Machine-learning nonconservative dynamics for new-physics detection

Machine-learning nonconservative dynamics for new-physics detection

Ziming Liu,[1,2,3,*] Bohan Wang,[1] Qi Meng,[1] Wei Chen,[1,†] Max Tegmark,[2,3,‡] and Tie-Yan Liu[1,§]
[1]*Microsoft Research Asia, Beijing, China*
[2]*Department of Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*
[3]*AI Institute for Artificial Intelligence and Fundamental Interactions (IAIFI), Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

(Received 30 May 2021; accepted 22 October 2021; published 9 November 2021)

Energy conservation is a basic physics principle, the breakdown of which often implies *new physics*. This paper presents a method for data-driven "new physics" discovery. Specifically, given a trajectory governed by unknown forces, our *neural new-physics detector (NNPhD)* aims to detect new physics by decomposing the force field into conservative and nonconservative components, which are represented by a Lagrangian neural network (LNN) and an unconstrained neural network, respectively, trained to minimize the force recovery error plus a constant λ times the magnitude of the predicted nonconservative force. We show that a phase transition occurs at $\lambda = 1$, universally for arbitrary forces. We demonstrate that NNPhD successfully discovers new physics in toy numerical experiments, rediscovering friction (1493) from a damped double pendulum, Neptune from Uranus' orbit (1846), and gravitational waves (2017) from an inspiraling orbit. We also show how NNPhD coupled with an integrator outperforms both an LNN and an unconstrained neural network for predicting the future of a damped double pendulum.

DOI: 10.1103/PhysRevE.104.055302

I. INTRODUCTION

Energy conservation is a fundamental physical law, so when nonconservation is observed, physicists often consider it evidence of an unseen body or novel external forces rather than questioning the conservation law itself. In this paper, we will therefore refer to energy nonconservation as simply *new physics* and strive to auto-detect it.[1] Many experimental new physics discoveries have manifested as apparent violation of energy conservation, for example friction [2], Neptune [3], neutrinos [4], dark matter [5,6], extra-solar planets [7], and gravitational waves [8]. We focus on classical mechanics in this paper, but the idea extends to all fields of physics including quantum mechanics. We illustrate several classic examples in Fig. 1. In these cases, the new physics was historically identified from the residual force after fitting data to a conservative force of a *known functional form*. The key novel contribution in this paper is that our proposed model, dubbed the neural new physics detector (NNPhD), can discover the new physics even when the form of the conservative "old physics" is *not known*.

Data-driven discovery has proven extremely useful in physics, yet also nontrivial. For example, Kepler spent 25 years analyzing astronomical data before formulating his eponymous three laws. In this paper, we aim to automate and accelerate data-driven new physics discovery using machine learning tools. More concretely, given the trajectory of one or several objects governed by some force, we aim to decompose the force into conservative and nonconservative parts, followed by a symbolic interpreter (symbolic regression or nonlinear fitting) which extracts symbolic formulas. As a trivial example, we aim to decompose the force $f = -kq - \gamma\dot{q}$ of a damped harmonic oscillator into conservative part $f_c = -kq$ and a nonconservative part $f_n = -\gamma\dot{q}$.

Conservation laws have been introduced into neural networks as strong inductive biases, such as in Lagrangian neural network (LNN) [9], the Hamiltonian neural network (HNN) [10] and variants [11–14]. The limitation of these models lies in their inability to model nonconservative dynamics, where the nonconservation can be caused by dissipation, external driving forces, etc. Our proposed NNPhD can resolve this limitation by augmenting a LNN with an unconstrained neural network (UNN), illustrated in Fig. 2. Although prior works [9–35] attempt to learn the general or conservative force from data, most of these methods are unable to perform force decomposition, except for Refs. [32,35], which assume (partial) knowledge of physics and thus lose generality. Moreover, while the current literature mostly focuses on model *predictability*, we pay extra attention to *explainability* made possible by symbolic regression or nonlinear fitting to obtain the coefficients of a given symbolic form.

The rest of this paper is organized as follows: In Sec. II, we review the problem framing and useful results, define *force decomposition with minimal nonconservation* and propose NNPhD to learn this force decomposition. In Sec. III, we carry out numerical experiments to verify our theoretical analysis of the presented algorithm, as well as to demonstrate the potential of NNPhD for new-physics discovery.

*zmliu@mit.edu
†wche@microsoft.com
‡tegmark@mit.edu
§tyliu@microsoft.com

[1]In contrast, "new physics" in high energy physics specifically refers to "new fundamental particles" or "new fundamental interactions" beyond the standard model [1].

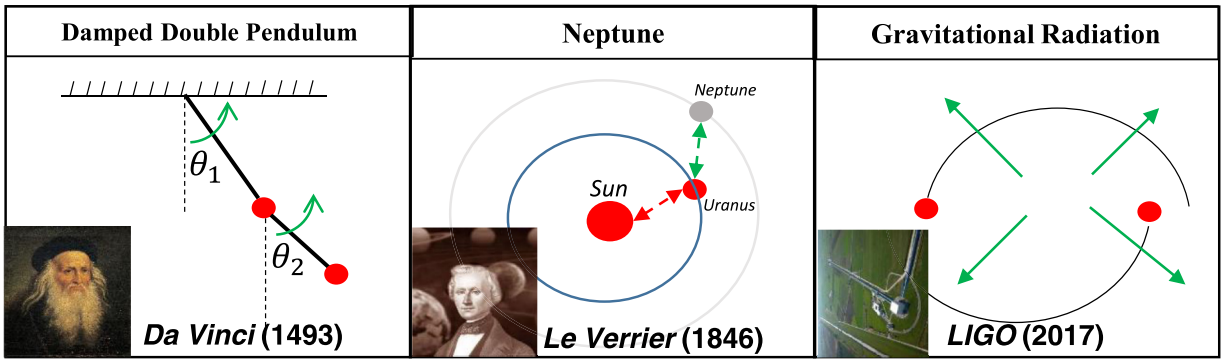

FIG. 1. NNPhD can auto-rediscover several classic examples.

II. METHOD

A. Notation

We consider the general classical physical system described by an n-dimensional vector $\mathbf{q}$ of generalized coordinates whose time-evolution $\mathbf{q}(t)$ is governed by a second-order ordinary differential equation,

$$\ddot{\mathbf{q}} = f(\mathbf{q}, \dot{\mathbf{q}}, t), \tag{1}$$

where $f : \mathbb{R}^{2n+1} \to \mathbb{R}^n$. The *acceleration* $\ddot{\mathbf{q}}$ is intimately related to *force* according to Newton's second law. In the following, we for simplicity refer to $f(\mathbf{q}, \dot{\mathbf{q}}, t)$ as a *force field* (dynamics perspective) or *acceleration field* (kinematics perspective) interchangeably. The dynamical systems in our numerical examples consist of k particles in d dimensions, so $n = kd$ and $\mathbf{q} \equiv [\mathbf{q}_1, \cdots, \mathbf{q}_k] \in \mathbb{R}^n$, but our NNPhD method is fully general and makes no such assumptions.

An important subset of dynamical systems are known as *conservative* because they conserve energy. Note that although physicists customarily limit the term 'conservative' to velocity-independent forces that can be written as the gradient of a potential function, we instead use the definition from the LNN formalism [9] and refer to a force as conservative as long as it can be described by an Euler-Lagrange equation. Our definition is deliberately less restrictive—for example, although the magnetic force cannot be written as a gradient, it conserves energy because it can be expressed as an Euler-Lagrange equation, and thus counts as "conservative" by our definition. Given a Lagrangian function $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$, the Euler-Lagrange equation is

$$\frac{d}{dt}\nabla_{\dot{\mathbf{q}}}\mathcal{L} = \nabla_{\mathbf{q}}\mathcal{L}, \tag{2}$$

where ∇ is the gradient operator. As reviewed in Appendix A and Ref. [9], the Lagrangian mechanics formalism implies that such systems allow Eq. (1) to be re-expressed as

$$\ddot{\mathbf{q}} = \left(\nabla_{\dot{\mathbf{q}}}\nabla_{\dot{\mathbf{q}}}^T\mathcal{L}\right)^{-1}\left[\nabla_{\mathbf{q}}\mathcal{L} - \left(\nabla_{\mathbf{q}}\nabla_{\dot{\mathbf{q}}}^T\mathcal{L}\right)\dot{\mathbf{q}}\right]. \tag{3}$$

For readers whose background is primarily in machine learning rather than physics, Appendix D provides a brief review of the Lagrangian mechanics formalism that we use in this paper.

