed neural network for time-dependent pdes. *arXiv:1909.10145* (2019).

[197] I Mezić. 2005. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dyn.* 41, 1-3 (2005), 309–325.

[198] AT Mohan et al. 2020. Embedding hard physical constraints in convolutional neural networks for 3D turbulence. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*.

[199] AT Mohan and DV Gaitonde. 2018. A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks. *arXiv:1804.09269* (2018).

[200] J Morton, FD Witherden, and MJ Kochenderfer. 2019. Deep variational koopman models: Inferring koopman observations for uncertainty-aware dynamics modeling and control. *arXiv:1902.09742* (2019).

[201] B Mu et al. 2020. A Climate Downscaling Deep Learning Model considering the Multiscale Spatial Correlations and Chaos of Meteorological Events. *Math. Probl. Eng* 2020 (2020).

[202] M Mudigonda et al. 2017. Segmenting and tracking extreme climate events using neural networks. In *Deep Learning for Physical Sciences (DLPS) Workshop in NIPS Conference*.

[203] N Muralidhar et al. 2018. Incorporating prior domain knowledge into deep neural networks. In *IEEE Big Data*. IEEE.

[204] N Muralidhar et al. 2020. PhyNet: Physics Guided Neural Networks for Particle Drag Force Prediction in Assembly. In *SDM*. SIAM, 559–567.

[205] F Noé et al. 2020. Machine learning for molecular simulation. *Annu. Rev. Phys. Chem.* 71 (2020), 361–390.

[206] P Nowack et al. 2018. Using machine learning to build temperature-based ozone parameterizations for climate sensitivity simulations. *Environ. Res. Lett.* 13, 10 (2018), 104016.

[207] PA O'Gorman and JG Dwyer. 2018. Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. *J. Adv. Model. Earth Syst.* 10, 10 (2018), 2548–2563.

[208] A Oord et al. 2016. Wavenet: A generative model for raw audio. *arXiv:1609.03499* (2016).

[209] SE Otto and CW Rowley. 2019. Linearly recurrent autoencoder networks for learning dynamics. *SIADS* 18, 1 (2019), 558–593.

[210] S Pan and K Duraisamy. 2020. Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability. *SIADS* 19, 1 (2020), 480–509.

[211] R Paolucci et al. 2018. Broadband ground motions from 3D physics-based numerical simulations using artificial neural networks. *BSSA* (2018).

[212] EJ Parish and K Duraisamy. 2016. A paradigm for data-driven predictive modeling using field inversion and machine learning. *J. Comput. Phys* (2016).

[213] J Park and J Park. 2019. Physics-induced graph neural network: An application to wind-farm power estimation. *Energy* 187 (2019), 115883.

[214] S Pawar et al. 2021. Physics guided machine learning using simplified theories. *Phys. Fluids* 33, 1 (2021), 011701.

[215] GCY Peng et al. 2021. Multiscale modeling meets machine learning: What can we learn? *Arch. Comput. Methods Eng.* 28, 3 (2021), 1017–1037.

[216] W Peng et al. 2020. Accelerating Physics-Informed Neural Network Training with Prior Dictionaries. *arXiv:2004.08151* (2020).

[217] CL Pettit. 2004. Uncertainty quantification in aeroelasticity: recent results and research challenges. *J. of Aircraft* (2004).

[218] L Pilozzi et al. 2018. Machine learning inverse problem for topological photonics. *Commun. Phys* (2018).

[219] A Pukrittayakamee et al. 2009. Simultaneous fitting of a potential-energy surface and its corresponding force fields using feedforward neural networks. *J. Chem. Phys* (2009).

[220] M Quade et al. 2018. Sparse identification of nonlinear dynamics for rapid model recovery. *Chaos* 28, 6 (2018), 063116.

[221] A Quarteroni et al. 2014. *Reduced order methods for modeling and computational reduction*. Vol. 9. Springer.

[222] P Raccuglia et al. 2016. Machine-learning-assisted materials discovery using failed experiments. *Nature* 533, 7601 (2016), 73–76.

[223] R Rai and CK Sahu. 2020. Driven by data or derived through physics? A Review of hybrid physics guided machine learning techniques with cyber-physical system (CPS) focus. *IEEE Access* 8 (2020), 71050–71073.

[224] M Raissi et al. 2019. Deep learning of vortex-induced vibrations. *J. Fluid Mech* (2019).

[225] M Raissi and GE Karniadakis. 2018. Hidden physics models: Machine learning of nonlinear partial differential equations. *J. Comput. Phys* 357 (2018), 125–141.

[226] M Raissi, P Perdikaris, and G Karniadakis. 2017. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv:1711.10561* (2017).

[227] M Raissi, P Perdikaris, and G Karniadakis. 2017. Physics informed deep learning (part II): Data-driven discovery of nonlinear partial differential equations. *arXiv:1711.10561* (2017).

