

# WHEN PHYSICS MEETS MACHINE LEARNING: A SURVEY OF PHYSICS-INFORMED MACHINE LEARNING

A PREPRINT

Chuizheng Meng, Sungyong Seo, Defu Cao, Sam Griesemer, Yan Liu

Department of Computer Science

University of Southern California

Los Angeles, CA 90089

{chuizhem, sungyongs, defuca, samgriesemer, yanliu.cs}@usc.edu

April 1, 2022

## ABSTRACT

Physics-informed machine learning (PIML), referring to the combination of prior knowledge of physics, which is the high level abstraction of natural phenomena and human behaviours in the long history, with data-driven machine learning models, has emerged as an effective way to mitigate the shortage of training data, to increase models' generalizability and to ensure the physical plausibility of results. In this paper, we survey an abundant number of recent works in PIML and summarize them from three aspects: (1) motivations of PIML, (2) physics knowledge in PIML, (3) methods of physics knowledge integration in PIML. We also discuss current challenges and corresponding research opportunities in PIML.

## 1 Introduction

Machine learning/deep learning models have already achieved tremendous success in a number of domains such as computer vision [1, 2, 3, 4, 5] and natural language processing [6, 7, 8, 9, 10, 11, 12, 13, 14], where large amounts of training data and highly expressive neural network architectures together give birth to solutions outperforming previously dominating methods. As a consequence, researchers have also started exploring the possibility of applying machine learning models to advance scientific discovery and to further improve traditional analytical modeling [15, 16, 17, 18, 19, 20, 21].

While given a set of input and output pairs, deep neural networks are able to extract the complicated relations between the input and output via appropriate optimization over adequate large amount of data, *prior knowledge* still acts as an important role in finding the optimal solution. As the high level extraction of data distributions and task properties, prior knowledge, if incorporated properly, can provide rich information not existing or hard to extract in limited training data, and helps improve the data efficiency, the ability to generalize, and the plausibility of resulting models.

Physics knowledge, which has been collected and validated explicitly both theoretically and empirically in the long history, contains tremendous abstraction and summary of natural phenomena and human behaviours in many important scientific and engineering applications. Thus in this paper, we focus on the topic of integrating prior physics knowledge into machine learning models, i.e. *physics-informed machine learning (PIML)*. Compared to the integration of other types of prior knowledge, such as knowledge graphs, logic rules and human feedback [22], the integration of physics knowledge requires specific design due to its special properties and forms.

In this paper, we survey a wide range of recent works in PIML and summarize them from three aspects. (1) Motivations of PIML, which can be further categorized to using machine learning to serve tasks in physics domains and incorporating physics principles to existing machine learning models for real-world tasks. (2) Physics knowledge in PIML, each type of which is a general principle covering a wide range of problems. (3) Methods of physics knowledge integration in PIML. Depending on the location of knowledge integration, we categorize the methods into data enhancement, neural network architecture design, and physics-informed optimization.

The paper is organized as follows. Sec 2 analyzes two main categories of motivations using PIML: one mainly serves tasks in physics domain, while the other serves real-world problems. Sec 3 introduces several general physics principles widely used in PIML. Sec 4 investigates methods of physics knowledge integration. Sec 5 discusses challenges and potential future research directions of PIML. Sec 6 serves as the summary of the whole paper.

## 2 Motivations of PIML

### 2.1 ML for Physics: Enhancement of Physics Models via Data-Driven Methods

Physical science problems involves various data-intensive tasks including spatiotemporal data modeling, causal reasoning, computer vision, probabilistic inference and so on. Since machine learning methods have achieved great success in these tasks, using machine learning models for furthering scientific discovery in physics has received increasing interest in recent years. Compared to existing numerical or pure physics based methods, physics-informed machine learning methods have advantages in flexibility, generalizability, and computation cost. Meanwhile, they still enforce physical plausibility. In this section, we introduce recent developments of exploiting machine learning for several physics related tasks, including surrogate simulation, data-driven PDE solvers, parameterization of physics models, reduced-order models, and knowledge discovery.

#### 2.1.1 Simulation

Many physics-informed machine learning models have been proposed to act as surrogate solutions to numerical simulation in many domains, such as turbulence simulation [23, 24, 25], climate simulation [26, 27, 28, 29], and particle system simulation [30, 31, 32, 33]. Compared to numerical simulation, neural network based machine learning models enjoy following advantages. (1) **Lower implementation costs.** To build a high-quality numerical simulator, researchers usually need years of engineering effort and must choose numerous physically meaningful parameters depending on the task. Instead, machine learning models can be trained directly from a large amount of observed data. (2) **Stronger ability to generalize.** Machine learning models can share the same architecture for different types of problems in the same category, and then can be further specialized for each problem with observed data. For example, [30] proposes a general framework to learn particle simulation, and can generalize across fluid, rigid, and deformable material systems. (3) **Lower computation costs.** Existing neural network building blocks such as multilayer perceptrons and convolutions can be efficiently accelerated by various hardware including CPU, GPU, FPGA and ASIC, giving advantages of computation costs to neural networks composed of these blocks. In fluid flow prediction, the model proposed in [34] achieves prediction results close to ground truth simulation at a running speed 60x faster than the numerical method.

Meanwhile, due to the large and complex optimization space of neural networks, it is critical to incorporate inductive bias of physics knowledge about the task into either the training data, the model architecture, or the optimization process. This ensures greater physical plausibility of resulting models, further improving the robustness in real-world settings. The detailed techniques of integration will be introduced in Section 4.

#### 2.1.2 Data-driven PDE Solver

Many real-world problems from physical systems are mostly about how to describe observations based on partial differential equations and numerically solve the equations. There are many traditional methods to numerically solve PDEs such as spectral methods, finite-difference methods (FDM), finite element method (FEM), finite volume method (FVM), etc. All of these methods are numerical and they require proper discretization or a finite number of steps to approximate continuous solution.

Combining machine learning techniques with PDE models has a long history in machine learning. [35] introduce a method to reconstruct the deterministic portion of the equations of motion directly from a data series. This approach employs an informational measure of model optimality to guide searching through the space of dynamical systems. [36] present a framework for computer-aided multi scale analysis, which enables models at a fine (microscopic/stochastic) level of description to perform modeling tasks at a coarse (macroscopic/systems) level. These macroscopic modeling tasks, yielding information over long time and large scales, are accomplished through approximately initialized calls to the microscopic similar for only short times and small spatial domains.

More recently, [37, 38] introduce a solution to solve PDE in a data-driven manner. Rather than analytically solving a given equation, they infer solutions to targeted PDE via supervised learning. Given an input tuple  $(x, t)$ , they compute spatial and time derivatives from black-box models and then the output are connected based on a form of PDE to update all learnable parameters in the black-box models. This method does not require any discretization and it is fully data-driven to find a surrogate model.

[39] approximate the unknown solution as well as the nonlinear dynamics by two deep neural networks. The first network acts as a prior on the unknown solution and essentially enables us to avoid numerical differentiations which are inherently ill-conditioned and unstable. The second network represents the nonlinear dynamics and helps us distill the mechanisms that govern the evolution of a given spatiotemporal dataset. [40] introduce a technique based on the singular vector canonical correlation analysis (SVCCA) for measuring the generality of neural network layers across a continuously-parameterized set of tasks. They illustrate this method by studying generality in neural networks trained to solve parameterized boundary value problems based on the Poisson partial differential equation. [41] directly model the mapping between a PDE’s solution and its initial conditions via message passing in the spatial domain. The following work [42] further extends the modeling to the frequency domain, which inherently generalizes to multiple spatial resolutions.

In many physical systems, the governing equations are known with high confidence, but direct numerical solution is prohibitively expensive. Often this situation is alleviated by writing effective equations to approximate dynamics below the grid scale. This process is often impossible to perform analytically and is often ad hoc. [43] propose data-driven discretization, a method that uses machine learning to systematically derive discretizations for continuous physical systems. [44] target the problem of reducing numerical errors of iterative PDE solvers and compare different learning approaches for finding complex correction functions. They integrate the PDE solver into the training loop and thereby allow the model to interact with the PDE during training.

**Downscaling** Directly solving PDEs require spatial and temporal discretization and finer resolution is more desired to capture physically reliable solutions. However, it increases the computational cost and modeling complexity. Downscaling techniques have been widely used as a solution to capture physical variables that need to be modeled at a finer resolution from a coarser resolution. Recently, artificial neural networks have shown a lot of promise for this problem, given their ability to model non-linear relationships. [45] present a downscaling algorithms using neural networks to leverage the relationships between Satellite precipitation estimates (SPEs) and cloud optical and microphysical properties in northeast Austria. [46] present DeepSD (Statistically Downscaling), a generalized stacked super resolution convolutional neural network (SRCNN) framework for statistical downscaling of climate variables. DeepSD augments SRCNN with multi-scale input channels to maximize predictability in statistical downscaling. [47] introduce a data-driven framework for the identification of unavailable coarse-scale PDEs from microscopic observations via machine-learning algorithms. Specifically, using Gaussian processes, artificial neural networks, and/or diffusion maps, the proposed framework uncovers the relation between the relevant macroscopic space fields and their time evolution (the right-hand side of the explicitly unavailable macroscopic PDE)

### 2.1.3 Parameterization

A common technique used in numerical models describing complex physics phenomena is replacing dynamic systems that are hard to model with parameterized simple processes, which is named *parameterization*. While the traditional way to decide parameters is to minimize the discrepancy between models’ output and observed data by conducting grid search or Bayesian inference within a certain parameter space, recent works have adopted advances in machine learning as the approximation of unknown dynamics.

In geology, [48] study the application of Wasserstein GAN [49] for the parametrization of geological models. The effectiveness of the method is assessed for uncertainty propagation tasks using several test cases involving different permeability patterns and subsurface flow problems. [50] develop a new predictor for near-bed suspended sediment reference concentration under unbroken waves using genetic programming, a machine learning technique.

In meteorological science, [51] show that a neural network-based parameterization is successfully trained using a near-global aqua-planet simulation with a 4-km resolution (NG-Aqua). The neural network predicts the apparent sources of heat and moisture averaged onto (160 km<sup>2</sup>) grid boxes. [52] present a novel approach to convective parameterization based on machine learning, using an aquaplanet with prescribed sea surface temperatures as a proof of concept. A deep neural network is trained with a super-parameterized version of a climate model in which convection is resolved by thousands of embedded 2-D cloud resolving models.