B. Lagrangian neural networks

To guarantee energy conservation, inductive biases have recently been embedded into neural networks, including the Lagrangian neural network [9], Hamiltonian neural network [10], and variants [11–14]. As shown in Fig. 2, a LNN uses a neural network to parametrize the Lagrangian $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ and output $f_c^{\mathrm{NN}}(\mathbf{q}, \dot{\mathbf{q}})$ through evaluating Eq. (3). For a given loss function defined between model output $f_c^{\mathrm{NN}}(\mathbf{q}, \dot{\mathbf{q}})$ and ground truth $\ddot{\mathbf{q}}$, the LNN parameters can be learned using standard optimization algorithms. A trained LNN therefore contains a Lagrangian that determines conservative dynamics. Since not all physical systems conserve energy, the Lagrangian mechan-

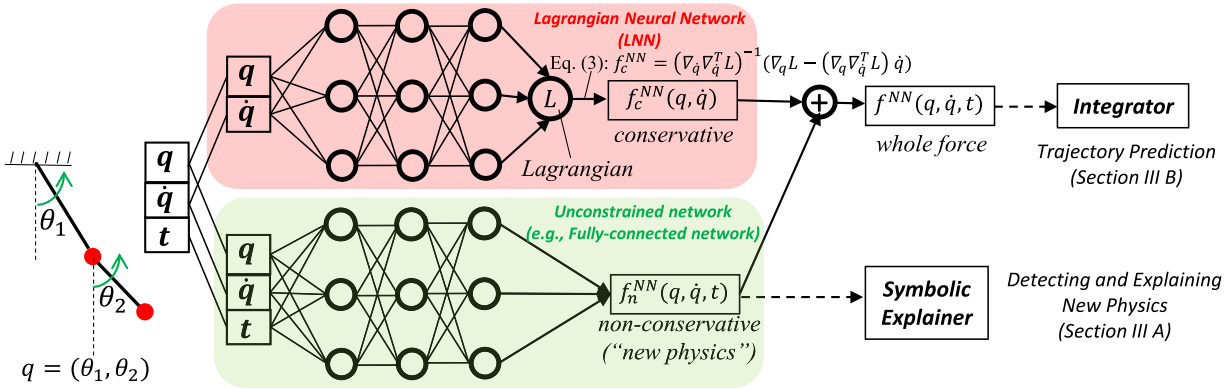

FIG. 2. NNPhD predicts dynamics by decomposing the force into conservative and nonconservative components, which can reveal new physics and improve trajectory extrapolation.

ics is insufficient for describing nonconservative dynamics, motivating the NNPhD framework.

C. The force decomposition minimizing nonconservation

Following the problem setting of LNN, we focus on the simple setting where the acceleration $\ddot{\mathbf{q}}$ is a known function, i.e., $\ddot{\mathbf{q}} \equiv f(\mathbf{q}, \dot{\mathbf{q}}, t)$. Our goal is therefore not to *learn* the force field, but to *decompose* the force field. In practice, where only discrete points on trajectory $\{(\mathbf{q}^{(i)}, t^{(i)})\}$ are known, $\dot{\mathbf{q}}^{(i)}$ and $\ddot{\mathbf{q}}^{(i)}$ can be extracted using a Neural ODE module [16].

The main goal is to decompose the force field $f(\mathbf{q}, \dot{\mathbf{q}}, t) : \mathbb{R}^{2n+1} \to \mathbb{R}^n$ into a (time-independent) conservative component $f_c(\mathbf{q}, \dot{\mathbf{q}}) : \mathbb{R}^{2n} \to \mathbb{R}^n$ and a nonconservative component $f_n(\mathbf{q}, \dot{\mathbf{q}}, t) : \mathbb{R}^{2n+1} \to \mathbb{R}^n$ such that

$$f(\mathbf{q}, \dot{\mathbf{q}}, t) = f_c(\mathbf{q}, \dot{\mathbf{q}}) + f_n(\mathbf{q}, \dot{\mathbf{q}}, t). \tag{4}$$

In general, the decomposition is not unique. We desire the decomposition that minimizes the nonconservative component $f_n(\mathbf{q}, \dot{\mathbf{q}}, t)$. To define the distance between two functions, we embed all functions $f(\mathbf{q}, \dot{\mathbf{q}}, t)$ in a normed vector space $(\mathcal{F}, \|\cdot\|)$ and define its conservative subspace $\mathcal{F}_c \subset \mathcal{F}$ as

$$\mathcal{F}_c = \Big\{ f \in \mathcal{F} \Big| \begin{smallmatrix} \exists \mathcal{L}(\mathbf{q},\dot{\mathbf{q}}):\mathbb{R}^{2n}\to\mathbb{R}, \quad s.t. \\ f(\mathbf{q},\dot{\mathbf{q}})=(\nabla_{\dot{\mathbf{q}}}\nabla_{\dot{\mathbf{q}}}^T\mathcal{L})^{-1}\left(\nabla_{\mathbf{q}}\mathcal{L}-(\nabla_{\mathbf{q}}\nabla_{\dot{\mathbf{q}}}^T\mathcal{L})\dot{\mathbf{q}}\right) \end{smallmatrix} \Big\}.$$

We formally define the force decomposition as follows:

Definition II.1. (force decomposition with minimal nonconservation). The *conservative component* of $f(\mathbf{q}, \dot{\mathbf{q}}, t)$ is defined as

$$f_c(\mathbf{q}, \dot{\mathbf{q}}) \equiv \arg\min_{g \in \mathcal{F}_c} \|f(\mathbf{q}, \dot{\mathbf{q}}, t) - g(\mathbf{q}, \dot{\mathbf{q}})\|. \tag{5}$$

We denote $f_n(\mathbf{q}, \dot{\mathbf{q}}, t) \equiv f(\mathbf{q}, \dot{\mathbf{q}}, t) - f_c(\mathbf{q}, \dot{\mathbf{q}})$ the *nonconservative component* of f and denote the decomposition $f(\mathbf{q}, \dot{\mathbf{q}}, t) = f_c(\mathbf{q}, \dot{\mathbf{q}}) + f_n(\mathbf{q}, \dot{\mathbf{q}}, t)$ the *force decomposition minimizing nonconservation*.

D. Neural new-physics detector (NNPhD) framework

To learn the force decomposition minimizing nonconservation, we define a learning framework dubbed the NNPhD. Specifically, NNPhD learns f_c and f_n jointly. As illustrated in Fig. 2, NNPhD consists of two parallel modules, a LNN and an UNN. Our method can be used with any feedforward architecture for the UNN, whose purpose of the UNN is to be sufficiently expressive to learn the nonconserving part of the force. In our numerical examples, we implement the UNN as a simple perceptron with two hidden layers, with good results; for smooth functions, perceptrons are known to be universal approximators in theory when arbitrarily wide [36], and they also perform well in practice when deep enough [37,38].

The LNN takes in $(\mathbf{q}, \dot{\mathbf{q}})$ to predict a Lagrangian $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}; \boldsymbol{w}_c)$ in the intermediate layer and outputs $f_c^{\mathrm{NN}}(\mathbf{q}, \dot{\mathbf{q}}; \boldsymbol{w}_c)$ calculated from Eq. (3), where $\boldsymbol{w}_c$ are LNN parameters. The UNN could be a fully connected network that takes in $(\mathbf{q}, \dot{\mathbf{q}}, t)$ and outputs $f_n^{\mathrm{NN}}(\mathbf{q}, \dot{\mathbf{q}}, t; \boldsymbol{w}_n)$ where $\boldsymbol{w}_n$ are UNN parameters. The two outputs are summed to predict the full force field:

$$f^{\mathrm{NN}}(\mathbf{q}, \dot{\mathbf{q}}, t; \boldsymbol{w}_c, \boldsymbol{w}_n) = f_c^{\mathrm{NN}}(\mathbf{q}, \dot{\mathbf{q}}; \boldsymbol{w}_c) + f_n^{\mathrm{NN}}(\mathbf{q}, \dot{\mathbf{q}}, t; \boldsymbol{w}_n). \tag{6}$$

We take both *recovery error* and *minimal nonconservation* into consideration to design our loss function: (1) f^{NN} should recover ground truth f; (2) we make maximal use of f_c^{NN} and reduce f_n^{NN} as much as possible (e.g., when f is conservative, we hope that f_n^{NN} vanishes). Guided by these two principles, we define our loss function as follows [denoting the ith sample $\mathbf{x}^{(i)} \equiv (\mathbf{q}^{(i)}, \dot{\mathbf{q}}^{(i)})$]:

$$\begin{aligned} L_{\mathrm{NNPhD}}(\boldsymbol{w}_c, \boldsymbol{w}_n) &= L_e(\boldsymbol{w}_c, \boldsymbol{w}_n) + \lambda L_b(\boldsymbol{w}_n), \\ L_b(\boldsymbol{w}_n) &\equiv \left(\frac{1}{Nn} \sum_{i=1}^{N} \left\| f_n^{\mathrm{NN}}(\mathbf{x}^{(i)}, t^{(i)}; \boldsymbol{w}_n) \right\|^p \right)^{\frac{1}{p}}, \\ L_e(\boldsymbol{w}_c, \boldsymbol{w}_n) &\equiv \left(\frac{1}{Nn} \sum_{i=1}^{N} \left\| f_c^{\mathrm{NN}}(\mathbf{x}^{(i)}; \boldsymbol{w}_c) + f_n^{\mathrm{NN}}(\mathbf{x}^{(i)}, t^{(i)}; \boldsymbol{w}_n) - f(\mathbf{x}^{(i)}, t^{(i)}) \right\|^p \right)^{\frac{1}{p}}, \end{aligned} \tag{7}$$

where $p \geqslant 1$ and the regularization coefficient $\lambda > 0$. The factors $\frac{1}{N}$ and $\frac{1}{n}$ average over samples and degrees of freedom, respectively. Here we use L_p function norms, i.e., $\|f\| \equiv (\int |f|^p d\mu)^{1/p}$, where the integral is replaced by averaging over finite training samples. L_e is the recovery error and L_b penalizes the unconstrained network to discourage it from learning conservative dynamics.

