Physics-Guided Deep Learning for Dynamical Systems

Rui (Ray) Wang
Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093
ruw020@ucsd.edu

Abstract

Modeling complex physical dynamics is a fundamental task in science and engineering. Traditional physics-based models are interpretable but rely on rigid assumptions. And the direct numerical approximation is usually computationally intensive, requiring significant computational resources and expertise. While deep learning (DL) provides novel alternatives for efficiently recognizing complex patterns and emulating nonlinear dynamics, it does not necessarily obey the governing laws of physical systems, nor do they generalize well across different systems. Thus, the study of physics-guided DL emerged and has gained great progress. It aims to take the best from both physics-based modeling and state-of-the-art DL models to better solve scientific problems. In this paper, we provide a structured overview of existing methodologies of integrating prior physical knowledge or physics-based modeling into DL and discuss the emerging opportunities.

1 Introduction

Modeling complex physics dynamics over a wide range of spatial and temporal scales is a fundamental task in a wide range of fields including, e.g., fluid dynamics, epidemiology, economics, and neuroscience. Physics as a discipline, has a long tradition of using first-principled models to describe spatiotemporal dynamics. The laws of physics have greatly improved our understanding of the physical world. Many physics laws are described by systems of highly nonlinear differential equations that have direct implications for understanding and predicting physical dynamics.

However, these equations are usually too complicated to be solvable. The current paradigm of numerical methods for solution approximation is purely physics-based: known physical laws encoded in systems of coupled differential equations are solved over space and time via numerical differentiation and integration schemes. However, these methods are tremendously computationally intensive, requiring significant computational resources and expertise. An alternative way is seeking simplified models that are based on some reasonable assumptions and can describe the dynamics to a satisfactory accuracy, such as Reynolds-averaged Navier-Stokes equations for turbulent flows and Euler equations for gas dynamics. But it is highly nontrivial to obtain a simplified model that can describe a phenomenon to a satisfactory accuracy. More importantly, for many complex dynamical phenomena, only partial knowledge of their dynamics is known.

Deep Learning (DL) provides efficient alternatives to learn high-dimensional spatiotemporal dynamics by bypassing numerical approximation. Recent works have shown that DL can generate realistic predictions and significantly accelerate the simulation of physical dynamics relative to numerical solvers, from turbulence modeling to weather prediction. This opens up new opportunities at the intersection of DL and physical sciences. Despite the tremendous progress, purely data-driven DL models still adhere to the fundamental rules of statistical inference. Without explicit constraints, DL models are prone to make physically implausible forecasts, violating the...
governing laws of physical systems. Additionally, DL models often struggle with generalization: models trained on one dataset cannot adapt properly to unseen scenarios with different distributions. Distribution shift occurs not only because the dynamics are nonstationary and nonlinear, but also due to the changes in system parameters, such as initial and boundary conditions. [125].

Neither an DL-only nor physics-only approach can be considered sufficient for learning complex scientific problems. Therefore, there is a growing need for integrating traditional physics-based approaches with DL models so that we can take the best of both types of approaches. While Physics-guided DL is an emerging area, there is already a vast amount of work on this topic [132; 50; 14; 67; 69; 104]. Physics-guided DL offers a set of tools to blend these physical concepts such as differential equations and symmetry with deep neural networks. On one hand, these hybrid models offer great computational benefits over traditional numerical solvers. On the other hand, these models impose appropriate inductive biases on the DL models, leading to accurate simulation, scientifically valid predictions, reduced sample complexity, and guaranteed improvement in generalization.

The goal of this paper is to provide structured overview of existing methodologies of incorporating prior physical knowledge into DL models. The paper is organized as follow. Section 2 describes five objectives of physics-guided DL. Section 3 outlines four different categories of existing physics-guided DL models including (1) physics-guided loss function and regularization, (2) physics-guided design of architecture, (3) hybrid physics-DL models, (4) invariant and equivariant DL models. Each is further categorized based on either application or method type, and (1), (2), and (4) will lead with a detailed review of our recent work. Section 4 summarizes the paper and discusses the emerging opportunities of physics-guided DL for future research.

2 Objectives of Physics-Guided Deep Learning

This section provides a brief overview of the objectives of physics-guided DL. By incorporating physical principles, governing laws, and domain knowledge into DL models, the rapidly growing field of physics-guided DL seeks to:

2.1 Accelerate Data Simulation.

Simulation modeling is an important method of analyzing real-world problems, which is easily verified, communicated, and understood. It provides valuable solutions by giving clear insights into complex physical systems. Traditional physics-based approaches often rely on running numerical simulations: known physical laws encoded in systems of coupled differential equations are solved over space and time via numerical differentiation and integration schemes. These methods require significant computational resources and expertise. [51; 52; 80; 55; 91; 106].

Recently, DL has demonstrated great success in the automation, acceleration, and streamlining of highly compute-intensive workflows for science [104; 119; 63]. Deep dynamics models can directly approximate high-dimensional spatiotemporal dynamics by directly forecasting the future states and bypassing numerical integration [123; 126; 31; 83; 103; 109; 127]. For instance, generative networks, such as generative adversarial network (GAN), can generate instantaneous high-resolution flow fields that are statistically similar to those of direct numerical simulation [62; 136]. The computer graphics community has also investigated using DL to speed up numerical simulations for generating realistic animations of fluids such as water and smoke [61; 119; 131].

2.2 Build Physically Consistent and Scientifically Sound DL Models.

Despite the tremendous progress of DL for science (e.g., atmospheric science [104], computational biology [2], material science [16], quantum chemistry [112]), it remains a grand challenge to incorporate physical principles in a systematic manner to the design, training, and inference of such models. Purely data-driven DL models still adhere to the fundamental rules of statistical inference. Without explicit constraints, DL models are prone to make physically implausible predictions, violating the governing laws of physical systems.

Thus, to build physically consistent and scientifically sound predictive models, we need to use known physical knowledge to guide DL models to learn the correct underlying dynamics instead of simply fitting the observed data. For instance, [59; 56; 135; 11] improve the physical and statistical...
consistency of DL models by explicitly regularising the loss function with physical constraints. Hybrid DL models, e.g., \[83; 6; 20\] integrate differential equations in DL for temporal dynamics forecasting and achieve promising performance. \[79\] and \[40\] studied tensor invariant neural networks that can learn the Reynolds stress tensor while preserving Galilean invariance. \[123\] proposed a hybrid model that physically consistent with Direct Numerical Simulation by marrying RANS-LES coupling method and custom-designed U-net. \[46; 27\] build models upon Hamiltonian and Lagrangian mechanics that respect conservation laws.

### 2.3 Improve the generalizability of DL models

DL models often struggle with generalization: models trained on one dataset cannot adapt properly to unseen scenarios with distributional shifts that may naturally occur in dynamical systems \[66; 3; 125\]. In addition, most current approaches are still trained to model a specific system. Thus, it is imperative to develop generalizable DL models that can learn and generalize well across systems with various parameter domains, initial and boundary conditions.

Prior physics knowledge can be considered as inductive bias that can put a prior distribution on model class and shrink the model parameter search space. With the guide of physics knowledge, DL models can better capture the generalizable dynamics from the data that are consistent with physical laws, and hence be able to generalize to different systems or outside of the training domain.

For instance, embedding symmetries into DL models is one way to improve generalization, which we will discuss in detail in Section 3.4. For example, \[126\] demonstrated that encoding of rotation, scaling and uniform motion symmetries into DL models greatly improves the generalization on forecasting turbulence. \[78\] defines a tensor basis neural network to embed the rotational invariance for improved prediction accuracy. There are many other ways to improve the generalization of DL models by incorporating other physical knowledge. \[127\] proposed a meta-learning approach that can generalize across heterogeneous domains by having an encoder which infers the system parameters. \[39\] encodes Lyapunov stability into an autoencoder model for predicting fluid flow and sea surface temperature. They show improved generalizability and reduced prediction uncertainty for neural nets that preserve Lyapunov stability.