In chemistry, supervised learning on molecules has incredible potential to be useful in chemistry, drug discovery, and materials science. [53] reformulate existing models into a single common framework we call Message Passing Neural Networks and explore additional novel variations within this framework. Using MPNNs we demonstrate state of the art results on an important molecular property prediction benchmark. [54] utilize molecular graph data for property prediction based on spatial graph convolution neural networks.

### 2.1.4 Reduced-Order Models

Instead of directly modeling the complex dynamics in the observation space, researchers have developed simplified but more interpretable models (reduced-order models, ROMs) in the hidden space, which is usually derived from the observation space via dimensionality reduction. In recent works, machine learning models have emerged as effective tools to discover such hidden spaces and more information preserving transformations between them and the observation space.

In dynamic system analysis, Koopman theory is a typical ROM. Koopman theory is based on the insight that the state space of a non-linear dynamic system can be encoded into an infinite-dimensional space where the dynamics is linear [55]. In practice, people assume the infinite-dimensional space can be approximated with a finite-dimensional space. The key problem is then to find a proper pair of encoder/decoder to map from/to the state space to/from the hidden space. Traditionally, people construct the encoder/decoder with hand-crafted functions, such as the identity function in Dynamic Mode Decomposition (DMD) [56], nonlinear functions in Extended DMD (EDMD) [57], and kernel functions in Kernel DMD (KDMD) [58]. However, hand-crafted functions may fail to fit complex dynamic systems and are hard to design without domain-specific knowledge. Thus, recent works [59, 60, 61] construct encoders/decoders using neural networks as trainable universal approximators. They demonstrate that the combination of neural networks and Koopman theory achieves comparable or even higher performance than the Koopman approximators with hand-crafted mapping functions, while enjoying the ability to generalize to multiple datasets with the same design. [59] further shows that the integration of Koopman theory allows the model to adapt to new systems with unknown dynamics faster than pure neural networks.

In fluid dynamics, ROMs is largely considered due to the unprecedented physical insight into turbulence offered by high-fidelity computational fluid dynamics (CFD). [62] develop a dimensionality reduction method called Non-Intrusive Reduced Order Model (NIROM) for predicting the turbulent air flows found within an urban environment. [63] demonstrate a deep learning based approach to build a ROM using the POD basis of canonical DNS datasets, for turbulent flow control applications. They find that a type of Recurrent Neural Network, the Long Short Term Memory (LSTM) which has been primarily utilized for problems like speech modeling and language translation, shows attractive potential in modeling temporal dynamics of turbulence.

### 2.1.5 Causality

**Causal Discovery and Causal Inference in Time Series** How to discover the underlying causal structure is a fundamental problem and has still been actively studied. [64, 65, 66] introduce the problem and provided a mathematical framework for causal reasoning and inference under causal graphical models (also known as Bayesian networks (BN) [67]). [68] formalize a concept of quantifiable causality in time series, called Granger causality. From the pioneering works, learning causal associations from time series has been an emerging topic in machine learning and deep learning community as well. [69] propose a method to distinguish direct from indirect dependencies and common drivers among multiple time series to reconstruct a causal network. [70] quantify causal associations in nonlinear time series and [71, 72] provide promising applications of the causal discovery in time series. [73] introduce a smooth acyclicity constraint to multivariate time series inspired by [74] that consider a causal discovery as a purely continuous optimization problem.

For causal inference part, G-computation formula, g-estimation of structural nested mean models [75], and inverse probability of treatment weighting in marginal structural models (MSMs) [76, 77] rely on linear predictors for estimation to estimate treatment effects. Recurrent marginal structural networks (RMSNs) [78] is proposed to further improve MSM's ability by capturing nonlinear dependencies. In addition, Gaussian process [79, 80] has been tailored to estimate treatment response in a continuous-time settings in order to incorporate non-deterministic quantification. Furthermore, [81] and [82] theoretically prove that observed proxy variables can be used to capture hidden confounders and estimate treatment effects. TSD [83] introduces recurrent neural networks in the factor model to estimate the dynamics of confounders. In a similar vein, [84] propose a sequential deconfounder to infer hidden confounders by using Gaussian process latent variable model. DTA [85] estimates treatment effects under dynamic setting using observed data as noisy proxies. Besides, DSW [86] infers the hidden confounders by using a deep recursive weighted neural network that combines current treatment assignment and historical information. DNDC [87] aims to learn how hidden confounders behave over time by using current network observation data and historical information. Although many works are successful to discover unknown causal structure from observational data directly and make effective causal inference, there are few works to leverage explicit causal relations from physical knowledge to improve data-driven models.

**Counterfactual Analysis of Physical Dynamics** Counterfactual analysis in the physical world is typically concerned with analytically predicting the effects of various types of interventions/treatments, including the physical laws

of current environment [88, 89, 90], the actions of the agent itself [91, 92, 93, 94], and the decision outcomes of other agents in multi-agent systems [95, 96]. With the development of 3D simulation in the field of machine learning, several benchmarks performing counterfactual analysis in physics have emerged. CLEVRER [97] asks agents to answer a counterfactual question after observing a video showing the movement and collision of a 3D object. Physics 101 [98] presents a video benchmark containing over 101 real-world objects physically interacting in four different physical scenarios. [99] proposes a counterfactual benchmark with two tasks: a scene where balls interact according to unknown interaction laws (e.g., gravity or elasticity), and a scene where clothes are folded by the wind. The agent needs to discover causal relationships between counterfactual variables and objects and then predicts future frames. CoPhy [100] separates the observed experiments from those of counterfactuals and includes three complex 3D scenarios involving rigid body dynamics.

The counterfactual analysis of the physics-informed models relies on the separation between the physical information features and the remaining features [101, 102] and may incorporate additional information based on the available prior knowledge of the scene [103]. Recently, PhyDNet [104] explicitly disentangles PDE dynamics from unknown complementary information. The interpretable intuitive physical model [105] proposes an encoder-decoder framework for predicting future collision event frames. The encoder layer infers physical properties, such as mass and friction, from the input frames. Then, the decoder decomposes the potential physical vectors by outputting optical flow. PIP [106] uses a deep generative model to build approximate mental simulations by generating a framework for future physical interactions and then employs selective temporal attention in the form of spanwise selection to predict the outcome of physical interactions. CWMs [99] allow unsupervised modeling of relationships between the intervened observations and the alternative futures by learning an estimator of the latent confounding factors. Cophy [100] predicts alternative outcomes of a physical experiment by estimating the potential performance of confounding factors. Filter-Cophy [107] further learns and acts on a suitable hybrid potential representation based on a combination of dense features, sets of 2D keypoints, and an additional latent vector per keypoint.

## 2.2 Physics for ML: Improvement of Data-Driven Models from External Knowledge

Data-driven methods represented by machine learning and neural networks have achieved great success recently in a wide range of real-world problems due to the universal approximation ability of neural networks [108], the increase of available data for model training, and the rapid development of hardware neural network accelerators. However, the optimization of neural networks is highly non-convex, and its convergence to global minima is hard to achieve in practice. Optimization processes converging to local minima without constraints may lead to models with limited generalization ability or results violating existing knowledge including commonsense, logic rules, and physics laws [22]. In this work, we focus on the integration of physics knowledge in machine learning, and the following sections will introduce the integration in multiple domains of machine learning respectively.

### 2.2.1 Supervised Learning

Typical supervised learning tasks including classification and forecasting cover various real-world applications from multi-agent systems, computer vision, time series analysis, and spatio-temporal data modeling. While deep neural network-based methods have been the dominating solution in these areas recently, people have also explored several novel approaches with the incorporation of physics knowledge in tasks on data from physics-related scenarios, such as object-centric data, spatio-temporal data, and geometry data.

**Object-centric Data** Object-centric data is generated by systems composed of multiple discrete objects. Examples of real-world object-centric data include trajectory data from multi-agent systems [109, 110, 111], position and velocity data from networks of motion sensors [110, 112], and molecule data [53]. Graph neural network (GNN)-based methods [88, 113, 114, 111] achieve state-of-the-art results on most tasks with object-centric data due to the matching of its inductive bias [113] and the interacting physics property in object-centric data. [115] demonstrates that GNNs generalize well in many popular object-centric tasks because the forwarding process of GNNs aligns to the underlying reasoning process. In addition to this alignment, other types of physics knowledge are also integrated in recent works. [116] obtains a joint model for the total energy and interatomic forces of molecules that follows the energy-conserving law. [117] proposes Deep Lagrangian Networks (DeLaN) for robotics, on which Lagrangian Mechanics is imposed and physical plausibility is maintained. [109] designs models for forecasting n-body systems that learn and respect exact conservation laws - Hamiltonian mechanics - in an unsupervised manner. Lagrangian Neural Networks (LNNs) in [118] further parameterize arbitrary Lagrangians using neural networks without canonical coordinates or restrictions of the functional form of learned energies, and can be applied to graphs and continuous systems. [59] learns compositional Koopman operators using GNNs and shows better efficiency and generalization than existing GNN baselines in multi-object systems such as ropes and soft robots.



**Spatio-Temporal Data** Spatio-temporal data records the dynamics of values of interest in multiple locations within a period of time. The most common form of spatio-temporal data is video, where all locations are aligned to rectangular mesh grids in some specific area. Spatio-temporal data can also be generated from networks of irregularly spaced sensors in domains such as traffic [119], weather [120], and electricity [121]. Since the underlying processes of spatio-temporal data are usually governed by physics laws, physics-informed methods have potentials to further improve the performance of neural network models. [104] explicitly disentangles known PDE dynamics from unknown factors and performs PDE-constrained prediction in latent-space, both of which contributes to video forecasting performance. [122] mimics the pipeline of physical flow simulation, and evolves and accumulates point features in point clouds using flow velocities generated from a high-dimensional force field. It demonstrates the efficacy of the proposed method in various point cloud classification and segmentation problems. [123] learns finite differences of sparsely available data points inspired by physics equations, and shows the superiority in synthetic graph signal prediction and real-world weather forecasting tasks. [124] proposes a general continuous-time differential model for dynamical systems, which admits arbitrary space and time discretizations, and enables efficient neural PDE inference.