E. The regularization phase transition

Does minimizing Eq. (7) yield the force decomposition of Eq. (5)? We offer an affirmative answer to this question by presenting Theorem 1 informally here. Appendix F provides a rigorous formulation and proof of this theorem.

Theorem 1. (Informal) Suppose f_c^{NN} and f_n^{NN} can represent any conservative force field and any (continuous) force field, and (f_c^*, f_n^*) denotes the pair that minimizes NNPhD loss from Eq. (7). Then we have a phase transition at $\lambda = 1$ such that $(f_c^*, f_n^*) = (f_c, f_n)$ when $0 < \lambda < 1$, and $(f_c^*, f_n^*) = (f_c, 0)$ when $\lambda > 1$.

Theorem 1 has two interesting and useful implications: (1) *sharp phase transition*: The recovery error $L_e = 0$ when $\lambda < 1$ and $L_e = \|f_n\| > 0$ when $\lambda > 1$. As a result, nonconservative dynamics predicts an error jump of L_e at $\lambda = 1$, while conservative dynamics does not. This phenomenon justifies the term "detector" in our model name, in the sense that nonconservative dynamics is *detected* by the sharp phase transition at $\lambda = 1$. (2) *effortless* λ *tuning*: Any $\lambda \in (0, 1)$ would achieve the force decomposition. Below we report numerical experiments

TABLE I. We test if NNPhD can automatically decompose these three force fields into a conservative part (first term) and a nonconservative part (second term) corresponding to the "new physics."

Model	Equation
Damped double pendulum	$\begin{pmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{pmatrix} = \begin{pmatrix}\frac{m_2 l_1 \dot{\theta}_1^2 \sin(\theta_2-\theta_1)\cos(\theta_2-\theta_1)+m_2 g\sin\theta_2+m_2 l_2\dot{\theta}_2^2\sin(\theta_2-\theta_1)-(m_1+m_2)g\sin\theta_1}{(m_1+m_2)l_1-m_2 l_1\cos^2(\theta_2-\theta_1)}\\ \frac{-m_2 l_2\dot{\theta}_2^2\sin(\theta_2-\theta_1)+(m_1+m_2)(g\sin\theta_1\cos(\theta_2-\theta_1)-l_1\dot{\theta}_1^2\sin(\theta_2-\theta_1)-g\sin\theta_2)}{(m_1+m_2)l_1-m_2 l_1\cos^2(\theta_2-\theta_1)}\end{pmatrix} - \gamma\begin{pmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{pmatrix}$
Neptune	$\begin{pmatrix}\ddot{x}\\ \ddot{y}\end{pmatrix} = \begin{pmatrix}-\frac{GM_\odot x}{(x^2+y^2)^{\frac{3}{2}}}\\ -\frac{GM_\odot y}{(x^2+y^2)^{\frac{3}{2}}}\end{pmatrix} + \begin{pmatrix}\frac{GM_n(-x+r_n\cos(\omega_n t))}{[(x-r_n\cos(\omega_n t))^2+(y-r_n\sin(\omega_n t))^2]^{\frac{3}{2}}}\\ \frac{GM_n(-y+r_n\sin(\omega_n t))}{[(x-r_n\cos(\omega_n t))^2+(y-r_n\sin(\omega_n t))^2]^{\frac{3}{2}}}\end{pmatrix}$
Gravitational radiation	$\begin{pmatrix}\ddot{x}\\ \ddot{y}\end{pmatrix} = \begin{pmatrix}-\frac{G(M_1+M_2)x}{(x^2+y^2)^{\frac{3}{2}}}\\ -\frac{G(M_1+M_2)y}{(x^2+y^2)^{\frac{3}{2}}}\end{pmatrix} + \frac{32M_1M_2(M_1^2+M_2^2)}{5Gc^5(M_1+M_2)^5}\begin{pmatrix}-(\dot{x}_i^2+\dot{y}_i^2)^4\dot{x}_i\\ -(\dot{x}_i^2+\dot{y}_i^2)^4\dot{y}_i\end{pmatrix}$

showing that in practice, too small λ do not regularize the unconstrained network effectively, and force decomposition results are robust for $0.05 \lesssim \lambda < 1$ independent of dynamical systems at study.

As we will see in Appendix F, the proof is more complicated than one might naively expect. *If conservative force fields were to form a linear subspace* where the norm $\|\cdot\|$ were the L_2-norm induced by an inner product, then the optimal conservative component f_c from Eq. (5) that minimizes L_{NNPHD} would simply be the orthogonal projection onto that subspace, and the nonconservative residual f_n would be orthogonal to that subspace—which would greatly simplify the computation of f_c in practice. Unfortunately, conservative force fields as we have defined them generally do *not* form a linear subspace, i.e., the sum of two energy-conserving force fields may not conserve energy, which is related to the nonlinear nature of Eq. (3).

III. RESULTS FROM NUMERICAL EXPERIMENTS

In this section, we test our NNPhD algorithm with a series of numerical examples defined in Table I. In Sec. III A, we quantify its ability to rediscover symbolic expressions for "new physics," such as friction, Neptune, and gravitational waves. In Sec. III B, we show that, although NNPhD is designed for new physics detection, it can also outperform baseline trajectory prediction for the damped double pendulum example. In Sec. III C, we use toy examples to verify and quantify the aforementioned λ-dependent phase transition, and explore how the choices of p and λ in Eq. (7) influence algorithm behavior. Finally we discuss how data quality affects identifiability of new physics in Sec. III D. Further technical details on model parameters, simulations and neural network architecture are provided in Appendix A.

A. Discovery of new physics

We now test NNPhD on three numerical examples defined in Table I, to see if it can rediscover friction (1493), Neptune (1846), and gravitational wave emission (2017). In all three cases, the force fields defined by the right-hand side are the sum of a conservative part (the first term) and a nonconservative "new physics" part (the second term) that we hope to discover. Before delving into our numerical experiments, let us briefly comment on how we model these three dynamical systems.

1. Physical systems tested

Friction. Italian polymath Leonardo da Vinci first recorded the basic laws of friction in 1493 [2]. We add friction to the double pendulum system and to test if NNPhD can automatically discover the friction force solely from data. The damped double pendulum example can be described by two angles and their derivatives, i.e., $\mathbf{q} = (\theta_1, \theta_2)$ and $\dot{\mathbf{q}} = (\dot{\theta}_1, \dot{\theta}_2)$. In our numerical experiment, we choose the physical parameters $m_1 = m_2 = g = l_1 = l_2 = 1$, $\gamma = 0.02$.

Neptune. *Le Verrier* postulated the existence of Neptune in 1846 [39]: astronomers had found that Uranus' orbit around the Sun precessed in a way suggesting the presence of a force of unknown cause, later identified as Neptune. Neptune was invisible at the time in the sense that contemporary astronomers could not observe its position or velocity, but Le Verrier (and NNPhD) were able to identify the existence of a third body by identifying a nonconservative contribution to the force field of the two-body system. For our numerical experiments, we make the simplifying assumptions that (1) the Sun remains fixed at the origin, (2) the elliptical orbits of Uranus and Neptune are circular (have eccentricity $e = 0$) and lie in the same plane, (3) Neptune's orbit is unaffected by Uranus, and (4) the effects of other planets are negligible. Here x and y denote the coordinates of Uranus, and time t is measured in units such that Uranus' orbital period is $2\pi\sqrt{2^3}$. We choose $G = 1$, mass of Sun $M_\odot = 1$. Neptune's mass, orbital radius, and angular velocity are $M_n = 0.005$, $r_n = 3$, and $\omega_n = 3^{-\frac{3}{2}} \approx 0.192$. It will be interesting to investigate in future work how to lift the four simplifying assumptions and simulate the whole solar system, possibly by combining NNPhD with graph neural networks using appropriate inductive biases [25].