### 2.4 Solve Partial Differential Equations

Although the governing equations of many physical systems are known, finding approximate solutions using numerical algorithms and computers is still prohibitively expensive. DL can greatly reduce the computation. For example, \[84; 85\] designed two PDE-Net that can accurately predict the dynamical behavior of data and have the potential to reveal the underlying PDE model that drives the observed data. The most common approach is that deep neural networks can be directly used to approximate the solution of complex coupled differential equations and trained to respect given physical laws. This approach has shown success in approximating a variety of PDEs \[101; 100; 18; 48\]. However, poor generalization to the unseen domain and slow convergence in training have limited its applicability to many complex physics problems.

### 2.5 Search for Physical laws

One of main themes of science is the search for fundamental laws of practical problems \[36\]. When the governing equations of dynamical systems are known, they allow for accurate mathematical modeling, robust forecasting, and increased interpretability. However, dynamical systems in many fields, such as epidemiology, finance, and neuroscience, have no formal analytical descriptions.

Current ML methods for discovery of governing equations is to select from a large set of possible mathematical terms. \[111\] applied symbolic regression to differences between computed derivatives and analytic derivatives to determine the underlying dynamical systems. The seminal work by \[15\] proposed to find ordinary differential equations by creating a dictionary of possible terms and applying sparse regression to select appropriate terms. \[70; 105\] further extend this method by using neural networks to construct the dictionary of functions. \[17\] contributed to this trend by introducing an efficient first-order conditional gradient algorithm for solving the optimization problem of finding the best sparse fit to observational data in a large library of potential nonlinear models. Thus, research in data-driven methods based on DL for discovering physics laws is still quite preliminary.
3 Physics-Guided DL Methods

This section will explore four classes of methodologies to integrate principles of physics-based modeling with DL: (1) physics-guided loss function and regularization, (2) physics-guided design of architecture, (3) hybrid physics-DL models, (4) invariant and equivariant DL models.

3.1 Physics-Guided Loss Functions and Regularization

Complex physics dynamics occur over a wide range of spatial and temporal scales. Standard DL models simply fit the observed data while fail to learn the correct underlying dynamics, thus leading to low physical consistency and poor generalizability. One of simplest and most widely used approaches to solve this problem is via incorporating physics into loss functions (regularization). Physics-guided loss functions (regularization) can assist DL models to capture correct and generalizable dynamic patterns that are consistent with physical laws. Furthermore, the loss functions constrained by physics laws can reduce the possible search space of parameters. This approach is sometimes referred to as imposing “soft” constraints, which will be contrasted with imposing “hard” constraints (physics-guided architecture) in the next section. In this chapter, we will start with a simple example of physics-guided loss from our previous work, and categorize this type of methods based on their objectives.

**Case Study: Divergence Free Regularizer** [124] studied the task of forecasting two-dimensional raw velocity fields of an incompressible turbulent flow, as shown in Figure 1. Incompressible fluid flows have zero divergence everywhere. The models are trained to make forward predictions given the historic frames. In the proposed physics-guided DL model, TF-Net, apart from the regular mean square error (MSE) loss between target and predictions, it includes additional a divergence-free regularizer to reduce the divergence of turbulent flow predictions during training, as shown below.

\[
\text{Loss} = \text{MSE}(\hat{w}, w) + \lambda \| \nabla \cdot \hat{w} \|^2
\]

where \( w \) is velocity and \( \lambda \) is a hyper-parameter that controls the weight of the divergence free regularizer. By generalizing the divergence of predictions, the proposed model is able to obey the mass conservation law and generate predictions with almost zero divergence. Thus, generally speaking, a physics-guided loss function includes an additional physics-based loss that ensures consistency with physical laws and is controlled by a hyper-parameter.

However, there might be a subtle trade-off between prediction errors and physics-guided regularizers. For instance, although constraining the TF-Net with the divergence-free regularizer can reduce the divergence of the predictions, too much constraint has the side effect of smoothing out the small eddies, which may result in a larger prediction error.

Figure 1: A snapshot of the velocity norm (\( \sqrt{v_x^2 + v_y^2} \)) fields of the 2D Rayleigh-Bénard convection flow [23]. The spatial resolution is 1792 x 256 pixels.

**Improving Prediction** Like the divergence-free loss in the case study, physics-guided loss functions or regularization have shown great success in improving the prediction accuracy and physical consistency of DL models. [59] used neural nets to model lake temperature at different times and different depths. They ensure that the predictions are physically meaningful by regularizing that the denser water predictions are at lower depths than predictions of less dense water. [56] further introduced a loss term that ensures thermal energy conservation between incoming and outgoing heat fluxes for modeling lake temperature. [12] designed conservation layers to strictly enforce conservation laws in their NN emulator of atmospheric convection. [10] introduced a more systematic way of enforcing nonlinear analytic constraints in neural networks via constraints in the loss function. [141] incorporated the loss of atomic force and atomic energy into neural nets for improved accuracy.
of simulating molecular dynamics. [81] proposed a novel multifidelity physics-constrained neural network for material modeling, in which the neural net was constrained by the losses caused by the violations of the model, initial conditions, and boundary conditions. [34] proposed a novel paradigm for spatiotemporal dynamics forecasting that performs spatiotemporal disentanglement using the functional variable separation. The specific-designed time invariance and regression loss functions ensure the separation of spatial and temporal information.

The Hamiltonian of a system is the sum of the kinetic energies of all particles, plus the potential energy of the particles associated with the system. [46] proposed Hamiltonian Neural Nets (HNN) that parameterizes a Hamiltonian with a neural network and then learn it directly from data. The conservation of desired quantities is constrained in the loss function during training. The proposed HNN has shown success in predicting mass-spring and pendulum systems. Lagrangian mechanics models the energies in a system rather than the forces. [27] proposed Lagrangian Neural Nets (LNN) used a neural network to parameterize the Lagrangian function that is the kinetic energy (energy of motion) minus the potential energy. They trained the neural network with the Euler-Lagrange constraint loss functions such that it can learns to approximately conserve the total energy of the system. [42] further simplify the HNN and LNN via explicit constraints. [72] further introduced a meta-learning approach in HNN to find the structure of the Hamiltonian that can be adapted quickly to a new instance of a physical system. [43] benchmark recent energy-conserving neural network models based on Lagrangian/Hamiltonian dynamics on four different physical systems.

Data Generation There is an increasing interest in constraining deep generative models of simulating physical dynamics with physics-guided loss for better physical consistency. [135] enforced the constraints of covariance into standard Generative Adversarial Networks (GAN) via statistical regularization, which leads to faster training and better physical consistency compared with standard GAN. [136] proposed tempoGAN for super-resolution fluid flow, in which an advection difference loss is used to enforce the temporal coherence of fluid simulation. [130] modified ESRGAN, which is a conditional GAN designed for super-resolution, by replacing the adversarial loss with a loss that penalizes errors in the energy spectrum between the generated images and the ground truth data. Conditional GAN is applied to emulating numeric hydroclimate models in [87]. The simulation performance is further improved by penalizing the snow water equivalent via loss function. [61] proposed a generative model to simulate fluid flows, in which a novel stream function-based loss function is designed to ensure divergence-free motion for incompressible flows. [44] proposed a physics-informed convolutional model for flow super-resolution, in which physical consistency of the generated high-resolution flow fields is improved by minimizing the residuals of Navier-Stokes equations.

Solving Differential Equations Physics-guided loss functions have also been widely used to solve complex differential equations. [100, 102, 101, 1, 115] directly approximate the solution differential equations with fully connected neural networks given space coordinates and time stamps as input. They used automatic differentiation to differentiate the neural networks to calculate the first or second order derivatives with respect to their input coordinates and time. Then the governing equation can be enforced in the loss function. Even though the prediction accuracy increases as the number of sampled points, this type of method cannot generalize to the unseen domain [66, 3]. [43] proposed a convolutional encoder-decoder architecture as well as a conditional flow-based generative model for surrogate modeling based on the loss of PDE constraints. [99] utilizes DL simultaneously with mechanistic modeling, where neural nets are used to approximate high-order or unknown parts of the differential equations from the observed data.