**Manifold Data** Manifold data describes signals defined on non-planar surfaces such as spheres and surfaces of complex 3D objects, where the Euclidean geometry only holds locally near each point. Examples are magnetoencephalography (MEG) brain activity signal data [125] (on sphere), omnidirectional image data [126] (on 3D objects), and human scan data [127] (on 3D objects). Without the restriction that data points are distributed on planes, general manifold data comes with richer information about the space including local structures and symmetries of the space. Meanwhile, existing convolution-based neural network architectures are either no longer applicable in non-planar manifolds (CNNs) or not capable of fully exploiting the spatial information (GNNs). Instead, physics-informed neural networks incorporate established knowledge of manifolds into the construction of new types of convolutions for manifolds and bridge the gap. [128] constructs intrinsic CNN-like architectures on non-planar surfaces under the geodesic polar coordinate system. [129] alternatively uses the principal curvature directions as fixed gauges to construct convolutions in corresponding tangent fields. [130] constructs convolution kernels as rotating filters and collects the strongest responses among all possible directions. [131, 132] further extends convolution filters to be gauge-equivariant via parallel transporting geometric features to the same vector space before applying the filters.

### 2.2.2 Model-Based Control

Due to the integration of physics knowledge, physics-informed machine learning models enjoy better physical plausibility, higher data efficiency and stronger generalization ability compared to pure neural network models, all of which are critical properties for constructing a good model describing the relation between the control input and the state transition of dynamic systems. Recent works have shown that physics-informed machine learning models achieve significant success in model predictive control and model-based reinforcement learning. [133] integrates a deep neural network (DNN) that learn high-order interactions into the dynamics model, and constrain the Lipschitz constant of the DNN to guarantee system stability. [117, 134] enforce Lagrangian and Hamiltonian dynamics in the modeling of underlying system dynamics respectively, and both outperform model learning approaches without physics knowledge in trajectory tracking error, learning speed, and robustness. [59] shows that dynamics learned via compositional Koopman operators can quickly adapt to new environments of unknown physical parameters in online learning. [135, 136] augment partial differential equations (PDEs) that approximately describe continuous physical systems with controllable force terms, and demonstrate that proposed methods successfully control the evolution of complex physical systems.

## 3 Physics Knowledge in PIML

In this section, we introduce several categories of general physics knowledge integrated in PIML. While there is much more domain/task-specific knowledge that can be incorporated for corresponding solutions, each category we introduce in this section covers a wide range of problems and inspires a series of works generally applicable to them instead of leading to only one or two task-specific solutions.

### 3.1 Classical Mechanics and Energy Conservation Laws

Newtonian, Lagrangian, and Hamiltonian mechanics are three typical approaches describing systems of classical mechanics. While the Newtonian mechanics has been widely used to describe the relations among locations, velocities, accelerations and forces, Lagrangian and Hamiltonian mechanics provide effective tools to enforce laws of conservation of energy in the modeling of dynamic systems. Since Newtonian mechanics described as the famous Newton’s Three Laws has been widely known, here we only focus on Lagrangian and Hamiltonian mechanics.

**Lagrangian Mechanics** The Lagrangian mechanics defines the Lagrangian function  $L$  of generalized coordinates  $\mathbf{q}$  and its gradient w.r.t time  $\dot{\mathbf{q}}$  to fully describe dynamics of a mechanical system. Usually  $L$  is chose as the difference between the kinetic energy  $T$  and the potential energy  $V$ , i.e.  $L(\mathbf{q}, \dot{\mathbf{q}}) \equiv T(\mathbf{q}, \dot{\mathbf{q}}) - V(\mathbf{q})$ .

The defined Lagrangian function must satisfy the principle of stationary action: the real physical trajectory of a system will always take is the one in which the action happens to be stationary, which is mathematically expressed as:

$$\delta \int_{t_1}^{t_2} L dt = 0. \quad (1)$$

We can further derive the important Euler-Lagrange equation from Eq 1:

$$\frac{d}{dt} \nabla_{\dot{\mathbf{q}}} L = \nabla_{\mathbf{q}} L. \quad (2)$$

According to Noether's theorem, the energy of a system is conserved if the system has a time-translation symmetry, i.e. if the Lagrangian function does not explicitly depend on time ( $\frac{\partial L}{\partial t} = 0$ ). Therefore, as long as we model the Lagrangian function as a function of  $\mathbf{q}, \dot{\mathbf{q}}$  without the explicit time term  $t$  and satisfies Eq 2, it automatically satisfies the law of conservation of energy. With Eq 2 we can derive the expression of  $\ddot{\mathbf{q}}$  with the chain rule [118]:

$$(\nabla_{\dot{\mathbf{q}}} \nabla_{\dot{\mathbf{q}}}^T L) \ddot{\mathbf{q}} + (\nabla_{\mathbf{q}} \nabla_{\dot{\mathbf{q}}}^T L) \dot{\mathbf{q}} = \nabla_{\mathbf{q}} L. \quad (3)$$

When  $L$  is modeled as a differential function such as neural networks, we can solve  $\ddot{\mathbf{q}}$  as:

$$\ddot{\mathbf{q}} = (\nabla_{\dot{\mathbf{q}}} \nabla_{\dot{\mathbf{q}}}^T L)^{-1} [\nabla_{\mathbf{q}} L - (\nabla_{\mathbf{q}} \nabla_{\dot{\mathbf{q}}}^T L) \dot{\mathbf{q}}]. \quad (4)$$

Usually one system's  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  can be observed as spatial coordinates and speeds of objects, thus the system can be solved with  $(\mathbf{q}, \dot{\mathbf{q}})|_{t=T} = \int_0^T (\dot{\mathbf{q}}, \ddot{\mathbf{q}}) dt$ .

For systems with non-conservative forces, Eq 2 can be extended as

$$\frac{d}{dt} \nabla_{\dot{\mathbf{q}}} L - \nabla_{\mathbf{q}} L = \boldsymbol{\tau}, \quad (5)$$

where  $\boldsymbol{\tau}$  are generalized forces and can be used for controllers such as a PD-Controller [117].

**Hamiltonian Mechanics** The Hamiltonian mechanics defines the Hamiltonian function  $H$  as the function of a pair of variables  $(\mathbf{q}, \mathbf{p})$ , where  $\mathbf{q}$  is the vector of spatial coordinates of system objects and  $\mathbf{p}$  is the vector of their momentum.  $(\mathbf{q}, \mathbf{p})$  must satisfy the canonical condition:

$$\mathbf{p} = \nabla_{\dot{\mathbf{q}}} L, \quad (6)$$

where  $L$  is the Lagrangian function of the system. The Hamiltonian function is defined as follows:

$$H(\mathbf{q}, \mathbf{p}) = \dot{\mathbf{q}} \cdot \mathbf{p} - L. \quad (7)$$

Combine Eq 6, Eq 7 and Eq 2 we can derive:

$$\dot{\mathbf{p}} = -\nabla_{\mathbf{q}} H, \quad \dot{\mathbf{q}} = \nabla_{\mathbf{p}} H. \quad (8)$$

With Eq 8, we can verify that

$$\frac{dH}{dt} = \dot{\mathbf{q}} \nabla_{\mathbf{q}} H + \dot{\mathbf{p}} \nabla_{\mathbf{p}} H = 0. \quad (9)$$

In the context of classical mechanics, we have  $\mathbf{p} = M\dot{\mathbf{q}}$ ,  $T = \frac{1}{2}\dot{\mathbf{q}}^T M \dot{\mathbf{q}}$ ,  $L = T - V$ , then Eq 7 can be rewritten as  $H(\mathbf{q}, \mathbf{p}) = \dot{\mathbf{q}}^T M \dot{\mathbf{q}} - T + V = T + V$ , where  $M$  is the mass matrix of the system. Here the Hamiltonian function  $H$  is exactly the total energy of the system. Therefore, Eq 9 shows that systems described with Hamiltonian mechanics follows the law of conservation of energy.

### 3.2 Symmetry, Invariant and Equivariant Functions

A symmetry defined on an object or system is some transformation that keeps certain properties unchanged. Typical symmetries include shifts in visual object classification problems, rotations in molecule property prediction problems, and permutations in particle systems. For one object or system, its symmetries form a *symmetry group*, where the following rules are satisfied: (1) associativity, (2) identity, (3) inverse, (4) closure. Some common symmetry groups are:  $T(n)$  ( $n$ -dim translation group),  $O(n)$  ( $n$ -dim distance-perserving group including both rotation and reflection),  $SO(n)$  ( $n$ -dim rotation group),  $E(n)$  ( $n$ -dim Euclidean group, including translation, rotation and reflection).

On a set  $\Omega$  representing a domain, we can define a *group action* as a mapping  $(\mathbf{g}, u) \mapsto \mathbf{g}.u$ , where  $\mathbf{g}$  is one element of a symmetry group  $\mathfrak{G}$ ,  $u$  and  $\mathbf{g}.u$  are two points in  $\Omega$ . A group action shall satisfy associativity:  $\mathbf{g}.\mathbf{h}.u = (\mathbf{gh}).u$  for all  $\mathbf{g}, \mathbf{h} \in \mathfrak{G}$  and  $u \in \Omega$ . In comparison, *group action* on a signal space  $\mathcal{X} : \Omega \rightarrow \mathbb{R}^n$  is defined as:

$$(\mathbf{g}.x)(u) = x(\mathbf{g}^{-1}u). \quad (10)$$

We can verify that the above definition satisfies the associativity:

$$(\mathbf{g}.\mathbf{h}.x)(u) = (\mathbf{h}.x)(\mathbf{g}^{-1}u) = x((\mathbf{gh})^{-1}u) = ((\mathbf{gh}).x)(u). \quad (11)$$

On a domain  $\Omega \in \mathbb{R}^n$ , we can define an  $n$ -dimensional real *representation* of a group  $\mathfrak{G}$  as a map  $\rho : \mathfrak{G} \rightarrow \mathbb{R}^{n \times n}$ , connecting each  $\mathbf{g} \in \mathfrak{G}$  to an invertible matrix  $\rho(\mathbf{g})$ , and satisfying the associativity  $\rho(\mathbf{gh}) = \rho(\mathbf{g})\rho(\mathbf{h})$  for all  $\mathbf{g}, \mathbf{h} \in \mathfrak{G}$ .

For a symmetry group  $\mathfrak{G}$  we have the definitions of  $\mathfrak{G}$ -invariant and  $\mathfrak{G}$ -equivariant. A function  $f : \mathcal{X}(\Omega) \rightarrow \mathcal{Y}$  is  $\mathfrak{G}$ -invariant if  $f(\rho(\mathbf{g})x) = f(x)$  for all  $\mathbf{g} \in \mathfrak{G}$  and  $x \in \mathcal{X}(\Omega)$ , and it is  $\mathfrak{G}$ -equivariant if  $f(\rho(\mathbf{g})x) = \rho(\mathbf{g})f(x)$ .

By stacking multiple neural network layers(functions), each of which satisfies either equivariance or invariance under some symmetry group, we can incorporate the knowledge of symmetries of domains into the resulting network. Table 1 [137] connects some neural network architectures with their corresponding symmetry groups and domains.