Gravitational Radiation. As predicted by Einstein, the gravitational two-body problem is nonconservative, since the system radiates gravitational radiation that carries away energy and causes orbital decay. Experimental confirmation of this garnered Nobel Prizes both in 1993 (for the Hulse-Taylor pulsar) [40] and in 2017 (for the LIGO discovery of gravitational waveforms from black hole mergers) [41], and there is great current interest in exploiting such signals both for

gravitational wave astronomy and for precision tests of general relativity. To test whether NNPhD can auto-discover the nonconservative force caused by gravitational wave back-reaction solely from black hole trajectories, we simulate a binary black hole inspiral using the approximation from Ref. [42] that the radiated gravitational wave power $P = \frac{32}{5}\frac{G}{c^5}\mu^2 r^4 \Omega^6 = \frac{32}{5}\frac{G}{c^5}\mu^2 \frac{v^6}{r^2}$ in a slowly decaying circular orbit (of radius r, angular frequency Ω and reduced mass $\mu \equiv (M_1^{-1} + M_2^{-1})^{-1}$) equals the energy loss rate $-dE/dt = vf$ from a dissipative back-reaction force f. Using $\Omega \propto r^{-3/2}$ and $v \propto r^{-1/2}$ from Kepler's 3rd law gives $P \propto v^{10}$, with a total force

$$\boldsymbol{f} = \mu\ddot{\boldsymbol{r}} = -\frac{GM_1M_2}{r^3}\boldsymbol{r} - \frac{32M_1^2M_2^2(M_1^2+M_2^2)}{5Gc^5(M_1+M_2)^6}v^8\boldsymbol{v}, \quad (8)$$

corresponding to an acceleration

$$\ddot{\boldsymbol{r}} = -\frac{G(M_1+M_2)}{r^3}\boldsymbol{r} - \frac{32M_1M_2(M_1^2+M_2^2)}{5Gc^5(M_1+M_2)^5}v^8\boldsymbol{v}, \quad (9)$$

where $\boldsymbol{r} = \boldsymbol{r}_2 - \boldsymbol{r}_1$ and $\boldsymbol{v} = \boldsymbol{v}_2 - \boldsymbol{v}_1$. We choose these physical parameters to be $G = M_1 = M_2 = 1, c = 3$.

2. Detection of new physics

These three physical systems have $d = 2$ degrees of freedom, obeying the second-order coupled differential equations in Table I. Including the corresponding conjugate momenta, a system's state is thus a point moving along some trajectory in a $2d$-dimensional phase space, satisfying a $2d$ first-order coupled differential equations. We solve these equations and compute the trajectories numerically using a fourth-order Runge-Kutta integrator at $N_{\text{step}} = \{300, 1000, 300\}$ timesteps of size $\varepsilon = \{0.1, 0.1, 0.05\}$ for the three physical systems, using the following initial conditions:

$$\begin{aligned}(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) &= (1, 0, 0, 0),\\ (x, y, \dot{x}, \dot{y}) &= \left(3, 0, 0, \frac{1}{\sqrt{3}}\right),\\ (x, y, \dot{x}, \dot{y}) &= (0, 2, -1, 0).\end{aligned} \quad (10)$$

Once trajectory points are calculated, the ground truth forces f at those points are evaluated using the formula in Table I.[2] We do not hold back any testing data in this section, since many insights can be gained solely from training data. We will hold back testing data and verify NNPhD's generalization ability in Sec. III B.

We then train NNPhD on the aforementioned trajectory data as detailed in Appendix B. Figure 3 shows the resulting NNPhD prediction loss L_e as a function of λ, revealing a striking phase transition at $\lambda = 1$: for $\lambda < 1$, L_e is almost zero, while for $\lambda > 1$, L_e is an approximately constant positive number, indicating the magnitude of nonconservative components.

As we showed above, such a phase transition is a smoking-gun signature of new physics manifesting as nonconservative dynamics. The observed phase transitions thus justify the NNPhD name.

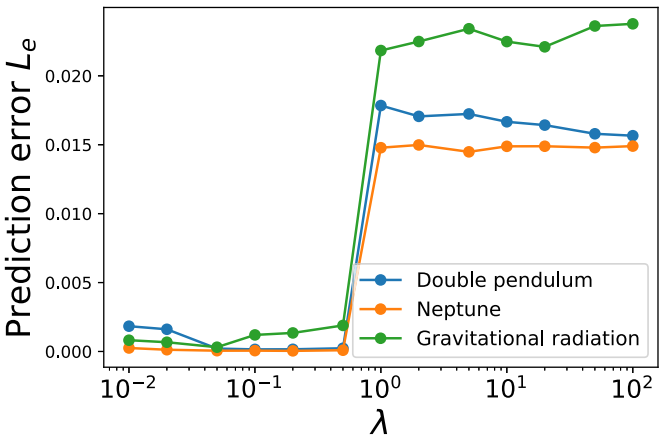

FIG. 3. In all our three examples, clear phase transitions at $\lambda = 1$ indicate the existence of new physics.

3. Modeling of new physics with symbolic expressions

After detecting the existence of new physics, physicists are interested in understanding and explaining this new physics by describing it with via symbolic expressions. We found that if we did not impose any inductive biases on the LNN, we unfortunately did not auto-discover any meaningful symbolic expressions. We therefore drew inspiration from the history of physics, where inductive biases have routinely been used. For example, physicists often knew and used analytic formulas for the old physics when quantifying new physics. In this spirit, we constrain the form of LNN Lagrangian so that only a set of coefficients are learnable, while the unconstrained network remains to a fully general feedforward neural network with two hidden layers containing 200 neurons each. Specifically, we parametrize the Lagrangians for our three examples as follows:

$$\begin{aligned}\mathcal{L}_{\text{fric}} &= c_1\cos\theta_1 + c_2\cos\theta_2 + c_3\dot{\theta}_1^2 + c_4\dot{\theta}_2^2\\ &\quad + c_5\dot{\theta}_1\dot{\theta}_2\cos(\theta_1 - \theta_2),\\ \mathcal{L}_{\text{neptune}} &= c_1\dot{x}^2 + c_2\dot{y}^2 + \frac{c_3}{\sqrt{x^2+y^2}},\\ \mathcal{L}_{\text{grav}} &= c_1\dot{x}^2 + c_2\dot{y}^2 + \frac{c_3}{\sqrt{x^2+y^2}}.\end{aligned} \quad (11)$$

This is implemented by inputting hand-crafted features ($\cos\theta, x^2$, etc.) into a learnable linear layer which outputs the predicted Lagrangian. We adopt a train-and-explain strategy:

(1) *Training.* Like before, we train the whole NNPhD (LNN and the unconstrained network are updated simultaneously) with $\lambda = 0.2$ using the ADAM optimizer with annealing learning rate $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ for 2000 steps.

(2) *Explaining.* After training, we aim to extract more interpretable physics from the unconstrained network via constrained nonlinear optimization of free parameters (displayed as **bold** in Table II) to explain the output of the unconstrained network, since ground truth symbolic forms are available.

[2]In more realistic settings, one would first extract $\ddot{\mathbf{q}}$ from trajectory data, e.g., with Neural ODE [16] or AI Physicist [27], and then use $\ddot{\mathbf{q}}$ as labels to train NNPhD. We treat $\ddot{\mathbf{q}} = f(\mathbf{q}, \dot{\mathbf{q}}, t)$ as an oracle in this paper since we focus on the force field decomposition aspect.

TABLE II. Symbolic formulas discovered by NNPhD.