3.2 Physics-Guided Design of Architecture

While incorporating physical constraints via loss functions has been demonstrated effective for improving performance of DL models, DL is still used as a black box in most cases. Since the modularity of neural networks offers opportunities for the design of novel neurons, layers or blocks that encode specific physical properties. Research into physics-guided neural architectures is getting increasingly active. The advantage of physics-guided NN architectures is that they can be used to impose “hard” constraints that are strictly enforced, compared to the “soft” constraints described in the previous section. The “soft” constraints are much easier to design and use than hard constraints, yet not required to be strictly satisfied. And their relative importance to the standard MSE loss is
tunable, so there are no generalizability guarantees. DL models with physics-guided architectures have theoretically guaranteed properties, and hence are more interpretable and generalizable.

Figure 2: Turbulent Flow Net: three identical encoders to learn the transformations of the three components of different scales, and one shared decoder that learns the interactions among these three components to generate the predicted 2D velocity field at the next instant. Each encoder-decoder pair can be viewed as a U-net and the aggregation is weighted summation.

**Case Study: Turbulent-Flow Net** [123] proposed Turbulent-Flow Net (TF-Net), as shown in Figure 2, which is a physics-guided DL model for turbulent flow prediction. It applies scale separation to model different ranges of scales of the turbulent flow individually. Computational techniques are at the core of present-day turbulence investigations. Direct Numerical Simulation (DNS) are accurate but not computationally feasible for practical applications. Great emphasis was placed on the alternative approaches including Large Eddy Simulation (LES) and Reynolds-averaged Navier-Stokes (RANS). Both resort to resolving large scales while modeling small scales, using various averaging techniques and/or low-pass filtering of the governing equations [91; 106].

Building upon a promising computational fluid dynamics technique, the RANS-LES coupling approach [38] that combines both RANS and LES approaches in order to take advantage of both methods, TF-Net replaces a priori spectral filters with trainable convolutional layers. The turbulent flow is decomposed into three components, each of which is approximated by a specialized U-net to preserve the multiscale properties of the flow. A shared decoder learns the interactions among these three components and generate the final prediction. The motivation for this design is to explicitly guide the ML model to learn the nonlinear dynamics of large-scale and SGS motions as relevant to the task of spatio-temporal prediction.

Besides RMSE, physically relevant metrics including divergence and energy spectrum are used to evaluate the performance of the models’ prediction. Figure 3 shows TF-Net consistently outperforms all baselines on physically relevant metrics (Divergence and Energy Spectrum) as well as average time to produce single velocity field. Constraining it with divergence free regularizer that we describe in the case study of last chapter can further reduce the RMSE and Divergence. Figure 4 shows the ground truth and predicted velocity along x direction by TF-Net and three best baselines. We see that the predictions by our TF-Net model are the closest to the target based on the shape and frequency of the motions. Thus, TF-Net is able to generate both accurate and physically meaningful predictions of the velocity fields that preserve critical quantities of relevance.

**Convolutional Models** Convolutional architecture remains dominant most tasks in computer vision, such as objection, image classification, and video prediction. Because of their efficiency and desired inductive biases, such as locality and translation equivariance, convolution neural nets have also been widely applied to emulating and predicting complex spatiotemporal physical dynamics. Researchers tend to bake more desired physical properties into the design of convolutional models.

For example, [57] designed a PDE layer within CNNs to strictly enforce PDE constraints for super-resolution of turbulence. [28] modified the LSTM units to introduce an intermediate variable to strictly preserve monotonicity in a convolutional auto-encoder model for lake temperature. [22]
proposed a physics-guided convolutional model, PhyDNN, which uses physics guided structural priors and physics-guided aggregate supervision for modeling the drag forces acting on each particle in a computational fluid dynamics-discrete element Method. Designed HybridNet for dynamics predictions that combines data-driven learning with ConvLSTM to predict external forces with model-driven computation with CeNN for system dynamics. HybridNet is shown higher accuracy on the tasks of forecasting heat convection-diffusion and fluid dynamics. Proposed to combine deep learning and a differentiable PDE solver for understanding and controlling complex nonlinear physical systems over a long time horizon. Proposed continuous-filter convolutional layers for modeling quantum interactions. The convolutional kernel is parametrized by neural nets that take relative positions between any two points as input. They obtained a joint model for the total energy and interatomic forces that follows fundamental quantumchemical principles.

**Graph Neural Networks**  Standard convolutional neural nets only operate on the regular or uniform mesh such as images. Graph neural networks moves beyond data on the regular grid and mesh towards modeling objects with arbitrary positions. For instance, designed a deep encoder-processor-decoder graphic architecture for simulating fluid dynamics under lagrangian description. The rich physical states are represented by graphs of interacting particles, and complex interactions are approximated by learned message-passing among nodes. Message-passing in world-space can estimate external dynamics, not captured by the mesh-space interactions, such as contact and collision. Utilized similar graphic networks to learn mesh-based simulation. The authors directly construct graphs on the irregular meshes used in the numerical simulation. In addition, they proposed a adaptive remeshing algorithm that allows people to accurately predict dynamics at both large and small scales.

Proposed a Neural Operator approach that learns the mapping between function spaces, and is invariant to different approximations and grids. More specifically, it used the message passing graph network to learn the Green’s function from the data and then the learned Green’s function can used to compute the final solution of PDEs. Further extended it to Fourier Neural Operator by replacing the kernel integral operator with a convolution operator defined in Fourier space.
In [108], graph networks were also used to represent, learn, and infer robotics systems, bodies and joints. [76] proposed to learn compositional Koopman operators, using graph neural networks to encode the state into object centric embeddings and using a block-wise linear transition matrix to regularize the shared structure across objects. [54] presented a physics-guided encoder-decoder video prediction model. The estimated velocity from the encoder is fed to an differentiable physics simulator in the middle, which generates future predictions of object positions. The output of this simulator is fed to a co-ordinate consistent graphic to render the corresponding output image.

**Fully Connected Neural Networks** The interplay between Koopman theory [65] and DL has gained increasing popularity in the last few years. Koopman theory shows that it is possible to represent a nonlinear dynamical system in terms of an infinite-dimensional linear operator acting on a Hilbert space of measurement functions of the system state. This Koopman operator is linear, and its spectral decomposition completely characterizes the behavior of a nonlinear system.

An approximation of the Koopman operator can be computed via the Dynamic Mode Decomposition algorithm [110] but we need to prepare nonlinear observables manually according to the underlying dynamics, which is not always possible since we usually do not have any prior knowledge about them. Thus, [137] and [117] proposed novel methods that use fully connected neural nets to directly learn a dictionary of nonlinear observables from data that spans a Koopman invariant subspace, and outperformed state-of-the-art methods. [86] further generalize fully connected neural nets for learning the Koopman operator to systems with continuous spectra. [7] designed an autoencoder architecture based on Koopman theory to forecast physical processes, in which the encoder can map the data to a finite Koopman-invariant subspace. In the latent space, the consistency of both the forward and backward systems is ensured, while other models only consider the forward system. The proposed model performs well on noisy data and for long time predictions.

### 3.3 Hybrid Physics-DL Model

Another efficient and effective way to combine physics and neural nets is to combine complete physics-based models with DL models.

**Case Study: DeepGLEAM** One of traditional and popular ways is residual modeling, where DL learns to predict the errors or residuals made by physics-based models. The key is to learn the bias of physics-based models and correct it with the help of DL models [43] [118].

Figure 5: Architecture of DeepGLEAM: a DCRNN is trained with the residual between reported incident death and predictions from GLEAM.