Architecture	Symmetry group $\mathfrak{G}$	Domain $\Omega$
CNN	Grid	Translation
Spherical CNN [138]	Sphere / SO(3)	Rotation SO(3)
Intrinsic Mesh CNN [128, 129]	Manifold	Isometry Iso( $\Omega$ ) / Gauge symmetry SO(2)
GNN	Graph	Permutation $\Sigma_n$
Deep Sets [139]	Set	Permutation $\Sigma_n$
Transformer	Complete Graph	Permutation $\Sigma_n$
LSTM	1D Grid	Time warping

Table 1: Neural network architectures, symmetry groups and domains.

### 3.3 Numerical Methods for Partial Differential Equations (PDEs)

A generic form of PDEs describing the evolution of a continuous value  $v(x, t)$  is as follows:

$$\frac{\partial v}{\partial t} = F(t, x, v, \frac{\partial v}{\partial x_i}, \frac{\partial^2 v}{\partial x_i \partial x_j}, \dots). \quad (12)$$

**Finite Difference Method** Finite difference methods approximate spatial derivatives on the right hand side of Eq 12 at a certain point  $x_0$  as a linear combination of function values at  $N$  neighbors of  $x_0$ . In 1D case, the approximation of the  $l$ -th order spatial derivative can be formulated as (here we omit  $t$ ):

$$\left. \frac{\partial^l f}{\partial x^l} \right|_{x=x_0} = \sum_{n=1}^N \alpha_n f(x_0 + h_n). \quad (13)$$

$\alpha_1, \alpha_2, \dots, \alpha_n$  can be solved via expanding  $\alpha_1 f(x_0 + h_1), \alpha_2 f(x_0 + h_2), \dots, \alpha_N f(x_0 + h_N)$  at  $x_0$  to the  $N$ -th order:

$$\begin{cases} \alpha_1 f(x_0 + h_1) &= \alpha_1 \sum_{k=0}^{N-1} \frac{1}{k!} \frac{\partial^k f}{\partial x^k} h_1^k + O(h_1^{N-1}) \\ \dots & \\ \alpha_N f(x_0 + h_N) &= \alpha_N \sum_{k=0}^{N-1} \frac{1}{k!} \frac{\partial^k f}{\partial x^k} h_N^k + O(h_N^{N-1}) \end{cases}, \quad (14)$$

which can be summed together as

$$\sum_{n=1}^N \alpha_n f(x_0 + h_n) = \sum_{k=0}^{N-1} \left( \frac{1}{k!} \sum_{i=1}^N \alpha_i h_i^k \right) \frac{\partial^k f}{\partial x^k} + O(\max_{i \in [N]} h_i^{N-1}) \quad (15)$$



To solve  $\alpha_1, \alpha_2, \dots, \alpha_n$  for a certain order  $l$ , we can let the multiplier of  $\frac{\partial^k f}{\partial x^k}$  be 1 and others be 0, and solve following linear equations:

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ h_1^1 & h_2^1 & \dots & h_N^1 \\ \vdots & \vdots & \ddots & \vdots \\ h_1^l & h_2^l & \dots & h_N^l \\ \vdots & \vdots & \ddots & \vdots \\ h_1^{N-1} & h_2^{N-1} & \dots & h_N^{N-1} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} = \begin{bmatrix} 0! \cdot 0 \\ 1! \cdot 0 \\ \vdots \\ l! \cdot 1 \\ \vdots \\ (N-1)! \cdot 0 \end{bmatrix}, \quad (16)$$

the solution  $\alpha_1^*, \dots, \alpha_N^*$  of which satisfies

$$\frac{\partial^k f}{\partial x^k} = \sum_{n=1}^N \alpha_n^* f(x_0 + h_n) + O(\max_{i \in [N]} h_i^{N-1}). \quad (17)$$

The above method can be extended to multi-dimensional cases with multi-dimensional Taylor expansion. After approximating spatial derivatives on the left hand side, Eq 12 can be solved with standard techniques of numerical integration. While Eq 16 can fully determine all coefficients, recent works [140, 141] relax it by removing some constraining equations and use neural networks to learn undetermined coefficients to combine prior knowledge with stronger expressivity.

**Finite Volume Method** Unlike finite difference method, finite volume method represent the value  $v(x, t)$  with its averages over a grid cell:  $v_i(t) = \Delta x^{-1} \int_{x_i - \Delta x/2}^{x_i + \Delta x/2} v(x', t) dx'$ . Similar to the finite difference method, spatial derivatives can also be estimated as the linear combination of averaged cell values:  $\frac{\partial^l v}{\partial x^l} = \sum_{i=1}^N \alpha_i^{(l)} v_i$ , and the coefficients are estimated in the same way as finite difference method.

For finite volume methods, the equation must be able to be rewritten as:

$$\frac{\partial v}{\partial t} = \frac{\partial}{\partial x} J(t, x, v, \frac{\partial v}{\partial x_i}, \frac{\partial^2 v}{\partial x_i \partial x_j}, \dots), \quad (18)$$

where  $J$  is called a *flux* and has an analytical form derived from the original PDE. The evolution along time can be carried out in following steps: (1) spatial derivatives are estimated on the boundary between grid cells; (2) the flux  $J$  is calculated with approximated derivatives using its analytical form; (3) the temporal derivative of averaged cell values is calculated via subtracting  $J$  at the cell's left and right boundaries. The final step can be conducted with techniques that promote stability, such as monotone numerical fluxes and Godunov flux [43]. Recent works have integrated data-driven models in step (1) to improve the estimation of spatial derivatives. For example, [43] estimate coefficients with the combination of neural network results and numerical results, and [142] uses policies trained via reinforcement learning to determine coefficients for estimation.

**Finite Element Method** Finite element method divide a space into small parts (elements) and approximate the PDE on each. As illustrated in [143], we use the Poisson's equation to show the basic idea of finite element method.

The formulation of the Poisson's equation is as follows:

$$\begin{aligned} -\Delta u &= \lambda \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \Gamma. \end{aligned} \quad (19)$$

In finite element method, we multiply a *test function*  $v$  on both sides, integrate over  $\Omega$ , and use integration in parts to derive the *weak formulation*:

$$a(u, v) = l(\lambda, v), \text{ for all } v \in \mathbb{V}, \quad (20)$$

$$\text{where } a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v dx, \quad l(\lambda, v) = \int_{\Omega} \lambda v dx. \quad (21)$$

Then we construct a finite-dimensional subspace  $\mathbb{V}_h \subset \mathbb{V}$ , where  $\mathbb{V}_h$  is a piece-wise polynomial function space spanned via  $\{\phi_1, \phi_2, \dots, \phi_n\}$ . To solve the weak formulation, we transform it to the *Galerkin weak formulation*, which is an approximation of Eq 20:

$$\mathbf{A} \mathbf{u} = \mathbf{f}, \quad (22)$$

where  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $A_{ij} = a(\phi_i, \phi_j)$ ,  $\mathbf{u} \in \mathbb{R}^n$  is the solution vector and  $\mathbf{f} \in \mathbb{R}^n$  is the source vector with  $f_i = l(\phi_i, \lambda)$ . The solution of Eq 22 gives the best solution of the PDE in  $\mathbb{V}_h$ .

### 3.4 Koopman Theory

Given a non-linear dynamic system with its state vector at time  $t$  denoted as  $\mathbf{s}_t \in \mathbb{R}^m$ . The system can be described as  $\mathbf{s}_{t+1} = F(\mathbf{s}_t)$ . As defined in [55], the Koopman operator  $\mathcal{K}_F$  is a linear transformation defined on a function space  $\mathcal{F}$  by  $\mathcal{K}_F g = g \circ F$  for every  $g : \mathbb{R}^m \rightarrow \mathbb{R}$  that belongs to the infinite-dimensional Hilbert space  $\mathcal{F}$ . With the definition we have  $\mathcal{K}_F g(\mathbf{s}_t) = g \circ F(\mathbf{s}_t) = g(\mathbf{s}_{t+1})$ .

The Koopman theory [55] guarantees the existence of  $\mathcal{K}$ , but in practice we often assume the existence of an invariant finite-dimensional subspace  $\mathcal{G}$  of  $\mathcal{F}$  spanned by  $k$  bases  $\{g_1, g_2, \dots, g_k\}$ . Define  $\mathbf{g}_t = [g_1(\mathbf{s}_t), g_2(\mathbf{s}_t), \dots, g_k(\mathbf{s}_t)]^T$  and  $\mathbf{g}_{t+1} = [g_1(\mathbf{s}_{t+1}), g_2(\mathbf{s}_{t+1}), \dots, g_k(\mathbf{s}_{t+1})]^T$ , under the assumption we have  $\mathbf{g}_t, \mathbf{g}_{t+1} \in \mathcal{G}$  and there exists a Koopman matrix  $\mathbf{K} \in \mathbb{R}^{k \times k}$  s.t.  $\mathbf{g}_{t+1} = \mathbf{K} \mathbf{g}_t$ . The key problem is to find the pair of mappings between the state space  $\mathbb{R}^m$  and the invariant subspace  $\mathcal{G} \in \mathbb{R}^k$ :  $\mathbf{g} : \mathbb{R}^m \rightarrow \mathbb{R}^k$  and  $\mathbf{g}^{-1} : \mathbb{R}^k \rightarrow \mathbb{R}^m$ . Recent works [61, 59, 60] utilize neural networks as  $\mathbf{g}$  and  $\mathbf{g}^{-1}$  to find the mappings in a data-driven way.

## 4 Methods of PIML

Typical solutions to a problem with machine learning involve three key parts: data, model, and optimization, each of which can be integrated with prior physics knowledge. In the following parts, we will introduce existing techniques of incorporating physics knowledge to each part respectively. However, we should notice that these techniques are not mutually exclusive: physics knowledge can be integrated to more than one parts of the machine learning solution.

Integration Method Knowledge Form	Data			Model		Optimization	
	Simulation Data	Transfer Learning	Auxiliary Tasks	Computation Graph	Fusion	Knowledge Based Loss	Regularization
Domain Knowledge in Analytical Form	[26][31][30][144] [28][29][30][31] [32][33][34][45] [46][47][63]	[145][146][147][148]	[116]	[24][148][122]	[149][34]	[37][38][39][150] [23][151][148][152][153]	[133][135][136]
Energy Conservation Law				[117][118][109][154][134]			
Symmetry				[155][116][156][157] [158][159][160][161] [27][128][129][130] [131][132]			
Numerical Methods for PDEs			[162][163]	[140][141][164][123] [43][142][165][143] [166][25][44]	[167][140][104][123][124]		
Koopman Theory				[61][168][59][60]			

Table 2: Existing works classified based on forms of physics knowledge and integration methods. For certain types of knowledge forms, inductive bias integrated in reusable computation graphs have advantages over integration with data and optimization.