Physics example	Target	Ground truth "new physics"	NNPhD+Symbolic
Double pendulum	$\begin{pmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{pmatrix}$	$\begin{pmatrix}-\mathbf{0.02}\dot{\theta}_1-\mathbf{0.00}\dot{\theta}_2\\ -\mathbf{0.00}\dot{\theta}_1-\mathbf{0.02}\dot{\theta}_2\end{pmatrix}$	$\begin{pmatrix}-\mathbf{0.018}\dot{\theta}_1-\mathbf{0.001}\dot{\theta}_2\\ -\mathbf{0.001}\dot{\theta}_1-\mathbf{0.018}\dot{\theta}_2\end{pmatrix}$
Neptune	$\begin{pmatrix}\ddot{x}\\ \ddot{y}\end{pmatrix}$	$\begin{pmatrix}\frac{\mathbf{0.005}(-x+\mathbf{3}\cos(\mathbf{0.192}t))}{[(x-\mathbf{3}\cos(\mathbf{0.192}t))^2+(y-\mathbf{3}\sin(\mathbf{0.192}t))^2]^{\frac{3}{2}}}\\ \frac{\mathbf{0.005}(-y+\mathbf{3}\sin(\mathbf{0.192}t))}{[(y-\mathbf{3}\sin(\mathbf{0.192}t))^2+(y-\mathbf{3}\sin(\mathbf{0.192}t))^2]^{\frac{3}{2}}}\end{pmatrix}$	$\begin{pmatrix}\frac{\mathbf{0.0052}(-x+\mathbf{3.004}\cos(\mathbf{0.192}t))}{[(x-\mathbf{3.004}\cos(\mathbf{0.192}t))^2+(y-\mathbf{3.004}\sin(\mathbf{0.192}t))^2]^{\frac{3}{2}}}\\ \frac{\mathbf{0.0052}(-y+\mathbf{3.004}\sin(\mathbf{0.192}t))}{[(y-\mathbf{3.004}\sin(\mathbf{0.192}t))^2+(y-\mathbf{3.004}\sin(\mathbf{0.192}t))^2]^{\frac{3}{2}}}\end{pmatrix}$
Gravitational radiation	$\begin{pmatrix}\ddot{x}_1\\ \ddot{y}_1\end{pmatrix}$	$\begin{pmatrix}-\mathbf{0.00165}(\dot{x}_1^2+\dot{y}_1^2)^{\mathbf{4}}\dot{x}_1\\ -\mathbf{0.00165}(\dot{x}_1^2+\dot{y}_1^2)^{\mathbf{4}}\dot{y}_1\end{pmatrix}$	$\begin{pmatrix}-\mathbf{0.00170}(\dot{x}_1^2+\dot{y}_1^2)^{\mathbf{3.94}}\dot{x}_1\\ -\mathbf{0.00170}(\dot{x}_1^2+\dot{y}_1^2)^{\mathbf{3.94}}\dot{y}_1\end{pmatrix}$

In Table II, we show ground truth "new physics" and NNPhD fitted symbolic expressions. Fitted coefficients are seen to match ground truth quite well: (1) damping coefficient; (2) orbital radius and angular velocity of Neptune around the Sun; (3) magnitude and velocity dependence of gravitational wave emission.

B. Prediction of trajectories

In addition to discovering new physics, as we saw above, NNPhD can also compete with other methods on simple trajectory prediction, and we will now test its performance for out-of-distribution generalization. Specifically, we test how accurately it can extrapolate the trajectory of the damped double pendulum from Sec. III A 2, whose state is specified by two angles (θ_1, θ_2) and corresponding angular velocities $(\dot{\theta}_1, \dot{\theta}_2)$. We compute a trajectory with a fourth-order Runge-Kutta integrator at $N_{\text{step}} = 2000$ timesteps of size $\varepsilon = 0.1$ using the initial conditions $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = (1, 0, 0, 0)$. Our test task is to extrapolate beyond $t = 30$, so we split the trajectory into a training dataset $(0 \leqslant t \leqslant 30)$ and a test dataset $(30 \leqslant t \leqslant 200)$.

We train NNPhD with $\lambda = 0.2$ and feed its prediction f into a fourth-order Runge-Kutta integrator to produce the predicted trajectory. Figure 4 compares the performance with that from a LNN and a UNN network. The left panel shows that both NNPhD and the UNN network can fit θ_1 well on training samples and extrapolate for a short period, but fail at larger times due to accumulated errors and sensitive phases. In contrast, we see that the LNN cannot even fit the training data, because it has the invalid energy-conservation assumption built in. The right panel shows that ground-truth energy is decaying exponential over time due to friction, while the LNN stubbornly predicts constant energy. NNPhD is seen to predict the energy decay best of the three methods, while the UNN network slightly overpredicts the energy for a while and then incorrectly transitions to predicting approximate energy conservation. In this paper we compare NNPhD with only its single components, but we admit there are plenty of off-the-shelf methods for trajectory prediction, including reservior computing [43], long short-term memory [44], neural ordinary differential equations [45], etc. Comparing the performance of NNPhD with that of other methods will be useful to understand the benefits and limitations of NNPhD, which will be a promising future direction.

C. Theory verification and algorithm benchmarking

In this section, to better understand its algorithmic behavior, we test NNPhD on the six simple dynamical systems in

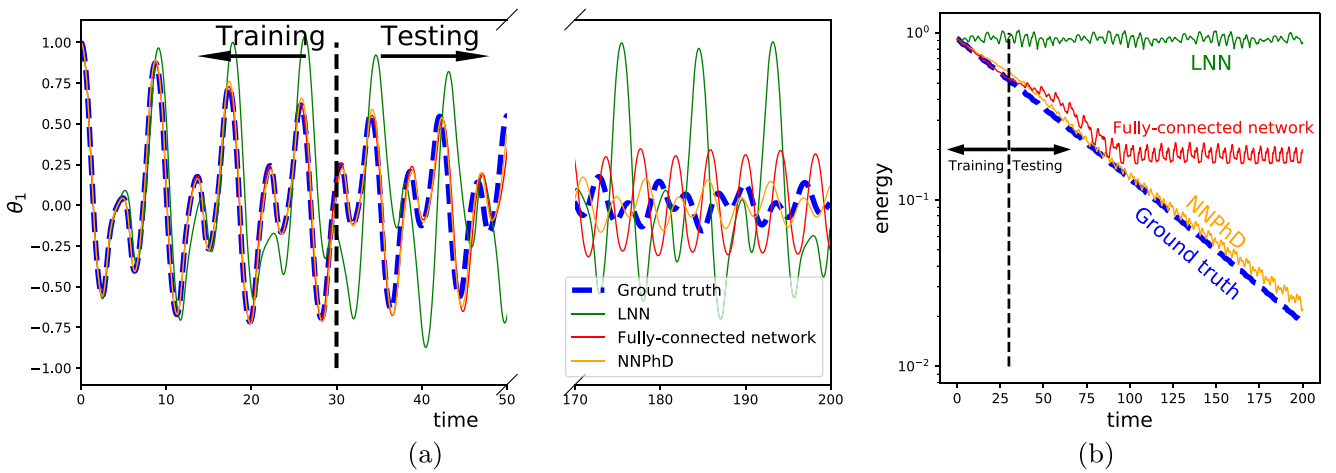

FIG. 4. Double pendulum example: (a) Both NNPhD and UNN can accurately fit the angle θ_1 of training samples and can successfully extrapolate for a brief period, while LNN fails to model the nonconservative dynamics; (b) NNPhD correctly predicts the exponential energy decay on testing samples, while the UNN generalizes worse, and LNN incorrectly conserves energy.

TABLE III. Examples of conservative and nonconservative dynamics.

Classes	Model	Equation	Lagrangian
Conservative ($f_c^{\rm PHY}$)	Harmonic oscillator (HO)	$\ddot{q} = -q$	$\mathcal{L} = \dot{q}^2/2 - q^2/2$
	Magnetic field (MF)	$\ddot{q}_1 = \dot{q}_2$ $\ddot{q}_2 = -\dot{q}_1$	$\mathcal{L} = (\dot{q}_1 - q_2)^2/2$ $+(\dot{q}_2 + q_1)^2/2$
	Constant gravity (CG)	$\ddot{q} = -1$	$\mathcal{L} = \dot{q}^2/2 - q$
Nonconservative ($f_n^{\rm PHY}$)	Linear damping (LD)	$\ddot{q} = -\dot{q}$	
	Constant damping (CD)	$\ddot{q} = -\mathrm{sgn}(\dot{q})$	NA
	Periodic force (PF)	$\ddot{q} = \sin(t)$	

physics in Table III: conservative examples involve a harmonic oscillator (HO), a magnetic field (MF), and constant gravity (CG), and nonconservative examples include linear damping (LD), constant damping (CD), and a periodic force (PF). We combine these into five examples to obtain two conservative systems (HO+MF, HO+CG) and three nonconservative systems (HO+LD, HO+CD, HO+PF), whose dynamical equations are summarized in Appendix A. For each system, we train NNPhD with the ADAM optimizer for 2000 iterations, using batch size 32, learning rate schedule $\{0.01, 0.001, 0.0001, 0.00001\}$, and 500 iterations for each learning rate.

We now explore how the performance of the NNPhD depends on the regularization coefficient λ and norm index p by testing $\lambda = \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100\}$ and $p = 1, 2, 3$. Instead of simulating trajectories to generate data as in previous sections, we compute $\dot{\mathbf{q}} = f(\mathbf{q}, \dot{\mathbf{q}}, t)$ at N random points $(\mathbf{q}^{(i)}, \dot{\mathbf{q}}^{(i)}, t_i)$. We first generate all positions, velocities, and times as independent Gaussian random variables with zero mean and unit standard deviation, then we explore more complicated coverage in Sec. III D. We generate 10^3 training samples and 10^3 testing samples $(\mathbf{q}, \dot{\mathbf{q}}, t)$.