A representative example is DeepGLEAM [134], as shown in Figure 5. DeepGLEAM combines a mechanistic stochastic simulation model GLEAM with deep learning. It uses a Diffusion Convolu-
tional RNN [75] to learn the correction terms from GLEAM, which leads to improved performance. Figure 6 shows one week ahead COVID-19 death count predictions by GLEAM, DCRNN, and DeepGLEAM, and it can be observed that DeepGLEAM can outperform purely mechanistic models and purely deep learning models.

Figure 6: One week ahead COVID-19 death count forecasts visualization in California. Comparison shown for GLEAM, DeepGLEAM and DCRNN.

Residual Modeling Apart from DeepGLEAM, [30] combines graph neural nets with a CFD simulator run on coarse mesh to generate high-resolution fluid flow prediction. CNNs are used to correct the velocity field from the numerical solver on a coarse grid in [63]. [90] utilized neural networks to do subgrid modelling for the LES of two-dimensional turbulence. In [107], a neural network closure model is implemented in the ROM framework to compensate the errors due to the model reduction. [58] proposed DR-RNN that is trained to find the residual minimizer of numerically discretized ODEs or PDEs. They showed that DR-RNN can greatly reduce both computational cost and time discretization error of the reduced order modeling framework. [138] introduced the APHYNITY framework that can efficiently augment approximate physical models with deep data-driven networks. A key feature is being able to decompose the problem in such a way that the data-driven model only models what cannot be captured by the physical model.

Intermediate Variable Modeling DL models can be used to replace one or more components of physics-based models that are difficult to compute or unknown. For example, [119] replaced the numerical pressure solver with convolution networks in the procedure of Eulerian fluid simulation, and the obtained results are realistic and showed good generalization properties. [95] proposed to use neural nets to reconstruct the model corrections in terms of variables that appear in the closure model. [32] applied a U-net to estimate the velocity field given the historical temperature frames, then used the estimated velocity to forecast the sea surface temperature based on the closed-form solution of the advection-diffusion equation. [88] combined the high-dimensional model representation with NNs to build multidimensional potential, in which NNs are used to represent HDMR component functions that minimize the error mode term by mode term.

Neural Differential Equations A growing volume of literature has also been exploring neural differential equations. [21] developed a continuous depth NN for solving ordinary differential equations, Neural ODE. They changed the traditionally discretized neuron layer depths into continuous equivalents such that the derivative of the hidden state can be parameterized using a neural network. The output of the network is computed using a blackbox differential equation solver. This allows for increased computational efficiency due to the simplification of the backpropagation step of training. Neural ODEs have been widely used for time-series modeling, supervised learning, and density estimation. [35] introduced Augmented Neural ODE that is more expressive, empirically more stable and more lower computationally efficient than Neural ODEs. More importantly, it can learn functions that Neural ODEs cannot represent. [82] further extended this idea of continuous neural nets to graph convolutions, and proposed Graph Neural ODE. [82] proposed Neural Stochastic Differential Equation (Neural SDE), which models stochastic noise injection by stochastic differential equations. They demonstrated that incorporating the noise injection regularization mechanism to the continuous neural network can reduce overfitting and achieve lower generalization error.
3.4 Invariant and Equivariant DL Models

Developing neural nets that preserve symmetries has been a fundamental task in image recognition [26, 128, 24, 22, 74, 64, 8, 133, 25, 41, 129, 33, 45]. In physics, there is also a deep connection between symmetries and physics. Noether’s law gives a correspondence between conserved quantities and groups of symmetries. For instance, translation symmetry corresponds to the conservation of energy and rotation symmetry corresponds to the conservation of angular momentum. By building a neural network which inherently respects a given symmetry, we thus make conservation of the associated quantity more likely and consequently the model’s prediction more physically accurate. Furthermore, by designing a model that is inherently equivariant to transformations of its inputs, we can guarantee that our model generalizes automatically across these transformations, making it robust to distributional shift.

An **group of symmetries** or simply **group** consists of a set $G$ together with an associative composition map $\circ : G \times G \rightarrow G$. The composition map has an identity $1 \in G$ and composition with any element of $G$ is required to be invertible. A group $G$ has an **action** on a set $S$ if there is an action map $\cdot : G \times S \rightarrow S$ which is compatible with the composition law. We say further that $S$ is a **$G$-representation** if the set $S$ is a vector space and the group acts on $S$ by linear transformations.

Let $f : X \rightarrow Y$ be a function and $G$ be a group. Assume $G$ acts on $X$ and $Y$. The function $f$ is **$G$-equivariant** if $f(gx) = gf(x)$ for all $x \in X$ and $g \in G$. The function $f$ is **$G$-invariant** if $f(gx) = f(x)$ for all $x \in X$ and $g \in G$. See Figure 7 for an illustration of an equivariant function.

![Figure 7](image)

**Figure 7:** Left: Illustration of equivariance: $f(x) = 2x$ w.r.t $T = \text{rot}(\pi/4)$; Right: Detail of the DyAd encoder. The conv3D layers are shift equivariant and global mean pooling is shift invariant. The network is approximately invariant to spatial and temporal shifts.

**Case Study: Equivariant Deep Dynamics Models** [126] studied the symmetries of fluid dynamics. The Navier-Stokes equations are invariant under the following five different transformations. Individually, each of these types of transformations generates a group of symmetries of the system.

- **Space translation:** $T^{sp}_c w(x, t) = w(x - c, t)$, $c \in \mathbb{R}^2$,
- **Time translation:** $T^{time}_\tau w(x, t) = w(x, t - \tau)$, $\tau \in \mathbb{R}$,
- **Uniform motion:** $T^{um}_c w(x, t) = w(x, t) + c$, $c \in \mathbb{R}^2$,
- **Rotation/Reflection:** $T^{rot}_R w(x, t) = Rw(R^{-1}x, t)$, $R \in O(2)$,
- **Scaling:** $T^{sc}_\lambda w(x, t) = \lambda w(\lambda x, \lambda^2 t)$, $\lambda \in \mathbb{R}_{>0}$.

We tailored different methods for incorporating each symmetry into CNNs for spatiotemporal dynamics forecasting. CNNs are time translation-equivariant when we use it in an autoregressive manner. Convolutions are also naturally space translation equivariant. Scale equivariance in dynamics is unique as the physical law dictates the scaling of magnitude, space and time simultaneously. To achieve this, they replaced the standard convolution layers with group correlation layers over the
group $G = (\mathbb{R}^\times_0, \cdot) \ltimes (\mathbb{R}^2, +)$ of both scaling and translations. The $G$-correlation upgrades this operation by both translating and scaling the kernel relative to the input,

$$v(p, s, \mu) = \sum_{\lambda \in \mathbb{R}^\times_0, t \in \mathbb{R}, q \in \mathbb{Z}^2} \mu v(p + \mu q, \mu^2 t, \lambda) K(q, s, t, \lambda),$$  \hspace{1cm} (2)

where $s$ and $t$ denote the indices of output and input channels respectively. We add an axis to the tensors corresponding the scale factor $\mu$.

And the rotational symmetry was modeled using SO(2)-equivariant convolutions and activations within the E(2)-CNN framework [128]. Uniform motion transformation is adding a constant vector field to the vector field, which is part of Galilean invariance and relevant to all non-relativistic physics modeling. And the uniform motion equivariance is enforced by conjugating the model with shifted input distribution. Basically, or each sliding local block in each convolutional layer, we shift the mean of input tensor to zero and shift the output back after convolution and activation function per sample. In other words, if the input is $\mathcal{P}_{b \times d_{in} \times s \times s}$ and the output is $\mathcal{Q}_{b \times d_{out}} = \sigma(\mathcal{P} \cdot K)$ for one sliding local block, where $b$ is batch size, $d$ is number of channels, $s$ is the kernel size, and $K$ is the kernel, then

$$\mu_i = \text{Mean}_{jkl}(\mathcal{P}_{ijkl}); \quad \mathcal{P}_{ijkl} \mapsto \mathcal{P}_{ijkl} - \mu_i; \quad \mathcal{Q}_{ij} \mapsto \mathcal{Q}_{ij} + \mu_i.$$  \hspace{1cm} (3)

This will allow the convolution layer to be equivariant with respect to uniform motion. If the input is a vector field, we apply this operation to each element.