In Table 2, we classify existing works based on the forms of physics knowledge and the integration methods. We notice that for domain knowledge taking analytical forms, existing works integrate the knowledge into all three aspects including data, model, and optimization. However, research works on integrating other general types of physics knowledge, including energy conservation law, symmetry, numerical methods for PDEs, and Koopman theory, mainly focus on incorporating corresponding knowledge into computation graphs. The main reason is that such general physics knowledge is possible to be transformed to inductive bias in reusable network architectures, which has advantages over data augmentation and physics knowledge based loss functions in terms of prediction performance and data efficiency [137]. This is due to that (1) general physics knowledge applies to various problems and thus leads to general network architectures, and (2) has simpler forms that can be translated to combinations of a limited number of differentiable operators compared to complex numerical simulators designed for domain-specific problems such as weather and turbulence.

### 4.1 Physics-Informed Data Enhancement

#### 4.1.1 Data Generated from Simulation

The Universal Approximation Theorem [108] guarantees that multilayer neural networks with as few as one hidden layer can approximate any continuous function from one finite dimensional space to another to any desired degree of accuracy. Therefore, one straight forward method to incorporate physics knowledge into neural networks is to generate training data from the desired physics knowledge. When the data amount is abundant, the neural network is expressive enough and trained properly, the trained neural network will be able to approximate the behavior of the physics knowledge governing the data generation. Usually the neural network models can be accelerated with hardware such as GPU, FPGA and ASIC, thus they can act as good surrogate models with much lower computation costs while maintaining comparable accuracy to the numerical simulation.

[26] present a benchmark dataset from results of numerical global weather simulation with high computation cost, and provide scores of deep learning models. Results show that data-driven models trained with simulation data can achieve competitive results compared to numerical solutions while enjoying lower computation costs. [31, 30, 144] utilize different variants of message passing graph neural networks [169, 113, 114] respectively and train them with the simulation data of particle systems. Compared to the simulation methods generating datasets, the trained surrogate model can accurately predict the dynamics of a wide range of physical systems within the same architecture, and runs orders of magnitude faster.

#### 4.1.2 Transfer Learning

For real-world tasks suffering from data limitation or labeling difficulties, the integration of prior physics knowledge about the tasks is critical. Simulators constructed with such physics knowledge can provide large amount of data with high label quality, and can be used to pre-train models. However, the differences between the target real-world data distributions and simulation data distributions call for techniques of transfer learning to mitigate the gap.

[145] present a system for training object detection models with synthetic images. This work adopts the technique of domain randomization, where important parameters of simulators - including lighting, pose, object textures - are randomized in non-realistic ways to encourage the model to learn essential features. [146] uses off-the-shelf simulators to render synthetic data for training a grasping system together with pixel-level domain adaptation between synthetic images and real-world ones. The utilization of synthetic data reduces the required amount of real-world samples by up to 50 times. [147] transfer driving policies trained from simulation to reality via modularity and abstraction, where the driving policy is exposed to segmentation results of input scenes and target way points, instead of raw perceptual input or low-level vehicle dynamics. [148] incorporate the task-specific prior knowledge into the model and pre-train it with synthetic data generated by imperfect physical models, which allows the model to get close enough to the target solution and only a small amount of real-world data is needed for refining.

#### 4.1.3 Multitask Learning and Meta Learning with Auxiliary Tasks

Synthetic data generated from physics-based simulators can also be used to construct auxiliary learning tasks for improving the model’s performance on the target task with techniques of multitask learning and meta learning. [116] constructs the auxiliary task as the prediction of interatomic forces for the main task of molecular energy prediction. The labels of the auxiliary task and the main task are generated from simulation at the same time, while the prediction of the auxiliary task is produced via differentiating the energy prediction model. Both tasks are used to train the model simultaneously. [162] adopts the multitask learning scheme by learning shared representations between multiple related PDEs, which are generated by varying coefficients, for better generalizability of the proposed neural network based PDE solver. [163] proposes a spatiotemporal forecasting model with decoupled spatial and temporal modules, where the spatial module is PDE-independent and are trained via model agnostic meta learning (MAML) [170] for fast adaptation on new tasks, while the task-dependent temporal module is trained from scratch for each task.

### 4.2 Physics-Informed Neural Network Architecture Design

#### 4.2.1 Physics-Informed Computation Graph

A typical way of physics-informed neural network architecture design is to design computation graphs that mimic the behavior of physics knowledge based methods. While the specific method is highly dependent on the physics knowledge, a general idea is to start from some existing physics based solution, then replace difficult-to-estimate variables with outputs of neural networks, or relax some fixed parameters by enabling them to adapt to the data. In following paragraphs, we will introduce several knowledge-specific neural network designs as well as techniques to directly fuse deep learning models with physics based solutions as a hybrid model.

**Energy Conservation Laws** As introduced in Sec 3.1, Lagrangian and Hamiltonian mechanics are powerful in enforcing energy conservation laws, thus a series of recent works develop neural network architectures based on them to incorporate the energy conservation property. [117] proposes a network topology named Deep Lagrangian Networks (DeLaN) encoding the Lagrange-Euler PDE originating from Lagrangian Mechanics, which can be trained with standard optimizers while maintaining physical plausibility. [118] designs Lagrangian Neural Networks (LNNs) to model arbitrary Lagrangian functions via neural networks, and solve the dynamics of the system with a numerical expression derived from the Euler-Lagrangian equation, where gradients from auto differentiation are utilized. Similarly, [109] (HNN) models the Hamiltonian function with a neural network. The derivatives of spatial coordinates and momentum with respect to time are derived from Eq 8. [154] proposes Neural Hamiltonian Flow (NHF), which is a powerful normalising flow model using Hamiltonian dynamics as the invertible function to model expressive densities. In NHF,

the density is decomposed into the "coordinate" part and the "momentum" part in the hidden space, both of which are then propagated with Eq 8. The propagation is (1) invertible and (2) has the volume ("energy" with respect to the hidden space) preserving property, which satisfies the requirement of normalising flows. Compared to other flow-based approaches, NHF enjoys higher computational efficiency since it avoids the expensive step of calculating the trace of Jacobians. [134] designs the computation graph of the proposed neural network following the Hamiltonian dynamics with control to incorporate the corresponding inductive bias. It further proposes a parameterization that can enforce the Hamiltonian mechanics with coordinates embedded in a high-dimensional space or velocity data instead of momentum.

**Symmetry** Table 1 in Sec 3.2 has shown the connection between some widely used neural network architectures and the corresponding symmetry groups. Here we introduce methods incorporating other types of symmetry groups into neural network architectures. Based on the representations of symmetry groups adopted, all the methods we introduce can be categorized to methods using (1) invariant treatment of coordinates (2) irreducible representations or (3) regular representations.

**Invariant Treatment of Coordinates** Depending on the symmetry group, spatial coordinates should be properly processed instead of being used as raw inputs. [155] develops equivariant message passing to  $E(3)$  via letting messages passed among nodes only depend on distances, which follows the property that  $E(3)$  preserves distances between nodes. The same technique is also used in [116]. [156] propose a set of transformation invariant and equivariant GNN models by tweaking the definition of an adjacency matrix named *isometric adjacency matrix*, which can be viewed as a weighted adjacency matrix for each direction and reflects spatial information.

**Irreducible Representations** All elements of a roto-translation group can be transformed into an irreducible form: a vector that is rotated by a block diagonal matrix. The full set of equivariant mappings for some symmetry group can be solved with equivariance constraint over convolution kernels. The solutions form a linear combination of equivariant basis matrices, which can be used for equivariant convolutions. [157] gives a general solution of the kernel space constraint for arbitrary representations of the Euclidean group  $E(2)$  and its subgroups, which forms a wide range of equivariant network architectures. [158] develops convolution filters locally equivariant to 3D rotations, translations, and permutations, which are built from spherical harmonics. [159] further enhances [158] with the self-attention mechanism.

**Regular Representations** Regular representation approaches store copies of latent feature embeddings for all elements of a symmetry group. To mitigate this computational burden, some recent works such as [160, 161] use *Lie groups* as the tool for rapid prototyping across various symmetry groups. Only the exponential and logarithm maps are required for incorporating equivariance to a new symmetry group.

**Numerical Methods** Sec 3.3 introduces several numerical methods for solving PDEs. In this paragraph, we introduce some recent works integrating numerical solutions for each method.

**Finite Difference Method** [140, 141] propose learnable differential operators by learning convolution kernels to approximate unknown nonlinear responses in PDEs. All kernels are properly constrained by fully exploiting the relation between the orders of differential operators and the orders of sum rules of filters, which originates from wavelet theory. These constraints ensure both the model's ability to identify PDEs and its expressivity. [164] proposes an efficient convolution kernel on unstructured grids of spherical signals using parameterized finite difference operators. [123] leverages differences of sparsely available data from physical systems via the spatial difference layer (SDL). SDL is inspired by finite difference operators on graph and triangulated mesh and replaces fixed parameters in these operators with the output of a GNN capturing spatial information of the input data.

**Finite Volume Method** [43] adopts CNNs to generate coefficients for approximating spatial derivatives, followed by the standard finite volume method. CNNs are optimized end-to-end to best satisfy the equations on low resolution grids, and produce accurate numerical results: it can be integrated at resolutions 4-8x coarser than is possible with standard finite volume methods. [142] creates new PDE solvers based on the WENO scheme [171] via generating its coefficients from a learned policy network trained with reinforcement learning.

**Finite Element Method** [165] mimics the behavior of finite element analysis: it assigns nodes of a GNN to selected spatial locations and uses message passing on the graph to model the relationship between an initial function and a resulting function defined in the same space. Both the locations of nodes and their connectivity can be optimized to focus on the most important parts of the space. [143] proposes a two-stage optimization framework for PDE-constrained optimization problem. At the first stage, the framework obtains a surrogate model to prediction solutions of finite element method directly from control parameters. At the second stage, the framework performs gradient-based PDE-constrained optimization. [166] introduces convolution operators on unstructured point clouds based on Generalized

Moving Least Squares (GMLS), which is a non-parametric technique in finite element method [172] for estimating linear bounded functionals from scattered data.