How performance depends on p. In Fig. 5(a), we plot the dependence of the prediction error L_e on λ ($p = 1$), again verifying the phase transition prediction from Sec. II E: The nonconservative systems (HO+LD, HO+CD, HO+PF) are seen to have a large error jump at $\lambda = 1$ while, in contrast, L_e does not increase at $\lambda = 1$ for the conservative systems (HO+MF, HO+CG). In fact, HO+MF has even lower

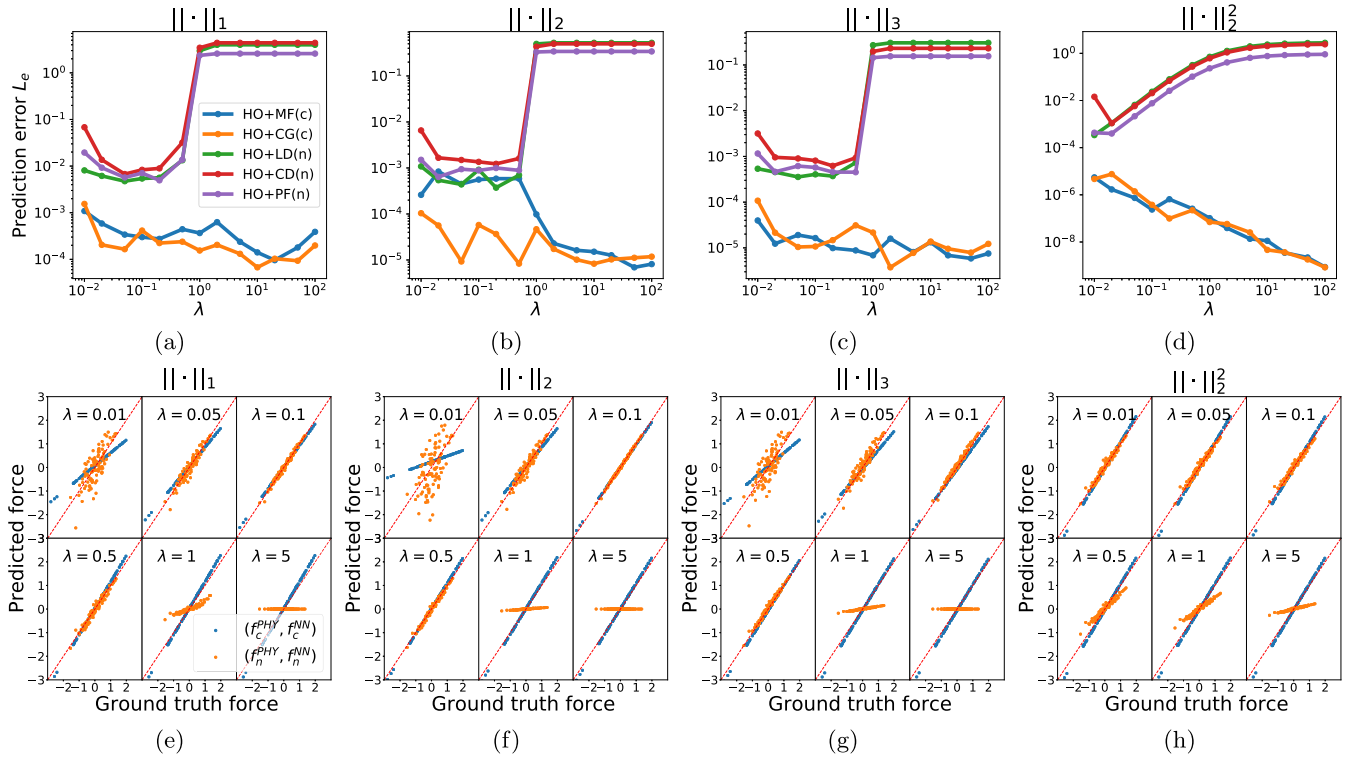

FIG. 5. NNPhD is seen to behave robustly for $0.05 \lesssim \lambda < 1$ and $p \geqslant 1$. We test NNPhD on five examples (the first two are conservative, and last three are nonconservative). (a)–(d) prediction error L_e as a function of λ with different norms as loss function: for (a)–(c) $\|\cdot\|_p (p = 1, 2, 3)$, nonconservative dynamics has an error jump at $\lambda = 1$, while conservative dynamics does not. In panel (d), mean squared loss leads to a smooth phase transition for nonconservative dynamics; (e)–(h) for the linear damping case $\ddot{q} = -q - \frac{1}{2}\dot{q}$, we show how $f_c^{\rm NN}$ and $f_n^{\rm NN}$ are aligned with $f_c^{\rm PHY}$ and $f_n^{\rm PHY}$ for different loss functions and different λ.

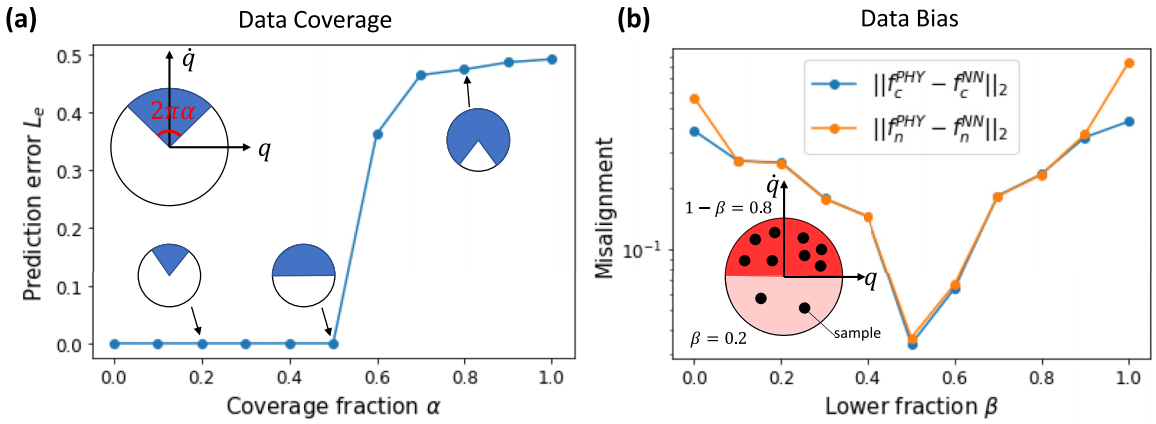

FIG. 6. Dependence on data distribution parameters α and β. Low-quality data might prevent new physics discovery via (a) incomplete data coverage and (b) biased data distribution.

prediction error at larger λ, showing the advantage of employing a Lagrangian neural network as opposed to an unconstrained neural network for conservative systems. Figure 5(a) shows that NNPhD has the ability to distinguish between conservative and nonconservative dynamics by looking at prediction loss around $\lambda = 1$, i.e., a sharp phase transition indicates nonconservative dynamics. The above observations also apply to Figs. 5(b) and 5(c) when $p = 2$ and $p = 3$. However, Fig. 5(d) shows that mean-squared-error loss (where the L_2-norm is squared) leads to a smooth transition.

How performance depends on λ. We then quantify how accurately the conservative and nonconservative components are modeled for different λ-values. Figure 5(e) shows our results for the damped oscillator example $\ddot{q} = -q - \frac{1}{2}\dot{q}$, comparing f_c^{NN} with $f_c^{PHY} = -q$ and f_n^{NN} with $f_n^{PHY} = -\frac{1}{2}\dot{q}$. As Theorem 1 suggests, we observe that (1) when $\lambda > 1$, f_n^{NN} predicts 0 while $f_c^{NN} \approx f_c^{PHY}$; (2) when $0.05 \lesssim \lambda < 1$, $f_c^{NN} \approx f_c^{PHY}$ and $f_n^{NN} \approx f_n^{PHY}$; (3) when $\lambda \lesssim 0.05$, although in theory it behaves similarly to (2), a small λ does not have much incentive to penalize the UNN network, which therefore absorbs part of the conservative component. Figures 5(f)–5(h) show that the alignments between the ground truth components and the predictions from NNPhD are quite robust for different choices of loss function.

D. Physics discovery requires high-quality data

Although NNPhD does not assume any data distribution to achieve the decomposition, we will now see that NNPhD can only learn to accurately decompose the force into conservative and nonconservative parts if the data has high quality, specifically, if the data distribution has (1) adequate coverage of the state space $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}})$ and (2) is unbiased.

Incomplete data coverage. We now explore the situation where data points are only accessible in a pie-shaped subset of space covering an angular fraction of $\alpha \in [0, 1]$, as illustrated in Fig. 6(a). We consider the 1D constant damped oscillator $\ddot{q} = -q - \frac{1}{2}\mathrm{sgn}(\dot{q})$, train NNPhD with $\lambda = 10$ on datasets with different fractions α and calculate the prediction loss L_e. Recall that when $\lambda = 10$, a high prediction error L_e is a sign of nonconservation. Figure 6(a) shows that when $\alpha \leqslant 0.5$, no samples are generated in the lower half plane (where $\dot{q} < 0$), then the prediction error L_e is nearly zero, revealing no sign of nonconservation. For $\alpha > 0.5$, however, NNPhD has a large L_e, revealing the nonconservative nature of the damping force. This observation makes physical sense since, if only $\dot{q} > 0$ samples are observed, the damping force acts as a constant conservative force (like gravity) which can be included as a $(-\frac{1}{2}q)$ term in a Lagrangian $\mathcal{L} = \frac{1}{2}\dot{q}^2 - \frac{1}{2}q^2 - \frac{1}{2}q$, making the dynamics appear energy conserving.