The DL models used are ResNet and U-Net, and their equivariant counterparts. Spatiotemporal prediction is done autoregressively. Standard RMSE and a RMSE computed on the energy spectra are used to measure performance. The models are tested on Rayleigh-Bénard convection (RBC) and reanalysis ocean current velocity data. For RBC, the test sets have random transformations from the relevant symmetry groups applied to each sample. This mimics real-world data in which each sample has an unknown reference frame. For ocean data, tests are also performed on different time ranges and different domains from the training set, representing distributional shifts. Figure 8 shows the equivariant models perform significantly better than their non-equivariant counterparts on both simulated RBC data and reanalysis ocean currents. They also show equivariant models also achieve much lower energy spectrum errors.

**Figure 8:** Top: The ground truth and the predicted velocity norm fields $\|w\|_2$ of RBC at time step 1, 5, and 10 by the ResNet and four Equ-ResNets on four test samples applied with random uniform motion, magnitude, rotation and scaling transformations respectively. The first column is the target, the second is ResNet predictions, and the third is predictions by Equ-ResNets. Bottom: The ground truth and predicted velocity norm fields of ocean currents by ResNet and four Equ-ResNets on the test set.

**Fluid Dynamics** As shown in the case study, there is a deep connection between fluid dynamics and symmetries. [127] utilized an encoder capable of extracting the time-invariant and translation-invariant part of a dynamical system, which then is used to guide the main forecaster to generate
accurate predictions across heterogeneous domains. Time-invariance is achieved by using 3D convolution and time-shift invariant loss. [78] designed a tensor basis neural network that embeds the fundamental principle of rotational invariance into turbulence modeling for improved prediction accuracy. It added a final higher-order multiplicative layer in NN to ensure the prediction lies on a rotationally invariant tensor basis. In [89], weights and biases of neurons are constrained so that predictions from the NN are guaranteed to preserve even/odd symmetry and energy conservation.

**Molecular Dynamics** Since the calculation in molecular dynamics is computationally expensive, DL has been widely applied to molecular simulation [94]. And symmetry is quite pervasive in molecular physics. For instance, the potential energy of a molecule is invariant to rotation, but the force is equivariant because it rotates in the same way as its coordinates. [112] proposed SchNet, which is a continuous convolution framework that generalizes the CNN approach to continuous convolutions between particles at arbitrary positions. Continuous convolution kernels are generated by dense neural networks that operate on the interatomic distances, which ensures rotational and translation invariance of the energy. [122] computed the force by directly computing the derivatives of the predicted energy with respect to the coordinates to achieve the equivariance of the force. Permutation invariance also exists in molecular dynamics. For instance, quantum mechanical energies are invariant if we exchange the labels of identical atoms. [9] ensures the energy permutation invariance by representing the total energy of the system as a sum of atomic contributions.

[5] designed Cormorant, a rotationally covariant neural network architecture for learning the behavior and properties of complex many-body physical systems. Cormorant achieves promising results in learning molecular potential energy surfaces on the MD-17 dataset and learning geometric, energetic, electronic, and thermodynamic properties of molecules on the GDB-9 dataset. [114] proposed a model for autoregressive generation of 3D molecular structures with reinforcement learning. The method uses equivariant state representations for autoregressive generation, built largely from Cormorant, and integrating such representations within an existing actor-critic RL generation framework. [140] proposed an end-to-end modeling framework that preserves all natural symmetries of a molecular system using an embedding procedure that maps the input to symmetry-preserving components.

**Other Applications** In a traffic forecasting application, [121] proposed a novel model, Equivariant Continuous COnvolution (ECCO) that uses rotationally equivariant continuous convolutions to embed the symmetries of the system for improved trajectory prediction. The rotational equivariance is achieved by a weight sharing scheme within kernels in polar coordinates. ECCO achieves superior performance to baselines on two real-world trajectory prediction datasets, Argoverse and TrajNet++. Invariance can also be embeded into generative models, [113] encodes known invariance by utilizing an invariance loss function for training the generator.

4 **Discussion**

In this paper, we systematically review the recent progress in physics-guided DL methods. By being constrained with physics-informed loss regularizers, equipped with physics-based models, leveraged with physics-guided design or encoded with symmetry, the DL models can achieve better physical consistency, higher accuracy, increased data efficiency, improved generalization and greater interpretability. This survey also uncovers emerging opportunities of learning physical dynamics with deep learning for future study.

**Improving Generalization** Generalization is a fundamental problem in machine learning. Most DL models for dynamics modeling still struggle with generalization and are still trained to model a specific system. Based on our review in the last section, we see that incorporating prior physics knowledge can guide DL models to better learn complex patterns that are consistent with the physics laws from data, thus more robust and generalizable to unseen scenarios. Additionally, this survey focuses on how prior physics knowledge can help DL to learn physical dynamics. Actually, some advanced learning strategies in machine learning can also contribute to improving generalization. For example, in [127], we proposed a model-based meta-learning method called DyAd which can generalize across heterogeneous domains of real-world ocean dynamics. However, a truely trustworthy and reliable model for learning physical dynamics should be able to generalize across systems with various
parameters, external forces, or boundary conditions while preserve high accuracy. Thus, further research into reliable and generalizable physics-guided DL is needed.

**Theoretical Analysis**  The research into the theoretical analysis of learning nonstationary and chaotic dynamics with DL is lacking. Current learning theory of DL is based on the typical assumption of both training and test data being identically and independently distributed (i.i.d.) samples from some unknown distribution \( [139, 73, 93] \). However, this assumption does not hold for most dynamical systems, where observations at different times and locations may be highly correlated. \( [125] \) empirically showed that DL models fail to generalize under shifted distributions in both the data and parameter domains that naturally happens in dynamical systems. \( [68] \) provided the first generalization guarantees for time series forecasting with sequence-to-sequence models. The derived upper bound is expressed in terms of measures of non-stationarity and correlation strength as well as the Rademacher complexity. To better understand the performance of DL on learning physical dynamics, we need to derive a generalization bound expressed in terms of the characteristics of the dynamics, such as the order and dimensions of the governing equations. Theoretical study can inspire research into developing generalizable and reliable models for learning dynamical systems.

**Control and Design**  This survey primarily focuses on how DL can be used to model and predict complex physical dynamics. Next promising step would be, given the dynamics, how to design the environment to control it. For instance, automated computational fluid dynamics (CFD) analysis and control theory have been widely applied to aircraft design \( [37] \). CFD can also be used to predict smoke and fire risks in buildings, quantify indoor environment quality and design natural ventilation systems \( [71] \). How DL can assist and accelerate these processes still requires in-depth study.

**Causal Inference in Dynamical Systems**  A fundamental pursuit in science is to find causal relationships. In terms of dynamical systems, one may ask which variables influence other variables, either directly or indirectly through intermediates. While traditional approaches to discovery of causation are through conducting controlled real experiments variables \( [96, 13] \), data-driven approaches have been proposed to detect causal relations from observational data in the past few decades \( [49, 47] \). Most data-driven approaches do not directly address learning causality with big data. Many questions remain open, such as using causality to improve DL models and disentangling complex and multiple treatments. Additionally, we are also interested in the system’s response under interventions. For instance, when we use DL to model the COVID-19 dynamics, we need to make accurate predictions under different government regulations such that the government can make the correct decisions and better control the spread of the pandemic.

**Search for Physical Laws**  Another promising direction is to seek physics laws with the help of DL. The search for fundamental laws of practical problems is the main theme of science. When the governing equations of dynamical systems are known, they allow for accurate mathematical modeling, increased interpretability, and robust forecasting. However, current methods are limited to selecting from a large dictionary of possible mathematical terms \( [111, 15, 70, 105] \). But research on data-driven methods based on DL for discovering physics laws are still quite preliminary.