**Koopman Theory** [61] first combines deep learning models with the Koopman operator. It utilizes the power of deep learning to identify nonlinear coordinates on which the dynamics are globally linear using the Koopman operator. The resulting method benefits from both the power and generality of deep learning models and the physical interpretability of Koopman embeddings. [168] proposes minimization of the residual sum of squares of linear least-squares regression to estimate the encoders and decoders that maps data into the Koopman invariant subspaces where the linear regression fits well. [59] extends deep learning based Koopman operators to scenarios with multiple objects. It adopts GNNs to encode object-centric states and uses a block-wise linear state transition matrix (Koopman matrix) to enforce the shared structure among objects. [60] incorporates the consistency by penalizing the consistency mismatch of forward and backward Koopman matrices.

#### 4.2.2 Fusion of Deep Learning and Physics-Based Modules

In addition to previously introduced methods that mimic the behaviour of physics based solutions in the computation graph design of neural network layers, deep learning and physics-based methods can also be fused in a higher level: modules constructed with standard neural network blocks and physics rules work interactively but only expose input/output to each other.

[149] adopts a graphical model derived from equations of motion in a physics model to predict the next future state while adding it with the predicted residual part from a GNN module. [167] presents HybridNet, a framework combining data-driven deep learning and model-driven computation for reliable spatiotemporal evolution prediction. The deep learning part, Convolutional LSTM (ConvLSTM) works as the backbone to predict the evolution of external input to the system. The model-driven part, Cellular Neural Network (CeNN), transforms numerical computation in PDE solvers and is able to infer unknown physical parameters. [34] proposes a hybrid approach for fluid flow prediction containing two components: one GNN-based module operating directly on the original fine-grained non-uniform mesh used in CFD, and one CFD solver operating on a much coarser resolution. The output of the CFD solver is upsampled to the fine-grained mesh and then concatenated to the hidden embeddings from GNN layers. The hybrid model generalizes better than pure GNN-based approaches and is still faster than directly running CFD simulation on the original mesh. [140, 104] uses constrained convolution kernels to extract approximated spatial derivatives as features, which are fed as the input to the following neural network layers that capture unknown dynamics and give the final prediction.

### 4.3 Physics-Informed Optimization

Prior physics knowledge can also be integrated into the optimization process in the form of loss functions directly derived from task-specific knowledge or regularizations from physics principles. The integration of physics knowledge in optimization targets reshapes the optimization space and encourages the training process to converge to physical plausible solutions.

#### 4.3.1 Task-Specific Knowledge Based Loss Terms

A variety of physics knowledge can be described in the form of PDEs, which provides connections between the spatial and temporal derivatives as well as constraints of values of interest on boundaries. [37, 38, 39, 150] adopts multi-layer perceptrons to directly model the mapping from input spatial and temporal coordinates to the value of PDE solutions. Instead of only minimizing the prediction error between the output and solutions from numerical methods, the series of works adds loss terms enforcing the equation structure, which penalizes the violation of PDEs using derivatives from auto-differentiation of the neural network solution. [23] optimizes the model for super-resolution of turbulent flows by minimizing a weighted combination of two losses: one is the norm of the difference between predictions and ground truth values, the other is the norm of residues of the governing PDEs. [151] predicts the traffic flow with a neural network taking time and coordinates as input, and constructs physical discrepancy loss terms with an existing second-order traffic model. [148] trains the proposed lake temperature prediction model with generalized loss function to include the physical consistency-based penalty, which encourages the consistency between lake energy and energy fluxes.

#### 4.3.2 Regularization

[133] adopts a deep neural network to predict the unknown disturbance forces in the controller of drones. To guarantee the system’s stability, the authors first derive the overall stability and robustness requirement indicating constraints on



the Lipschitz constant, then minimizes its upper bound - the spectral norm of weights in each layer - together with the prediction error in the optimization. [135, 136] both augment purely physics laws/rule-based prediction models with learnable control signals to mitigate the approximation errors of prior knowledge. While they differ in terms of processes for solving PDEs, both have constraints minimizing the norm of control signals, which originates from the least action principle [173].

## 5 Challenges and Future Directions

### 5.1 Challenge 1: Handcrafted Selection of Physics Knowledge for Incorporation

Existing works require expertise of the domain-specific knowledge of tasks to incorporate the most appropriate physics knowledge. While this serves the purpose of leveraging domain-knowledge to mitigate the deficiencies of pure data-driven methods, it lacks the flexibility of identifying the correct physics knowledge depending on the task. For example, [23] directly uses the governing equation of the Rayleigh-Benard instability problem as the prior knowledge for turbulence super-resolution, while [24] chooses the derived Hybrid RANS(Reynolds-averaged Navier-Stokes)-LES(Large Eddy Simulation) Coupling method for turbulence prediction. Although both super-resolution and prediction tasks are defined on the same turbulence system governed by the same physics laws, choosing the form of physics laws (the original form/the derived approximation form) to incorporate is heuristic.

**Research Direction 1: Automatic Identification of Proper Physics Knowledge To Incorporate** A promising research direction is to reach the middle ground between the domain-specific knowledge and the pure data-driven way. Here we discuss some potential approaches to realizing it.

**Neural Architecture Search (NAS)** The development of NAS allows the automatic design of neural network architectures and NAS methods have outperformed manually designed architectures on many tasks including image classification and semantic segmentation [174]. By restricting the available physics knowledge within a pre-selected search space and enabling the model to discover the optimal knowledge or combination of knowledge using NAS techniques, the resulting architecture can reach the balance between exploiting prior knowledge and adapting to observed data. [175] presents a neural block dynamics design space of neural network components that encompasses various state-space models as the search space for NAS, and models given by NAS in such a space achieve highly accuracy with physically consistent results.

**Automatic Modularization of Network Architectures** Modularization of network architectures plays a key part in physics-informed network architecture design. For example, DeLaN [117] and HNN [109] contain modules estimating the Lagrangian/Hamiltonian function of the system and following modules deriving the prediction. [43, 123] separate modules approximating spatial and temporal derivatives in the network. Modularity provides better generalization ability [115], and some modules can further be trained in multiple tasks and benefit the overall performance on all tasks [176, 163]. However, existing works still require a certain selection of physics knowledge to guide the modularization. Some recent works have started exploring discovering functional modules automatically during the training process. For example, [177] divides models into reusable modules and task-specific modules by estimating the variance of parameters across tasks. [178] lets multiple groups of recurrent cells compete with each other so that they are only updated at time steps where they are most relevant. This enables the model to learn modular structures and leads to improved generalization.

### 5.2 Challenge 2: Lack of Benchmarks and Evaluations of PIML Methods

Comprehensive benchmarks have shown as great boosters for the development of corresponding research areas. Examples include ImageNet Large Scale Visual Recognition Challenge(ILSVRC) [179] and Common Objects in Context(COCO) [180] from computer vision, Workshop on Statistical Machine Translation (WMT) [181] and Stanford Question Answering Dataset (SQuAD) [182] from natural language processing. However, due to the complexity and heterogeneity of problem settings, PIML still lacks comprehensive benchmarks for evaluating various methods of knowledge integration, which creates barriers in the development of PIML. First, most problems in PIML come from physics or engineering applications, where acquiring the data and formalizing the task can be challenging for researchers without domain knowledge and experience. Second, existing works such as [117, 109, 123, 60, 140, 43, 34] heavily rely on heterogeneous domain-specific datasets, which greatly increases the difficulty of fairly comparing different PIML methods.

**Research Direction 2: Comprehensive Benchmarks for PIML Methods** Constructing comprehensive benchmarks for PIML is of great need for boosting its development. According to the above discussion, ideal benchmarks

(1) must provide publicly available and organized datasets, and formulate benchmark tasks as typical machine learning tasks, such as classification and regression; (2) must be general enough to accommodate various PIML methods as well as data-driven and pure physics-based methods. Recent development along this direction includes **WeatherBench** [26] and **Open Graph Benchmark (OGB)** [183]. WeatherBench provides processed weather data together with clearly defined tasks and evaluation metrics for medium-range weather forecasting. It also presents performance of purely data-driven models and numerical models as baselines. OGB offers data of protein structures and molecule structures, and construct node/link/graph property prediction tasks, where performance of PIML methods and data-driven baselines can be directly compared.

### 5.3 Challenge 3: Suboptimal Existing Neural Network Architectures and Optimization Methods for PIML

Established theories and empirical conclusions of neural network architectures and optimization methods are mostly developed in areas where neural network methods first gain advantages, such as computer vision and natural language processing. However, they may no longer be effective in PIML. The reason is that PIML methods usually involve explicit use of gradients in forwarding processes and objective functions, leading to the existence of high-order derivatives in the backward process, which shapes optimization spaces significantly different from typical deep learning models. For example, [118] notices that regular parameter initialization methods such as Kaiming [184] and Xavier [185] are insufficient since the unusual optimization objective is very nonlinear. [186] also empirically demonstrates that the widely used Rectified Linear Unit (ReLU) activation is not effective in the physics-informed PINN architecture proposed by [37, 38].

**Research Direction 3: Novel Neural Network Designs for PIML** The drastic differences in network architecture and objectives between PIML and conventional deep learning tasks signify the importance of novel neural network designs for PIML from both the architecture and the optimization aspects.

From the architecture perspective, since many PIML methods involve the utilization of gradients from the auto-differentiation of neural networks, designing new architectures/components better preserving gradient information is one promising direction. [187] leverages periodic activation functions for implicit neural representations and demonstrate that they are ideally suited for representing complex natural signals and their spatial/temporal derivatives.

From the optimization perspective, multiple objectives (including both the task objective and the physics-informed constraints) may contradict with each other and lead to suboptimal results under vanilla optimization methods. [188] notices the discrepancy of updating directions between the boundary constraint loss and the approximation loss in PINN [37, 38], and proposes the dynamic pulling method (DPM) to align their updating directions, which significantly improves the extrapolation performance of PINNs.

## 6 Summary

In this paper, we provide a thorough and comprehensive survey of existing works in PIML. We summarize them from three aspects: (1) motivations of PIML, (2) physics knowledge in PIML, (3) methods of knowledge integration in PIML. In the end, we discuss existing challenges of PIML and indicate potential future research directions accordingly. We expect the paper can serve as the guide for PIML users to select proper physics knowledge and appropriate integration methods, as well as the guide for PIML researchers to identify existing gaps and promising research directions.