Imbalanced data distribution. Even in the case when data is available everywhere in all relevant parts of phase space, the data set can still be imbalanced, e.g., contain more $\dot{q} > 0$ samples than $\dot{q} < 0$ ones. Figure 6(a) show that this is not a sever problem in the sense that it does not preclude us from identifying the *existence* of nonconservative dynamics, since the presence of since merely a few samples with $\dot{q} < 0$ suffices to give a clear signal of nonconservation. However, such imbalance may harm the accuracy of our decomposition. We consider the linear damped oscillator $\ddot{q} = -q - \frac{1}{2}\dot{q}$ where a fraction β of the data is in the upper half plane while the remaining fraction $1 - \beta$ is in the lower half-plane. We set $\lambda = 0.5$, train on datasets with different β and compare learned conservative and nonconservative force fields with ground truth. We found the learned functions f_c^{NN} and f_n^{NN} are not necessarily aligned with the ground truth decomposition $f_c^{PHY} = -q$ and $f_n^{PHY} = -\frac{1}{2}\dot{q}$. We define misalignment as

$$m_c = \left[\frac{1}{nN}\sum_{i=1}^{N}\left\|f_c^{NN}(\mathbf{x}^{(i)}) - f_c^{PHY}(\mathbf{x}^{(i)})\right\|^2\right]^{\frac{1}{2}},$$
$$m_n = \left[\frac{1}{nN}\sum_{i=1}^{N}\left\|f_n^{NN}(\mathbf{x}^{(i)}, t^{(i)}) - f_n^{PHY}(\mathbf{x}^{(i)}, t^{(i)})\right\|^2\right]^{\frac{1}{2}}. \quad (12)$$

Figure 6(b) shows this misalignment as a function of β, revealing a minimum with nearly zero misalignment for the $\beta = 0.5$ case when the data is balanced. In summary, these last numerical experiments show that high-quality data is important for new physics discovery, regardless of whether the data is analyzed by intelligent human scientists or machine learning.

IV. CONCLUSION

We have presented the Neural New-physics Detector (NNPhD), a method for decomposing a general force field into components that do and do not conserve energy. We showed that NNPhD was able to do this robustly for a series of physical examples without access to symbolic equations, providing clear evidence of the existence of conservation-violating new physics. We also found that NNPhD could extrapolate time series more accurately than both LNN and unconstrained neural networks. As ever-larger science and engineering datasets become available for dynamical systems, we hope that NNPhD will help enable more accurate prediction as well as aid discovery of interesting new phenomena.

ACKNOWLEDGMENTS

We thank Yuanqi Du and Jieyu Zhang for helpful discussions, and we thank the Center for Brains, Minds, and Machines (CBMM) for hospitality. This work was supported by the Casey and Family Foundation, the Foundational Questions Institute, the Rothberg Family Fund for Cognitive Science, and AI Institute for Artificial Intelligence and Fundamental Interactions (IAIFI) through NSF Grant No. PHY-2019786.

APPENDIX A: TOY EXAMPLE DETAILS

For each dynamical system, the left-hand side is $\ddot{\mathbf{q}}$, and right-hand side is physical ground truth where conservative and nonconservative dynamics is explicitly separated as $\{f_c^{\mathrm{PHY}}(\mathbf{q}, \dot{\mathbf{q}})\} + \{f_n^{\mathrm{PHY}}(\mathbf{q}, \dot{\mathbf{q}}, t)\}$.

HO+MF ($k = B = 1$):

$$\begin{pmatrix}\ddot{x}\\ \ddot{y}\end{pmatrix} = \left\{-k\begin{pmatrix}x\\ y\end{pmatrix} + B\begin{pmatrix}\dot{y}\\ -\dot{x}\end{pmatrix}\right\} + \left\{\begin{pmatrix}0\\ 0\end{pmatrix}\right\}, \tag{A1}$$

HO+CG ($k = g = 1$):

$$\ddot{x} = \{-kx - g\} + \{0\}, \tag{A2}$$

HO+LD ($k = 1, \gamma = \frac{1}{2}$):

$$\ddot{x} = \{-kx\} + \{-\gamma\dot{x}\}, \tag{A3}$$

HO+CD ($k = 1, \gamma = \frac{1}{2}$):

$$\ddot{x} = \{-kx\} + \{-\gamma\,\mathrm{sgn}(\dot{x})\}, \tag{A4}$$

HO+PF ($k = 1, a = \frac{1}{2}$):

$$\ddot{x} = \{-kx\} + \{a\sin(t)\}. \tag{A5}$$

APPENDIX B: NEURAL NETWORK TRAINING DETAILS

We parametrize both the our LNN (conservative) and our unconstrained network (nonconservative) force models as non-weight-sharing fully connected feedforward neural networks with two hidden 200-neuron layers. The LNN uses a mixture of softplus and quadratic activation (see Appendix C for details) and has Eq. (3) hard-coded right before outputting f_c^{NN}, while the unconstrained network uses LeakyReLU activation (with negative slope $\alpha = 0.2$) and does not involve in any other inductive biases.

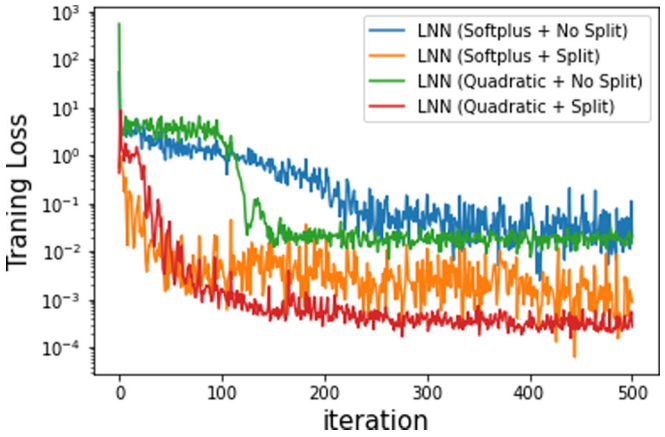

FIG. 7. Tricks to boost LNN training.

We measure the performance of NNPhD for

$$\lambda \in \{.01, .02, .05, .1, .2, .5, 1, 2, 5, 10, 20, 50, 100\} \tag{B1}$$

by first initializing and training NNPhD with $\lambda = 0.01$ using the ADAM optimizer with learning rate $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ for 2000 steps (500 steps for each learning rate), and iteratively increasing λ and train for 2000 steps for each new λ value. The model parameters of LNN and the unconstrained network are updated simultaneously.

APPENDIX C: TRICKS TO BOOST LNN TRAINING

As mentioned in Ref. [9], LNN is unstable and inefficient to train with traditional initializations in ML. As a result, expensive grid search of proper initializations is required. We propose two simpler tricks that have some improvements and are easy to implement. We use the example of a harmonic oscillator. The Lagrangian $\mathcal{L} = \frac{1}{2}\dot{q}^2 - \frac{1}{2}q^2$ contains only quadratic terms. We build a two hidden-layer networks with width 4-200-200-2.

Activation Trick. Reference [9] uses Softplus as activation function, which is general but inefficient to represent a quadratic function. However the quadratic function is common and useful in physics, we propose to divide neurons into two groups, where one group uses Softplus as activation, and the other group uses quadractic function as activation.

Split Trick. One of the instabilities in an LNN comes from inversion of $\nabla_{\dot{\mathbf{q}}}\nabla_{\dot{\mathbf{q}}}\mathcal{L}$. In physical terms, $\nabla_{\dot{\mathbf{q}}}\nabla_{\dot{\mathbf{q}}}\mathcal{L}$ represents a mass scalar (matrix) which is positive (positive definite). However this constraint is not explicitly embedded to LNN, leading to training instabilities. We split $\mathcal{L}$ into two parts:

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 = \mathcal{L}_{\mathrm{NN}} + \tfrac{1}{2}a\dot{\mathbf{q}}^T\dot{\mathbf{q}}, \tag{C1}$$

where $\mathcal{L}_1$ is learned by LNN, while $\mathcal{L}_2$ is a fixed quadratic term (we choose $a = 1$). At initializations when $\mathcal{L}_{\mathrm{NN}} \approx 0$, $\mathcal{L} \approx \frac{1}{2}a\dot{\mathbf{q}}^T\dot{\mathbf{q}}$ is positive definite.