**Efficient Computation**  Given the rapid growth in high-performance computation, we need to improve automation, acceleration streamlining of highly compute-intensive workflows for science. We should focus on how to efficiently train, test, and deploy complex physics-guided DL models on large datasets and high performance computing systems, such that these models can be quickly utilized to solve real-world scientific problems. In conclusion, given the availability of abundant data and rapid growth in computation and DL, we envision that the integration of physics and DL will play an increasingly essential role in advancing scientific discovery and addressing important environmental modeling problems.
References

[1] Ali Al-Aradi, Adolfo Correia, Danilo Naiff, Gabriel Jardim, and Yuri Saporito. Solving nonlinear and high-dimensional partial differential equations via deep learning. *arXiv preprint arXiv:1811.08782*, 2018.

[2] B. Alipanahi, Andrew Delong, Matthew T. Weirauch, and B. Frey. Predicting the sequence specificities of dna- and rna-binding proteins by deep learning. *Nature Biotechnology*, 33:831–838, 2015.

[3] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2019.

[4] Anima Anandkumar, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Nikola Kovachki, Zongyi Li, Burigede Liu, and Andrew Stuart. Neural operator: Graph kernel network for partial differential equations. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020. URL [https://openreview.net/forum?id=fg2ZFmXFO3](https://openreview.net/forum?id=fg2ZFmXFO3).

[5] Brandon Anderson, Truong-Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In *Advances in neural information processing systems (NeurIPS)*, 2019.

[6] Ibrahim Ayed, Emmanuel De Bézenac, Arthur Pajot, and Patrick Gallinari. Learning partially observed PDE dynamics with neural networks, 2019. URL [https://openreview.net/forum?id=HyefgnCqFm](https://openreview.net/forum?id=HyefgnCqFm).

[7] Omri Azencot, N. Erichson, Vanessa Lin, and Michael W. Mahoney. Forecasting sequential data using consistent koopman autoencoders. *arXiv Preprint arXiv:2003.02236*, 2020.

[8] Erkao Bao and Linqi Song. Equivariant neural networks and equivarification. *arXiv preprint arXiv:1906.07172*, 2019.

[9] J. Behler and M. Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98 14:146401, 2007.

[10] Tom Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi, and P. Gentine. Enforcing analytic constraints in neural-networks emulating physical systems. *arXiv: Computational Physics*, 2019.

[11] Tom Beucler, Michael Pritchard, Stephan Rasp, Pierre Gentine, Jordan Ott, and Pierre Baldi. Enforcing analytic constraints in neural-networks emulating physical systems. *arXiv preprint arXiv:1909.00912*, 2019.

[12] Tom Beucler, S. Rasp, M. Pritchard, and P. Gentine. Achieving conservation of energy in neural network emulators for climate modeling. *ArXiv*, abs/1906.06622, 2019.

[13] E. Bollt, J. Sun, and J. Runge. Introduction to focus issue: Causation inference and information flow in dynamical systems: Theory and applications. *Chaos*, 2018.

[14] S. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *ArXiv*, abs/1905.11075, 2019.

[15] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data: Sparse identification of nonlinear dynamical systems. *arXiv preprint arXiv:1509.03580*, 2015.

[16] Ruijin Cang, Hechao Li, Hope Yao, Yang Jiao, and Yi Ren. Improving direct physical properties prediction of heterogeneous materials from imaging data via convolutional neural network and a morphology-aware generative model. *arXiv: Computational Physics*, 2017.

[17] Alejandro Carderera, Sebastian Pokutta, Christof Schütte, and Martin Weiser. Cindy: Conditional gradient-based identification of non-linear dynamics – noise-robust recovery. *arXiv preprint arXiv:2101.02630*, 2021.

[18] G. Carleo and M. Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355:602 – 606, 2017.
[19] Bruno Chaoua. The state of the art of hybrid rans/les modeling for the simulation of turbulent flows. *Springer Netherlands*, 99:279–327, 2017. doi: https://doi.org/10.1007/s10494-017-9828-8.

[20] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6571–6583. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7892-neural-ordinary-differential-equations.pdf.

[21] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[22] Benjamin Chidester, Minh N. Do, and Jian Ma. Rotation equivariance and invariance in convolutional neural networks. *arXiv preprint arXiv:1805.12301*, 2018.

[23] Dragos Bogdan Chirila. *Towards lattice Boltzmann models for climate sciences: The GeLB programming language with applications*. PhD thesis, University of Bremen, 2018.

[24] Taco S. Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning (ICML)*, pages 2990–2999, 2016.

[25] Taco S. Cohen and Max Welling. Steerable CNNs. *arXiv preprint arXiv:1612.08498*, 2016.

[26] Taco S. Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 1321–1330, 2019.

[27] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho. Lagrangian neural networks. *ArXiv*, abs/2003.04630, 2020.

[28] Arka Daw, R. Thomas, C. Carey, J. Read, A. Appling, and A. Karpatne. Physics-guided architecture (pga) of neural networks for quantifying uncertainty in lake temperature modeling. *ArXiv*, abs/2007.04439, 2020.

[29] Richard H. Day. Complex economic dynamics-vol. 1: An introduction to dynamical systems and market mechanisms. *MIT Press Books*, 1, 1994.

[30] Filipe de Avila Belbute-Peres, Thomas D. Economon, and J. Z. Kolter. Combining differentiable pde solvers and graph neural networks for fluid flow prediction. *ArXiv*, abs/2007.04439, 2020.

[31] Emmanuel de Bezenac, Arthur Pajot, and Patrick Gallinari. Deep learning for physical processes: Incorporating prior scientific knowledge. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=By4HsfWAZ.

[32] Emmanuel de Bezenac, Arthur Pajot, and Patrick Gallinari. Deep learning for physical processes: Incorporating prior scientific knowledge. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=By4HsfWAZ.

[33] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 2016.

[34] Jérémie Donà, Jean-Yves Franceschi, sylvain lamprier, and patrick gallinari. {PDE}–driven spatiotemporal disentanglement. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=vLaHRtHvfFP.

[35] E. Dupont, A. Doucet, and Y. Teh. Augmented neural odes. In *NeurIPS*, 2019.

[36] Weinan E, Jiequn Han, and Linfeng Zhang. Integrating machine learning with physics-based modeling. *arXiv:2006.02619*, 2019.
[37] H. El-Sadi. Aircraft design analysis, cfd and manufacturing. *International Journal of Advanced Network, Monitoring and Controls*, 1:59–67, 2016.

[38] E. Labourasse and P. Sagaut. Reconstruction of turbulent fluctuations using a hybrid rans-les approach. *Journal of Computational Physics*, 182:301–336, 2002.

[39] N. Ericsson, Michael Muehlebach, and Michael W. Mahoney. Physics-informed autoencoders for lyapunov-stable fluid flow prediction. *arXiv preprint arXiv:1905.10866*, 2019.

[40] Rui Fang, David Sondak, Pavlos Protopapas, and Sauro Succi. Deep learning for turbulent channel flow. *arXiv preprint arXiv:1812.02241*, 2018.

[41] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. *arXiv preprint arXiv:2002.12880*, 2020.

[42] Marc Finzi, K. Wang, and A. Wilson. Simplifying hamiltonian and lagrangian neural networks via explicit constraints. *ArXiv*, abs/2010.13581, 2020.

[43] U. Forssell and P. Lindskog. Combining semi-physical and neural network modeling: An example of its usefulness u. forssell and p. lindskog. 1997.

[44] Han Gao, Luning Sun, and J. Wang. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *arXiv: Fluid Dynamics*, 2020.

[45] Rohan Ghosh and Anupam K. Gupta. Scale steerable filters for locally scale-invariant convolutional neural networks. *arXiv preprint arXiv:1906.03861*, 2019.

[46] S. Greydanus, Misko Dzamba, and J. Yosinski. Hamiltonian neural networks. *ArXiv*, abs/1906.01563, 2019.