## References

- [1] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

- [6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *International Conference on Learning Representations*, 2013.
- [7] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.
- [8] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 2014.
- [9] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.
- [10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [11] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. PMLR, 2016.
- [12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017.
- [13] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
- [14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [15] William W Hsieh. *Machine learning methods in the environmental sciences: Neural networks and kernels*. Cambridge university press, 2009.
- [16] Željko Ivezić, Andrew J Connolly, Jacob T VanderPlas, and Alexander Gray. *Statistics, data mining, and machine learning in astronomy: a practical Python guide for the analysis of survey data*, volume 1. Princeton University Press, 2014.
- [17] Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering*, 29(10):2318–2331, 2017.
- [18] Anuj Karpatne, Imme Ebert-Uphoff, Sai Ravela, Hassan Ali Babaie, and Vipin Kumar. Machine learning for the geosciences: Challenges and opportunities. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1544–1554, 2018.
- [19] J Nathan Kutz. Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814:1–4, 2017.
- [20] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, et al. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.
- [21] Zhen Wang, Haibin Di, Muhammad Amir Shafiq, Yazeed Alaudah, and Ghassan AlRegib. Successful leveraging of image processing and machine learning in seismic structural interpretation: A review. *The Leading Edge*, 37(6):451–461, 2018.
- [22] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfommer, Annika Pick, Rajkumar Ramamurthy, et al. Informed machine learning—a taxonomy and survey of integrating knowledge into learning systems. *arXiv preprint arXiv:1903.12394*, 2019.
- [23] Soheil Esmailzadeh, Kamyar Azizzadenesheli, Karthik Kashinath, Mustafa Mustafa, Hamdi A Tchelepi, Philip Marcus, Mr Prabhat, Anima Anandkumar, et al. Meshfreeflownet: a physics-constrained deep continuous space-time super-resolution framework. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15. IEEE, 2020.
- [24] 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.

- [25] Karthik Kashinath, Philip Marcus, et al. Enforcing physical constraints in cnns through differentiable pde layer. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.
- [26] Stephan Rasp, Peter D Dueben, Sebastian Scher, Jonathan A Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.
- [27] Jonathan A Weyn, Dale R Durran, and Rich Caruana. Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. *Journal of Advances in Modeling Earth Systems*, 12(9):e2020MS002109, 2020.
- [28] Peter Grönquist, Chengyuan Yao, Tal Ben-Nun, Nikoli Dryden, Peter Dueben, Shigang Li, and Torsten Hoefler. Deep learning for post-processing ensemble weather forecasts. *Philosophical Transactions of the Royal Society A*, 379(2194):20200092, 2021.
- [29] K Kashinath, M Mustafa, A Albert, JL Wu, C Jiang, S Esmailzadeh, K Azizzadenesheli, R Wang, A Chattopadhyay, A Singh, et al. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, 379(2194):20200093, 2021.
- [30] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pages 8459–8468. PMLR, 2020.
- [31] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations*, 2019.
- [32] Benjamin Ummenhofer, Lukas Prantl, Nils Thuerey, and Vladlen Koltun. Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*, 2019.
- [33] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021.
- [34] Filipe de Avila Belbute-Peres, Thomas Economou, and Zico Kolter. Combining differentiable pde solvers and graph neural networks for fluid flow prediction. In *International Conference on Machine Learning*, pages 2402–2411. PMLR, 2020.
- [35] James P Crutchfield and BS McNamara. Equations of motion from a data series. *Complex systems*, 1(417-452):121, 1987.
- [36] Ioannis G Kevrekidis, C William Gear, James M Hyman, Panagiotis G Kevrekidis, Olof Runborg, Constantinos Theodoropoulos, et al. Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.
- [37] 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.
- [38] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10566*, 2017.
- [39] Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *arXiv preprint arXiv:1801.06637*, 2018.
- [40] Martin Magill, Faisal Qureshi, and Hendrick W de Haan. Neural networks trained to solve differential equations learn general representations. *Advances in Neural Information Processing Systems*, 2018.
- [41] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.
- [42] Zongyi Li, Nikola Borislovov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, et al. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.
- [43] Yohai Bar-Sinai, Stephan Hoyer, Jason Hickey, and Michael P Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, 2019.
- [44] Kiwon Um, Robert Brand, Yun Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers. *Advances in Neural Information Processing Systems*, 2020.

- [45] Ehsan Sharifi, B Saghafian, and R Steinacker. Downscaling satellite precipitation estimates with multiple linear regression, artificial neural networks, and spline interpolation techniques. *Journal of Geophysical Research: Atmospheres*, 124(2):789–805, 2019.
- [46] Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R Ganguly. DeepSD: Generating high resolution climate change projections through single image super-resolution. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, pages 1663–1672, 2017.
- [47] Seungjoon Lee, Mahdi Kooshkbaghi, Konstantinos Spiliotis, Constantinos I Siettos, and Ioannis G Kevrekidis. Coarse-scale pdes from fine-scale observations via machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(1):013141, 2020.
- [48] Shing Chan and Ahmed H Elsheikh. Parametrization and generation of geological models with generative adversarial networks. *arXiv preprint arXiv:1708.01810*, 2017.
- [49] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.
- [50] EB Goldstein, G Coco, AB Murray, and MO Green. Data-driven components in a model of inner-shelf sorted bedforms: a new hybrid model. *Earth Surface Dynamics*, 2(1):67–82, 2014.
- [51] Noah D Brenowitz and Christopher S Bretherton. Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, 45(12):6289–6298, 2018.
- [52] Pierre Gentine, Mike Pritchard, Stephan Rasp, Gael Reinaudi, and Galen Yacalis. Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, 45(11):5742–5751, 2018.
- [53] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.
- [54] Xiaofeng Wang, Zhen Li, Mingjian Jiang, Shuang Wang, Shugang Zhang, and Zhiqiang Wei. Molecule property prediction based on spatial graph embedding. *Journal of chemical information and modeling*, 59(9):3817–3828, 2019.
- [55] Bernard 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, 1931.
- [56] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.
- [57] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.
- [58] I Kevrekidis, Clarence W Rowley, and M Williams. A kernel-based method for data-driven koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, 2016.
- [59] Yunzhu Li, Hao He, Jiajun Wu, Dina Katabi, and Antonio Torralba. Learning compositional koopman operators for model-based control. In *International Conference on Learning Representations*, 2020.
- [60] Omri Azencot, N Benjamin Erichson, Vanessa Lin, and Michael Mahoney. Forecasting sequential data using consistent koopman autoencoders. In *International Conference on Machine Learning*, pages 475–485. PMLR, 2020.
- [61] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):1–10, 2018.
- [62] D Xiao, CE Heaney, L Mottet, F Fang, W Lin, IM Navon, Y Guo, OK Matar, AG Robins, and CC Pain. A reduced order model for turbulent flows in the urban environment using machine learning. *Building and Environment*, 148:323–337, 2019.
- [63] Arvind T Mohan and Datta V Gaitonde. A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks. *arXiv preprint arXiv:1804.09269*, 2018.
- [64] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.
- [65] Judea Pearl. *Causality*. Cambridge university press, 2009.
- [66] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.



- [67] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [68] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.
- [69] Jakob Runge. Causal network reconstruction from time series: From theoretical assumptions to practical estimation. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):075310, 2018.
- [70] Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science Advances*, 5(11):eaau4996, 2019.
- [71] Jakob Runge, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D Mahecha, Jordi Muñoz-Marí, et al. Inferring causation from time series in earth system sciences. *Nature communications*, 10(1):1–13, 2019.
- [72] Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):312–340, 2019.
- [73] Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, pages 1595–1605. PMLR, 2020.
- [74] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 9472–9483. Curran Associates, Inc., 2018.
- [75] Miguel A Hernán and James M Robins. Causal inference, 2010.
- [76] James M Robins, Miguel Angel Hernan, and Babette Brumback. Marginal structural models and causal inference in epidemiology, 2000.
- [77] Garrett Fitzmaurice, Marie Davidian, Geert Verbeke, and Geert Molenberghs. Estimation of the causal effects of time-varying exposures. In *Longitudinal Data Analysis*, pages 567–614. Chapman and Hall/CRC, 2008.
- [78] Bryan Lim, Ahmed Alaa, and Mihaela van der Schaar. Forecasting treatment responses over time using recurrent marginal structural networks. *NeurIPS*, 18:7483–7493, 2018.
- [79] Peter Schulam and Suchi Saria. Reliable decision support using counterfactual models. *Advances in Neural Information Processing Systems*, 30:1697–1708, 2017.
- [80] Hossein Soleimani, Adarsh Subbaswamy, and Suchi Saria. Treatment-response models for counterfactual reasoning with continuous-time, continuous-valued interventions. *arXiv preprint arXiv:1704.02038*, 2017.
- [81] Judea Pearl. On measurement bias in causal inference. *arXiv preprint arXiv:1203.3504*, 2012.
- [82] Manabu Kuroki and Judea Pearl. Measurement bias and effect restoration in causal inference. *Biometrika*, 101(2):423–437, 2014.
- [83] Ioana Bica, Ahmed M Alaa, and Mihaela van der Schaar. Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders. *International Conference on Machine Learning*, 2020.
- [84] Tobias Hatt and Stefan Feuerriegel. Sequential deconfounding for causal inference with unobserved confounders. *arXiv preprint arXiv:2104.09323*, 2021.
- [85] Milan Kuzmanovic, Tobias Hatt, and Stefan Feuerriegel. Deconfounding temporal autoencoder: estimating treatment effects over time using noisy proxies. In *Machine Learning for Health*, pages 143–155. PMLR, 2021.
- [86] Ruoqi Liu, Changchang Yin, and Ping Zhang. Estimating individual treatment effects with time-varying confounders. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 382–391. IEEE, 2020.
- [87] Jing Ma, Ruocheng Guo, Chen Chen, Aidong Zhang, and Jundong Li. Deconfounding with networked observational data in a dynamic environment. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 166–174, 2021.
- [88] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, pages 4502–4510, 2016.
- [89] Michael B Chang, Tomer Ullman, Antonio Torralba, and Joshua B Tenenbaum. A compositional object-based approach to learning physical dynamics. *arXiv preprint arXiv:1612.00341*, 2016.
- [90] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems*, 28, 2015.