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

[229] MM Rajabi and H Ketabchi. 2017. Uncertainty-based simulation-optimization using Gaussian process emulation: application to coastal groundwater management. *J. Hydrol* (2017).

[230] S Rasp, MS Pritchard, and P Gentine. 2018. Deep learning to represent subgrid processes in climate models. *PNAS* 115, 39 (2018), 9684–9689.

[231] JS Read et al. 2019. Process-guided deep learning predictions of lake water temperature. *Water Resour. Res.* (2019).

[232] M Reichstein et al. 2019. Deep learning and process understanding for data-driven Earth system science. *Nature* 566, 7743 (2019), 195–204.

[233] K Rudd and S Ferrari. 2015. A constrained integration (CINT) approach to solving partial differential equations using artificial neural networks. *Neurocomputing* (2015).

[234] SH Rudy et al. 2017. Data-driven discovery of partial differential equations. *Science Advances* (2017).

[235] Samuel H Rudy, J Nathan Kutz, and Steven L Brunton. 2019. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. *J. Comput. Phys* 396 (2019), 483–506.

[236] L Ruthotto and E Haber. 2018. Deep neural networks motivated by partial differential equations. *J. Math. Imaging Vis.* (2018).

[237] JJ Rutz et al. 2019. The atmospheric river tracking method intercomparison project (ARTMIP): quantifying uncertainties in atmospheric river climatology. *J. Geophys. Res. Atmos.* (2019).

[238] K Ryan, J Lengyel, and M Shatruk. 2018. Crystal structure prediction via deep learning. *J. Am. Chem. Soc* 140, 32 (2018), 10158–10168.

[239] M Sadoughi and C Hu. 2019. Physics-based convolutional neural network for fault diagnosis of rolling element bearings. *IEEE Sensors Journal* (2019).

[240] P Sadowski et al. 2016. Synergies Between Quantum Mechanics and Machine Learning in Reaction Prediction. *J. Chem. Inf. Model.* 56, 11 (2016), 2125–2128. https://doi.org/10.1021/acs.jcim.6b00351 arXiv:https://doi.org/10.1021/acs.jcim.6b00351 PMID: 27749058.

[241] O San and R Maulik. 2018. Machine learning closures for model order reduction of thermal fluids. *Appl. Math. Model.* (2018).

[242] O San and R Maulik. 2018. Neural network closures for nonlinear model order reduction. *Adv Comput Math* 44, 6 (2018), 1717–1750.

[243] GR Schleder et al. 2019. From DFT to machine learning: recent approaches to materials science–a review. *J. Phys. Materials* 2, 3 (2019), 032001.

[244] M Schmidt and H Lipson. 2009. Distilling free-form natural laws from experimental data. *Science* (2009).

[245] K Schütt et al. 2017. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *NeurIPS*. 991–1001.

[246] O Senouf et al. 2019. Self-supervised learning of inverse problem solvers in medical imaging. In *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data*. Springer.

[247] S Shah et al. 2018. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*. Springer, 621–635.

[248] V Shah et al. 2019. Encoding invariances in deep generative models. *arXiv:1906.01626* (2019).

[249] E Sharifi, B Saghafian, and R Steinacker. 2019. Downscaling satellite precipitation estimates with multiple linear regression, artificial neural networks, and spline interpolation techniques. *J. Geophys. Res. Atmos.* 124, 2 (2019), 789–805.

[250] AS Sharma, I Mezić, and BJ McKeon. 2016. Correspondence between Koopman mode decomposition, resolvent mode decomposition, and invariant solutions of the Navier-Stokes equations. *Phys. Rev. Fluids* 1, 3 (2016), 032402.

[251] R Sharma et al. 2018. Weakly-supervised deep learning of heat transport via physics informed loss. *arXiv:1807.11374* (2018).

[252] J Sirignano and K Spiliopoulos. 2018. DGM: A deep learning algorithm for solving partial differential equations. *J. Comput. Phys* 375 (2018), 1339–1364.

[253] D Solle et al. 2017. Between the poles of data-driven and mechanistic modeling for process operation. *Chemie Ingenieur Technik* (2017).

[254] PK Srivastava et al. 2013. Machine learning techniques for downscaling SMOS satellite soil moisture using MODIS land surface temperature for hydrological application. *Water Resour. Manag.* 27, 8 (2013), 3127–3144.

[255] P Sturmfels et al. 2018. A domain guided CNN architecture for predicting age from structural brain images. *arXiv:1808.04362* (2018).

[256] J Sun et al. 2020. A theory-guided deep-learning formulation and optimization of seismic waveform inversion. *Geophysics* 85, 2 (2020), R87–R99.