[47] Ruocheng Guo, Lu Cheng, Jundong Li, P. R. Hahn, and Huan Liu. A survey of learning causality with data. *ACM Computing Surveys (CSUR)*, 53:1 – 37, 2020.

[48] Jiequn Han, Linfeng Zhang, and E. Weinan. Solving many-electron schrödinger equation using deep neural networks. *J. Comput. Phys.*, 399, 2019.

[49] Michael Harradon, Jeff Druce, and Brian E. Ruttenberg. Causal learning and explanation of deep neural networks via autoencoded activations. *ArXiv*, abs/1802.00541, 2018.

[50] Philipp Holl, Nils Thuerey, and Vladlen Koltun. Learning to control pdes with differentiable physics. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HyeSin4FPB.

[51] B. Houska, F. Logist, M. Diehl, and J. Van Impe. A tutorial on numerical methods for state and parameter estimation in nonlinear dynamic systems. In D. Alberer, H. Hjalmarsson, and L. Del Re, editors, *Identification for Automotive Systems, Volume 418, Lecture Notes in Control and Information Sciences*, page 67–88. Springer, 2012.

[52] Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[53] Eugene M. Izhikevich. *Dynamical systems in neuroscience*. MIT press, 2007.

[54] Miguel Jaques, Michael Burke, and Timothy Hospedales. Physics-as-inverse-graphics: Unsupervised physical parameter estimation from video. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJeKwTNfV.

[55] J.C. Butcher. *Applied Numerical Mathematics*, volume 20. Elsevier B.V., 1996.

[56] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan Read, Jacob Zwart, Michael Steinbach, and Vipin Kumar. Physics guided rns for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 558–566. SIAM, 2019.
[57] Chiyu "Max" Jiang, Karthik Kashinath, Prabhat, and Philip Marcus. Enforcing physical constraints in {cnn}s through differentiable {pde} layer. In ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations, 2020. URL https://openreview.net/forum?id=q2noHUqMkK.

[58] J. Kani and A. H. Elsheikh. Dr-rnn: A deep residual recurrent neural network for model reduction. ArXiv, abs/1709.00939, 2017.

[59] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. arXiv Preprint arXiv:1710.11431, 2017.

[60] K. Kashinath, M. Mustafa, J-L. Wu A. Albert, C. Jiang, K. Azizzadenesheli S. Esmaeilzadeh, R. Wang, A. Chattopadhyay, A. Singh, A. Manepalli, D. Chirila, R. Yu, R. Walters, B. White, H. Xiao, H. A. Tchelepi, P. Marcus, A. Anandkumar, and P. Hassanzadeh. Physics-informed machine learning: case studies for weather and climate modelling. Journal of Philosophical Transactions of the Royal Society A, 2020.

[61] B. Kim, V. C. Azevedo, N. Thürey, Theodore Kim, M. Gross, and B. Solenthaler. Deep fluids: A generative network for parameterized fluid simulations. Computer Graphics Forum, 38, 2019.

[62] Junhyuk Kima and Changhoon Lee. Deep unsupervised learning of turbulence for inflow generation at various reynolds numbers. arXiv:1908.10515v1, 2019.

[63] D. Kochkov, J. A. Smith, A. Alieva, Qifeng Wang, M. Brenner, and Stephan Hoyer. Machine learning accelerated computational fluid dynamics. ArXiv, abs/2102.01010, 2021.

[64] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In Proceedings of the 35th International Conference on Machine Learning (ICML), volume 80, pages 2747–2755, 2018.

[65] B. O. Koopman. Hamiltonian systems and transformation in hilbert space. Proceedings of the National Academy of Sciences of the United States of America, 17 5:315–8, 1931.

[66] Wouter M. Kouw and Marco Loog. An introduction to domain adaptation and transfer learning. arXiv preprint arXiv:1812.11806, 2018.

[67] J. Kutz. Deep learning in fluid dynamics. Journal of Fluid Mechanics, 814:1–4, 2017.

[68] Vitaly Kuznetsov and Zelda E. Mariet. Foundations of sequence-to-sequence modeling for time series. ArXiv, abs/1805.03714, 2019.

[69] E. Labourasse and P. Sagaut. Advance in rans-les coupling, a review and an insight on the nlde approach. Archives of Computational Methods in Engineering, 11:199–256, 2004.

[70] John H. Lagergren, John T. Nardini, G. Michael Lavigne, E. Rutter, and K. Flores. Learning partial differential equations for biological transport models from noisy spatio-temporal data. Proceedings of the Royal Society A, 476, 2020.

[71] Tom Lawson. Building Aerodynamics. World Scientific Publishing Company, 2001. ISBN 9781860947537.

[72] Seungjun Lee, Haesang Yang, and Woojae Seong. Identifying physical law of hamiltonian systems via meta-learning. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=45NZvF1UHam.

[73] P. Lemberger. On generalization and regularization in deep learning. ArXiv, abs/1704.01312, 2017.

[74] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 991–999, 2015.
[75] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In International Conference on Learning Representations (ICLR ’18), 2018.

[76] Yunzhu Li, Hao He, Jiajun Wu, D. Katabi, and A. Torralba. Learning compositional koopman operators for model-based control. arXiv Preprint arXiv:1910.08264.

[77] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO

[78] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modeling using deep neural networks with embedded invariance. Journal of Fluid Mechanics, 807: 155–166, 2016.

[79] Julia Ling, Andrew Kurzawskim, and Jeremy Templeton. Reynolds averaged turbulence modeling using deep neural networks with embedded invariance. Journal of Fluid Mechanics, 2017.

[80] Vadim Lisitsa, Galina Reshetova, and Vladimir Tcheverda. Finite-difference algorithm with local time-space grid refinement for simulation of waves. Computational geosciences, 16(1): 39–54, 2012.

[81] Dehao Liu and Yan Wang. Multi-fidelity physics-constrained neural network and its application in materials modeling. Journal of Mechanical Design, 141, 2019.

[82] Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. Neural sde: Stabilizing neural ode networks with stochastic noise. ArXiv, abs/1906.02355, 2019.

[83] Yun Long, Xueyuan She, and Saibal Mukhopadhyay. Hybridnet: Integrating model-based and data-driven learning to predict evolution of dynamical systems. ArXiv Preprint arXiv:1806.07439, 2019.

[84] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In International Conference on Machine Learning, pages 3214–3222, 2018.

[85] Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. Journal of Computational Physics, page 108925, 2019.

[86] Bethany Lusch, J. N. Kutz, and S. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. Nature Communications, 9, 2018.

[87] A. Manepalli and A. Albert. Emulating numeric hydroclimate models with physics-informed cgan. 2019.

[88] S. Manzhos and T. Carrington. A random-sampling high dimensional model representation neural network for building potential energy surfaces. The Journal of chemical physics, 125 8: 084109, 2006.

[89] Marios Mattheakis, Pavlos Protopapas, D. Sondak, Marco Di Giovanni, and Efthimios Kaxiras. Physical symmetries embedded in neural networks. arXiv preprint arXiv:1904.08991, 2019.

[90] R. Maulik, O. San, A. Rasheed, and P. Vedula. Subgrid modelling for two-dimensional turbulence using neural networks. Journal of Fluid Mechanics, 858:122–144, 2019.

[91] J. M. McDonough. Introductory Lectures on Turbulence. Mechanical Engineering Textbook Gallery, 2007. URL https://uknowledge.uky.edu/me_textbooks/2

[92] Nikhil Muralidhar, Jie Bu, Z. Cao, Long He, N. Ramakrishnan, D. Tafti, and A. Karpatne. Phynet: Physics guided neural networks for particle drag force prediction in assembly. In SDM, 2020.
[93] Behnam Neyshabur, Srinadh Bhojanapalli, D. McAllester, and Nathan Srebro. Exploring generalization in deep learning. In NIPS, 2017.

[94] F. Noé, A. Tkatchenko, K. Müller, and C. Clementi. Machine learning for molecular simulation. Annual review of physical chemistry, 2020.