- [91] Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. *Advances in neural information processing systems*, 27, 2014.
- [92] Minne Li, Lisheng Wu, Jun Wang, and Haitham Bou Ammar. Multi-view reinforcement learning. *Advances in neural information processing systems*, 32, 2019.
- [93] Niklas Wahlström, Thomas B Schön, and Marc Peter Deisenroth. From pixels to torques: Policy learning with deep dynamical models. *arXiv preprint arXiv:1502.02251*, 2015.
- [94] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28, 2015.
- [95] He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. In *International conference on machine learning*, pages 1804–1813. PMLR, 2016.
- [96] Zheng Tian, Ying Wen, Zhichen Gong, Faiz Punakkath, Shihao Zou, and Jun Wang. A regularized opponent model with maximum entropy objective. *arXiv preprint arXiv:1905.08087*, 2019.
- [97] Kexin Yi\*, Chuang Gan\*, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. In *International Conference on Learning Representations*, 2020.
- [98] Jiajun Wu, Joseph J Lim, Hongyi Zhang, Joshua B Tenenbaum, and William T Freeman. Physics 101: Learning physical object properties from unlabeled videos. In *British Machine Vision Conference*, 2016.
- [99] Yunzhu Li, Antonio Torralba, Anima Anandkumar, Dieter Fox, and Animesh Garg. Causal discovery in physical systems from videos. *Advances in Neural Information Processing Systems*, 33:9180–9192, 2020.
- [100] Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf. Cophy: Counterfactual learning of physical dynamics. In *International Conference on Learning Representations*, 2020.
- [101] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *ICLR*, 2017.
- [102] Emily L Denton and vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [103] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *international conference on machine learning*, pages 3560–3569. PMLR, 2017.
- [104] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020.
- [105] Tian Ye, Xiaolong Wang, James Davidson, and Abhinav Gupta. Interpretable intuitive physics model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102, 2018.
- [106] Jiafei Duan, Samson Yu, Soujanya Poria, Bihan Wen, and Cheston Tan. Pip: Physical interaction prediction via mental imagery with span selection. *arXiv preprint arXiv:2109.04683*, 2021.
- [107] Steeven JANNY, Fabien Baradel, Natalia Neverova, Madiha Nadri, Greg Mori, and Christian Wolf. Filtered-cophy: Unsupervised learning of counterfactual physics in pixel space. In *International Conference on Learning Representations*, 2022.
- [108] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [109] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, pages 15353–15363, 2019.
- [110] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. *International Conference on Machine Learning*, 2018.
- [111] Guangyu Li, Bo Jiang, Hao Zhu, Zhengping Che, and Yan Liu. Generative attention networks for multi-agent behavioral modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7195–7202, 2020.
- [112] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

- [113] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.
- [114] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. *International Conference on Machine Learning*, 2018.
- [115] Keylu Xu, Jingling Li, Mozhi Zhang, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? In *ICLR*, 2020.
- [116] KT Schütt, P-J Kindermans, Huziel E Sauceda, S Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, pages 992–1002, 2018.
- [117] Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *International Conference on Learning Representations*, 2019.
- [118] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.
- [119] 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.
- [120] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*, 2021.
- [121] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 753–763, 2020.
- [122] Xingzhe He, Helen Lu Cao, and Bo Zhu. Advectivenet: An eulerian-lagrangian fluidic reservoir for point cloud processing. In *International Conference on Learning Representations*, 2020.
- [123] Sungyong Seo, Chuizheng Meng, and Yan Liu. Physics-aware difference graph networks for sparsely-observed dynamics. In *International Conference on Learning Representations*, 2020.
- [124] Valerii Iakovlev, Markus Heinonen, and Harri Lähdesmäki. Learning continuous-time {pde}s from sparse data with graph neural networks. In *International Conference on Learning Representations*, 2021.
- [125] Michaël Defferrard, Martino Milani, Frédéric Gusset, and Nathanaël Perraudin. Deepsphere: a graph-based spherical cnn. In *International Conference on Learning Representations*, 2020.
- [126] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.
- [127] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014.
- [128] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.
- [129] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proc. CVPR*, 2017.
- [130] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Michael M Bronstein, and Daniel Cremers. Anisotropic diffusion descriptors. In *Computer Graphics Forum*, volume 35, pages 431–441. Wiley Online Library, 2016.
- [131] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2019.
- [132] Pim De Haan, Maurice Weiler, Taco Cohen, and Max Welling. Gauge equivariant mesh cnns: Anisotropic convolutions on geometric graphs. In *International Conference on Learning Representations*, 2020.
- [133] Guanya Shi, Xichen Shi, Michael O’Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural lander: Stable drone landing control using learned dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9784–9790. IEEE, 2019.
- [134] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. In *International Conference on Learning Representations*, 2020.

- [135] Philipp Holl, Nils Thuerey, and Vladlen Koltun. Learning to control pdes with differentiable physics. In *International Conference on Learning Representations*, 2019.
- [136] 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.
- [137] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.
- [138] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018.
- [139] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in Neural Information Processing Systems*, 30, 2017.
- [140] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. *International Conference on Machine Learning*, 2018.
- [141] 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*, 399:108925, 2019.
- [142] Yufei Wang, Ziju Shen, Zichao Long, and Bin Dong. Learning to discretize: solving 1d scalar conservation laws via deep reinforcement learning. *arXiv preprint arXiv:1905.11079*, 2019.
- [143] Tianju Xue, Alex Beatson, Sigrid Adriaenssens, and Ryan Adams. Amortized finite element analysis for fast pde-constrained optimization. In *International Conference on Machine Learning*, pages 10638–10647. PMLR, 2020.
- [144] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2020.
- [145] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977, 2018.
- [146] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4243–4250. IEEE, 2018.
- [147] Matthias Mueller, Alexey Dosovitskiy, Bernard Ghanem, and Vladlen Koltun. Driving policy transfer via modularity and abstraction. In *Conference on Robot Learning*, pages 1–15. PMLR, 2018.
- [148] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan S Read, Jacob A Zwart, Michael Steinbach, and Vipin Kumar. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ACM/IMS Transactions on Data Science*, 2(3):1–26, 2021.
- [149] Victor Garcia Satorras, Zeynep Akata, and Max Welling. Combining generative and discriminative models for hybrid inference. *Advances in Neural Information Processing Systems*, 32:13825–13835, 2019.
- [150] 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.
- [151] Rongye Shi, Zhaobin Mo, and Xuan Di. Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):540–547, May 2021.
- [152] Zeng Yang, Jin-Long Wu, and Heng Xiao. Enforcing deterministic constraints on generative adversarial networks for emulating physical systems. *arXiv preprint arXiv:1911.06671*, 2019.
- [153] Jin-Long Wu, Karthik Kashinath, Adrian Albert, Dragos Chirila, Heng Xiao, et al. Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. *Journal of Computational Physics*, 406:109209, 2020.
- [154] Peter Toth, Danilo J. Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. In *International Conference on Learning Representations*, 2020.
- [155] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International Conference on Machine Learning*, 2021.

- [156] Masanobu Horie, Naoki Morita, Toshiaki Hishinuma, Yu Ihara, and Naoto Mitsume. Isometric transformation invariant and equivariant graph convolutional networks. In *International Conference on Learning Representations*, 2021.
- [157] Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in Neural Information Processing Systems*, 32:14334–14345, 2019.
- [158] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.
- [159] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33, 2020.
- [160] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pages 3165–3176. PMLR, 2020.
- [161] Michael J Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. Lietransformer: Equivariant self-attention for lie groups. In *International Conference on Machine Learning*, pages 4533–4543. PMLR, 2021.
- [162] Pongpisit Thanasutives, Ken-ichi Fukui, and Masayuki Numao. Adversarial multi-task learning enhanced physics-informed neural networks for solving partial differential equations. *arXiv preprint arXiv:2104.14320*, 2021.
- [163] Sungyong Seo, Chuizheng Meng, Sirisha Rambhatla, and Yan Liu. Physics-aware spatiotemporal modules with auxiliary tasks for meta-learning. In *IJCAI*, 2021.
- [164] Chiyu Max Jiang, Jingwei Huang, Karthik Kashinath, Prabhat, Philip Marcus, and Matthias Niessner. Spherical CNNs on unstructured grids. In *International Conference on Learning Representations*, 2019.
- [165] Ferran Alet, Adarsh Keshav Jeewajee, Maria Bauza Villalonga, Alberto Rodriguez, Tomas Lozano-Perez, and Leslie Kaelbling. Graph element networks: adaptive, structured computation and memory. In *International Conference on Machine Learning*, pages 212–222. PMLR, 2019.
- [166] Nathaniel Trask, Ravi G Patel, Ben J Gross, and Paul J Atzberger. Gmls-nets: A framework for learning from unstructured data. *arXiv preprint arXiv:1909.05371*, 2019.
- [167] Yun Long, Xueyuan She, and Saibal Mukhopadhyay. Hybridnet: integrating model-based and data-driven learning to predict evolution of dynamical systems. In *Conference on Robot Learning*, pages 551–560. PMLR, 2018.
- [168] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. In *NIPS*, 2017.
- [169] Yunzhu Li, Jiajun Wu, Jun-Yan Zhu, Joshua B Tenenbaum, Antonio Torralba, and Russ Tedrake. Propagation networks for model-based control under partial observation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1205–1211. IEEE, 2019.
- [170] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.
- [171] Chi-Wang Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In *Advanced numerical approximation of nonlinear hyperbolic equations*, pages 325–432. Springer, 1998.
- [172] Jae Hyuk Lim, Seyoung Im, and Young-Sam Cho. Mls (moving least square)-based finite elements for three-dimensional nonmatching meshes and adaptive mesh refinement. *Computer methods in applied mechanics and engineering*, 196(17-20):2216–2228, 2007.
- [173] Richard P Feynman. The principle of least action in quantum mechanics. In *Feynman’s Thesis—A New Approach To Quantum Theory*, pages 1–69. World Scientific, 2005.
- [174] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, 2019.
- [175] Elliott Skomski, Ján Drgoňa, and Aaron Tuor. Automating discovery of physics-informed neural state space models via learning and evolution. In *Learning for Dynamics and Control*, pages 980–991. PMLR, 2021.
- [176] Ferran Alet, Erica Weng, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Neural relational inference with fast modular meta-learning. In *Advances in Neural Information Processing Systems*, pages 11804–11815, 2019.



- [177] Yutian Chen, Abram L Friesen, Feryal Behbahani, Arnaud Doucet, David Budden, Matthew Hoffman, and Nando de Freitas. Modular meta-learning with shrinkage. *Advances in Neural Information Processing Systems*, 33:2858–2869, 2020.
- [178] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. In *International Conference on Learning Representations*, 2021.
- [179] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [180] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [181] Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58, 2014.
- [182] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, 2016.
- [183] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.
- [184] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [185] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.
- [186] Ameya D Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136, 2020.
- [187] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020.
- [188] Jungeun Kim, Kookjin Lee, Dongeun Lee, Sheo Yon Jhin, and Noseong Park. Dpm: A novel training method for physics-informed neural networks in extrapolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8146–8154, 2021.