[257] DH Svendsen et al. 2017. Joint Gaussian processes for biophysical parameter retrieval. *IEEE Trans Geosci Remote Sens* 56, 3 (2017), 1718–1727.

[258] L Swiler et al. 2020. A Survey of Constrained Gaussian Process Regression: Approaches and Implementation Challenges. *arXiv:2006.09319* (2020).

[259] N Tajbakhsh et al. 2016. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Trans. Med. Imaging.* 35, 5 (2016), 1299–1312.

[260] N Takeishi, Y Kawahara, and T Yairi. 2017. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Adv. Neural Inf. Process. Syst.* 1130–1140.

[261] A Tanaka, A Tomiya, and K Hashimoto. 2021. *Deep Learning and Physics*. Springer Nature.

[262] K Tayal et al. 2020. Inverse problems, deep learning, and symmetry breaking. *arXiv:2003.09077* (2020).

[263] K Tayal et al. 2020. Unlocking inverse problems using deep learning: Breaking symmetries in phase retrieval. In *NeurIPS*.

[264] C Tesche et al. 2018. Coronary CT angiography–derived fractional flow reserve: machine learning algorithm versus computational fluid dynamics modeling. *Radiology* 288, 1 (2018), 64–72.

[265] FD Theurer, KA Voos, and WJ Miller. 1984. Instream water temperature model. *Div. Biol. Serv., Tech. Rep. FWS OBS* 84, 15 (1984), 11–42.

[266] ML Thompson and MA Kramer. 1994. Modeling chemical processes using prior knowledge and neural networks. *AIChE Journal* 40, 8 (1994), 1328–1340.

[267] J Tobin et al. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ IROS*. IEEE, 23–30.

[268] P Toth et al. 2019. Hamiltonian generative networks. *arXiv:1909.13789* (2019).

[269] RK Tripathy and I Bilionis. 2018. Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *J. Comput. Phys* (2018).

[270] S Udrescu and M Tegmark. 2020. AI Feynman: A physics-inspired method for symbolic regression. *Sci. Adv.* 6, 16 (2020), eaay2631.

[271] D Ulyanov, A Vedaldi, and V Lempitsky. 2018. Deep image prior. In *CVPR*.

[272] J Vamaraju and MK Sen. 2019. Unsupervised physics-based neural networks for seismic migration. *Interpretation* (2019).

[273] T Vandal et al. 2017. Deepsd: Generating high resolution climate change projections through single image super-resolution. In *SIGKDD '17*. 1663–1672.

[274] T Vandal et al. 2018. Quantifying uncertainty in discrete-continuous and skewed data with Bayesian deep learning. In *SIGKDD '18*. 2377–2386.

[275] C Varadharajan. 2021. *Using Machine Learning to Develop a Predictive Understanding of the Impacts of Extreme Water Cycle Perturbations on River Water Quality*. Technical Report. AI4ESP.

[276] PR Vlachas et al. 2018. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proc. R. Soc. A* (2018).

[277] L von Rueden et al. 2020. Informed machine learning–a taxonomy and survey of integrating knowledge into learning systems. *arXiv:1903.12394* (2020).

[278] ZY Wan et al. 2018. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PLoS one* (2018).

[279] JX Wang, JL Wu, and H Xiao. 2017. Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Phys. Rev. Fluids* 2, 3 (2017), 034603.

[280] L Wang, J Chen, and M Marathe. 2020. TDEFSI: Theory-guided Deep Learning-based Epidemic Forecasting with Synthetic Information. *ACM Trans. Spat. Algorithms Syst.* 6, 3 (2020), 1–39.

[281] R Wang, R Walters, and R Yu. 2020. Incorporating Symmetry into Deep Dynamics Models for Improved Generalization. *arXiv:2002.03061* (2020).

[282] S Wang, Y Teng, and P Perdikaris. 2021. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J Sci Comput* 43, 5 (2021), A3055–A3081.

[283] Z Wang et al. 2018. Successful leveraging of image processing and machine learning in seismic structural interpretation: A review. *The Leading Edge* 37, 6 (2018), 451–461.

[284] C Wehmeyer and F Noé. 2018. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *J. Chem. Phys* 148, 24 (2018), 241703.

[285] H Wei et al. 2018. Predicting the effective thermal conductivities of composite materials and porous media by machine learning methods. *Int. J. Heat Mass Transf.* 127 (2018), 908–916.

[286] JD Willard et al. 2021. Predicting Water Temperature Dynamics of Unmonitored Lakes With Meta-Transfer Learning. *Water Resour. Res.* 57, 7 (2021), e2021WR029579.

[287] CKI Williams and CE Rasmussen. 2006. *Gaussian processes for machine learning*. Vol. 2. MIT press Cambridge, MA.

[288] N Winovich, K Ramani, and G Lin. 2019. ConvPDE-UQ: Convolutional neural networks with quantified uncertainty for heterogeneous elliptic partial differential equations on varied domains. *J. Comput. Phys* 394 (2019), 263–279.