[95] Eric J. Parish and K. Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. J. Comput. Phys., 305:758–774, 2016.

[96] J. Pearl. Causal inference in statistics: An overview. Statistics Surveys, 3:96–146, 2009.

[97] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=roNqYL0_XP

[98] Michael Poli, Stefano Massaroli, Junyoung Park, A. Yamashita, H. Asama, and Jinkyoo Park. Graph neural ordinary differential equations. ArXiv, abs/1911.07532, 2019.

[99] Christopher Rackauckas, Y. Ma, Julius Martensen, Collin Warner, K. Zubov, Rohit Supekar, D. Skinner, and Ali Ramadhan. Universal differential equations for scientific machine learning. ArXiv, abs/2001.04385, 2020.

[100] Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. Journal of Computational Physics, 357:125–141, 2018.

[101] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561, 2017.

[102] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378:686–707, 2019.

[103] Stephan Rasp, Peter D. Dueben, Sebastian Scher, Jonathan A. Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: A benchmark dataset for data-driven weather forecasting. arXiv:2002.00469, 2020.

[104] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and Prabhat. Deep learning and processing understanding for data-driven earth system science. Nature, 566(7743):195–204, 2019. doi: 10.1038/s41586-019-0912-1. URL https://doi.org/10.1038/s41586-019-0912-1

[105] Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Data-driven discovery of partial differential equations. arXiv preprint arXiv:1609.06401, 2016.

[106] Pierre Sagaut, Sebastien Deck, and Marc Terracol. Multiscale and Multiresolution Approaches in Turbulence. Imperial College Press, 2006.

[107] O. San and R. Maulik. Neural network closures for nonlinear model order reduction. Advances in Computational Mathematics, 44:1717–1750, 2018.

[108] Alvaro Sanchez-Gonzalez, N. Heess, Jost Tobias Springenberg, J. Merel, Martin A. Riedmiller, Raia Hadsell, and P. Battaglia. Graph networks as learnable physics engines for inference and control. ArXiv, abs/1806.01242, 2018.

[109] Alvaro Sanchez-Gonzalez, J. Godwin, T. Pfaff, Rex Ying, J. Leskovec, and P. Battaglia. Learning to simulate complex physics with graph networks. ArXiv, abs/2002.09405, 2020.

[110] P. Schmid. Dynamic mode decomposition of numerical and experimental data. Journal of Fluid Mechanics, 656:5–28, 2008.

[111] Michael D. Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. Science, 324:81 – 85, 2009.
[112] Kristof Schütt, P. Kindermans, Huziel Enoc Saucedo Felix, Stefan Chmiela, A. Tkatchenko, and K. Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In NIPS, 2017.

[113] V. Shah, Ameya Joshi, Sambuddha Ghosal, B. Pokuri, S. Sarkar, B. Ganapathysubramanian, and C. Hegde. Encoding invariances in deep generative models. ArXiv, abs/1906.01626, 2019.

[114] Gregor N. C. Simm, Robert Pinsler, Gábor Csányi, and José Miguel Hernández-Lobato. Symmetry-aware actor-critic for 3d molecular design. In International Conference on Learning Representations, 2021. URL [https://openreview.net/forum?id=jEYkJPeIxYN](https://openreview.net/forum?id=jEYkJPeIxYN)

[115] Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. arXiv preprint arXiv:1708.07469, 2018.

[116] Steven H. Strogatz. Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering. CRC press, 2018.

[117] Naoya Takeishi, Y. Kawahara, and T. Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. In NIPS, 2017.

[118] Michael L. Thompson and M. Kramer. Modeling chemical processes using prior knowledge and neural networks. Aiche Journal, 40:1328–1340, 1994.

[119] Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. Accelerating eulerian fluid simulation with convolutional networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 3424–3433. JMLR.org, 2017.

[120] John Wainwright. Dynamical systems in cosmology. Cambridge University Press, 2005.

[121] Robin Walters, Jinx Li, and Rose Yu. Trajectory prediction using equivariant continuous convolution. In International Conference on Learning Representations, 2021. URL [https://openreview.net/forum?id=J8_GttYLFgr](https://openreview.net/forum?id=J8_GttYLFgr)

[122] Jiang Wang, C. Wehmeyer, F. Noé, and C. Clementi. Machine learning of coarse-grained molecular dynamics force fields. ACS Central Science, 5:755 – 767, 2019.

[123] Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2020.

[124] Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 1457–1466, 2020.

[125] Rui Wang, Danielle Maddix, Christos Faloutsos, Yuyang Wang, and Rose Yu. Bridging physics-based and data-driven modeling for learning dynamical systems. arXiv Preprint arXiv:2101.10616, 2020.

[126] Rui Wang, Robin Walters, and Rose Yu. Incorporating symmetry into deep dynamics models for improved generalization. 2021. URL [https://openreview.net/forum?id=wta_8HxZ2Kd](https://openreview.net/forum?id=wta_8HxZ2Kd)

[127] Rui Wang, Robin Walters, and Rose Yu. Bridging physics-based and data-driven modeling for learning dynamical systems. arXiv preprint arXiv:2102.10271, 2021.

[128] Maurice Weiler and Gabriele Cesa. General E(2)-equivariant steerable CNNs. In Advances in Neural Information Processing Systems (NeurIPS), pages 14334–14345, 2019.

[129] Maurice Weiler, Fred A. Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant CNNs. Computer Vision and Pattern Recognition (CVPR), 2018.

[130] B. White, A. Singh, and A. Albert. Downscaling numerical weather models with gans. 2019.
[131] Steffen Wiewel, Moritz Becher, and Nils Thuerey. Latent space physics: Towards learning the temporal evolution of fluid flow. In Computer Graphics Forum, volume 38, pages 71–82, 2019.

[132] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael S. Steinbach, and Vipin Kumar. Integrating physics-based modeling with machine learning: A survey. CoRR, abs/2003.04919, 2020. URL https://arxiv.org/abs/2003.04919

[133] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5028–5037, 2017.

[134] Dongxian Wu, Liyao Gao, X. Xiong, Matteo Chinazzi, Alessandro Vespignani, Y. Ma, and Rose Yu. Deepgleam: a hybrid mechanistic and deep learning model for covid-19 forecasting. ArXiv, abs/2102.06684, 2021.

[135] Jin-Long Wu, Karthik Kashinath, Adrian Albert, Dragos Chirila, Prabhat, and Heng Xiao. Enforcing Statistical Constraints in Generative Adversarial Networks for Modeling Chaotic Dynamical Systems. arXiv e-prints, May 2019.

[136] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. ACM Transactions on Graphics (TOG), 37(4):95, 2018.

[137] E. Yeung, Soumya Kundu, and Nathan Oken Hodas. Learning deep neural network representations for koopman operators of nonlinear dynamical systems. 2019 American Control Conference (ACC), pages 4832–4839, 2019.

[138] Yuan Yin, Vincent LE GUEN, Jérémie DONA, Emmanuel de Bezenac, Ibrahim Ayed, Nicolas THOME, and patrick gallinari. Augmenting physical models with deep networks for complex dynamics forecasting. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=kmG8vRXTFV

[139] C. Zhang, S. Bengio, Moritz Hardt, B. Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. ArXiv, abs/1611.03530, 2017.

[140] L. Zhang, Jiequn Han, Han Wang, W. Saidi, R. Car, and E. Weinan. End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems. In NeurIPS, 2018.

[141] Linfeng Zhang, Jiequn Han, Han Wang, R. Car, and E. Weinan. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. Physical review letters, page 143001, 2018.

[142] Yaofeng Zhong, Biswadip Dey, and Amit Chakraborty. Benchmarking energy-conserving neural networks for learning dynamics from data. arXiv Preprint arXiv:2012.02334, 2020.

[143] Yinhao Zhu, N. Zabaras, P. Koutsourelakis, and P. Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. ArXiv, abs/1901.06314, 2019.