[289] H Wu et al. 2017. Variational Koopman models: slow collective variables and molecular kinetics from short off-equilibrium simulations. *J. Chem. Phys* 146, 15 (2017), 154104.

[290] JL Wu et al. 2016. Physics-informed machine learning for predictive turbulence modeling: A priori assessment of prediction confidence. *arXiv:1607.04563* (2016).

[291] JL Wu et al. 2019. Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. *J. Comput. Phys* (2019), 109209.

[292] JL Wu et al. 2020. Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. *J. Comput. Phys* (2020).

[293] JL Wu, H Xiao, and E Paterson. 2018. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Phys. Rev. Fluids* 3, 7 (2018), 074602.

[294] Y Wu et al. 2020. Inductive Graph Neural Networks for Spatiotemporal Kriging. *arXiv:2006.07527* (2020).

[295] D Xiao et al. 2019. A reduced order model for turbulent flows in the urban environment using machine learning. *Building and Environment* 148 (2019), 323–337.

[296] You Xie et al. 2018. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Trans. Graph* 37, 4 (2018), 1–15.

[297] TF Xu and AJ Valocchi. 2015. Data-driven methods to improve baseflow prediction of a regional groundwater model. *Comput. Geosci.* (2015).

[298] W Yan et al. 2011. Bayesian migration of Gaussian process regression for rapid process modeling and optimization. *Chem. Eng. J.* 166, 3 (2011), 1095–1103.

[299] L Yang et al. 2019. Highly-Ccalable, Physics-Informed GANs for Learning Solutions of Stochastic PDEs. In *2019 IEEE/ACM Deep Learning on Supercomputers*. IEEE.

[300] L Yang, X Meng, and G Karniadakis. 2020. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *arXiv:2003.06097* (2020).

[301] L Yang, D Zhang, and GE Karniadakis. 2018. Physics-informed generative adversarial networks for stochastic differential equations. *arXiv:1811.02033* (2018).

[302] T Yang et al. 2019. Evaluation and machine learning improvement of global hydrological model-based flood simulations. *Environ. Res. Lett* 14, 11 (2019), 114027.

[303] Y Yang and P Perdikaris. 2018. Physics-informed deep generative models. *arXiv:1812.03511* (2018).

[304] Y Yang and P Perdikaris. 2019. Adversarial uncertainty quantification in physics-informed neural networks. *J. Comput. Phys* (2019).

[305] K Yao et al. 2018. The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics. *Chem. Sci.* (Jan 2018).

[306] A Yazdani, M Raissi, and G Karniadakis. 2019. Systems biology informed deep learning for inferring parameters and hidden dynamics. *bioRxiv* (2019), 865063.

[307] E Yeung, S Kundu, and N Hodas. 2019. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. In *American Control Conference*. IEEE, 4832–4839.

[308] Y Zeng et al. 2019. Enforcing deterministic constraints on Generative Adversarial Networks for emulating physical systems. *arXiv:1911.06671* (2019).

[309] L Zepeda-Núñez, Y Chen, J Zhang, W Jia, L Zhang, and L Lin. 2019. Deep Density: circumventing the Kohn-Sham equations via symmetry preserving neural networks. *arXiv:1912.00775* (2019).

[310] L Zhang et al. 2018. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.* 120, 14 (2018), 143001.

[311] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. 2018. DeePCG: Constructing coarse-grained models via deep neural networks. *J. Chem. Phys* 149, 3 (2018), 034101.

[312] L Zhang, G Wang, and GB Giannakis. 2019. Real-time power system state estimation and forecasting via deep unrolled neural networks. *IEEE Trans. Signal Process* 67, 15 (2019), 4069–4077.

[313] R Zhang, Y Liu, and H Sun. 2019. Physics-guided Convolutional Neural Network (PhyCNN) for Data-driven Seismic Response Modeling. *arXiv:1909.08118* (2019).

[314] X Zhang et al. 2009. Estimating uncertainty of streamflow simulation using Bayesian neural networks. *Water Resour. Res.* (2009).

[315] Z Zhang et al. 2014. Facial landmark detection by deep multi-task learning. In *ECCV*. Springer.

[316] Q Zheng et al. 2019. Physics-informed semantic inpainting: Application to geostatistical modeling. *arXiv:1909.09459* (2019).

[317] YD Zhong, B Dey, and A Chakraborty. 2019. Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control. *arXiv:1909.12077* (2019).

[318] Y Zhu et al. 2019. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *J. Comput. Phys* (2019).

[319] Y Zhu and N Zabaras. 2018. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *J. Comput. Phys* (2018).

[320] N Zobeiry and KD Humfeld. 2021. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. *Eng. Appl. Artif. Intell.* 101 (2021), 104232.