To test how the proposed two tricks operate, we implement four models in Fig. 7 to fit the 1D harmonic oscillator: Softplus or quadratic activation, and with or without the split trick. The one with best performance is the LNN using quadratic activation and the split trick.

APPENDIX D: LAGRANGIAN MECHANICS FOR MACHINE LEARNING READERS

For readers whose background is primarily in machine learning rather than physics, this section provides a brief review of the Lagrangian mechanics formalism that we use in this paper.

Conservative dynamics describes a dynamic where there exist conserved quantities (energy, momentum, angular momentum etc). Conservation laws are important in physics, as they correspond to symmetries of our mother nature, according to Noether's theorem [46]. In particular, energy conservation is equivalent to time translational symmetry. To describe dynamics that conserves energy, physicists employ the (time-independent) *Lagrangian* or *Hamiltonian* formulation. Since our work and the prior work on LNN [9] are based on Lagrangian mechanics, we provide a brief introduction here.

The Lagrangian formalism models a classical physics system with trajectory $\mathbf{x}(t) = (\mathbf{q}, \dot{\mathbf{q}})$ that begins in one state $\mathbf{x}(t_0)$ and ends up in another state $\mathbf{x}(t_1)(t_1 > t_0)$, where $\mathbf{q}$ and $\dot{\mathbf{q}}$ are called the generalized coordinates and velocities, respectively. There are many paths that these states might take as they pass from $\mathbf{x}(t_0)$ to $\mathbf{x}(t_1)$, and Lagrangian mechanics tells that there is only one path that the physical system will take, i.e., the path that minimizes $\int_{t_0}^{t_1}\{T[\mathbf{q}(t), \dot{\mathbf{q}}(t)] - V[\mathbf{q}(t), \dot{\mathbf{q}}(t)]\}dt$, where T is kinetic energy and V is the potential energy. The term $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) \equiv T(\mathbf{q}, \dot{\mathbf{q}}) - V(\mathbf{q}, \dot{\mathbf{q}})$ is called Lagrangian and the path (trajectory) of the system is determined by the *Euler-Lagrange equation*:

$$\frac{d}{dt}\nabla_{\dot{\mathbf{q}}}\mathcal{L} = \nabla_{\mathbf{q}}\mathcal{L}.$$

Based on the formulas in the LNN [9], the Euler-Lagrange equation $\frac{d}{dt}\nabla_{\dot{\mathbf{q}}}\mathcal{L} = \nabla_{\mathbf{q}}\mathcal{L}$ can be rewritten by applying a chain rule $\frac{d}{dt}\nabla^T_{\dot{\mathbf{q}}}\mathcal{L} = (\nabla_{\dot{\mathbf{q}}}\nabla^T_{\dot{\mathbf{q}}}\mathcal{L})\ddot{\mathbf{q}} + (\nabla_{\mathbf{q}}\nabla_{\dot{\mathbf{q}}}\mathcal{L})\dot{\mathbf{q}}$ resulting in

$$\ddot{\mathbf{q}} = \left(\nabla_{\dot{\mathbf{q}}}\nabla^T_{\dot{\mathbf{q}}}\mathcal{L}\right)^{-1}\left[\nabla_{\mathbf{q}}\mathcal{L} - \left(\nabla_{\mathbf{q}}\nabla^T_{\dot{\mathbf{q}}}\mathcal{L}\right)\dot{\mathbf{q}}\right]. \tag{D1}$$

One inductive bias brought by Lagrangian mechanics is that Eq. (3) describes **conservative physical dynamics**. That is, the energy function defined as

$$H(\mathbf{q}, \dot{\mathbf{q}}) = \nabla^T_{\dot{\mathbf{q}}}\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) \tag{D2}$$

is constant along a trajectory $[\mathbf{q}(t), \dot{\mathbf{q}}(t)]$ driven by Eq. (3). The proof of $H(\mathbf{q}, \dot{\mathbf{q}})$ conservation can be found in standard physics textbooks [47] and is included here for completeness.

Lemma 1. Given a Lagrangian $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$, the energy defined in Eq. (3) is conserved along the trajectory $[\mathbf{q}(t), \dot{\mathbf{q}}(t)]$ driven by Eq. (2).

Proof. Invoking the chain rule one obtains the time derivative of $\mathcal{L}$:

$$\frac{d\mathcal{L}}{dt} = \dot{\mathbf{q}}^T\nabla_{\mathbf{q}}\mathcal{L} + \ddot{\mathbf{q}}^T\nabla_{\dot{\mathbf{q}}}\mathcal{L}. \tag{D3}$$

Equation (3) is equivalent to the Euler-Lagrangian equation $\frac{d}{dt}\nabla_{\dot{\mathbf{q}}}\mathcal{L} = \nabla_{\mathbf{q}}\mathcal{L}$, so we replace $\nabla_{\mathbf{q}}\mathcal{L}$ with $\frac{d}{dt}\nabla_{\dot{\mathbf{q}}}\mathcal{L}$:

$$\begin{aligned}\frac{d\mathcal{L}}{dt} &= \dot{\mathbf{q}}^T\frac{d}{dt}\nabla_{\dot{\mathbf{q}}}\mathcal{L} + \ddot{\mathbf{q}}^T\nabla_{\dot{\mathbf{q}}}\mathcal{L} \\ &= \frac{d}{dt}(\dot{\mathbf{q}}^T\nabla_{\dot{\mathbf{q}}}\mathcal{L}) \longrightarrow \frac{dH}{dt} \equiv \frac{d}{dt}(\dot{\mathbf{q}}^T\nabla_{\dot{\mathbf{q}}}\mathcal{L} - \mathcal{L}) = 0.\end{aligned} \tag{D4}$$

■

Since not all physical systems conserve energy, Lagrangian mechanics is insufficient to describe nonconservative dynamics, motivating the design of the NNPhD framework. We prove that linear damp example is nonconservative, i.e., it cannot be represented by Lagrangian mechanics.

Lemma 2. Let function $f : \mathbb{R}^2 \to \mathbb{R}$ be defined as $f(\mathbf{q}, \dot{\mathbf{q}}) = c\dot{\mathbf{q}}$, where c can be any real nonzero constant. Then, f cannot be represented by Eq. (3) for any Lagrangian $\mathcal{L} \in D^2(\mathbf{q}, \dot{\mathbf{q}})$ [$D^2(\mathbf{q}, \dot{\mathbf{q}})$ is the function space consisting of all twice-differentiable functions with respect to $(\mathbf{q}, \dot{\mathbf{q}})$].

Proof. We prove the claim by reduction to absurdity. Suppose there exists a Lagrangian $\mathcal{L} \in D^2(\mathbf{q}, \dot{\mathbf{q}})$, such that

$$c\dot{\mathbf{q}} = \left[\frac{\partial^2\mathcal{L}}{\partial\dot{\mathbf{q}}^2}(\mathbf{q}, \dot{\mathbf{q}})\right]^{-1}\left\{\frac{\partial\mathcal{L}}{\partial\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}) - \left[\frac{\partial^2\mathcal{L}}{\partial\dot{\mathbf{q}}\partial\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}})\right]\dot{\mathbf{q}}\right\}. \tag{D5}$$

By multiplying $[\frac{\partial^2\mathcal{L}}{\partial\dot{\mathbf{q}}^2}(\mathbf{q}, \dot{\mathbf{q}})]$ to both sides of Eq. (D5), we have

$$c\left[\frac{\partial^2\mathcal{L}}{\partial\dot{\mathbf{q}}^2}(\mathbf{q}, \dot{\mathbf{q}})\right]\dot{\mathbf{q}} = \frac{\partial\mathcal{L}}{\partial\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}) - \left[\frac{\partial^2\mathcal{L}}{\partial\dot{\mathbf{q}}\partial\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}})\right]\dot{\mathbf{q}},$$

which by Eq. (D2) further leads to

$$c\frac{\partial H}{\partial\dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}}) + \frac{\partial H}{\partial\mathbf{q}}(\mathbf{q}, \dot{\mathbf{q}}) = 0. \tag{D6}$$

By variable substitution, let $H(\mathbf{q}, \dot{\mathbf{q}}) = g(c\mathbf{q} + \dot{\mathbf{q}}, \dot{\mathbf{q}} - c\mathbf{q})$. By Eq. (D6),

$$\begin{aligned}\frac{\partial g(c\mathbf{q} + \dot{\mathbf{q}}, \dot{\mathbf{q}} - c\mathbf{q})}{\partial(c\mathbf{q} + \dot{\mathbf{q}})} &= \frac{\partial H(\mathbf{q}, \dot{\mathbf{q}})}{\partial\mathbf{q}}\frac{\partial\mathbf{q}}{\partial(c\mathbf{q} + \dot{\mathbf{q}})} \\ &\quad + \frac{\partial H(\mathbf{q}, \dot{\mathbf{q}})}{\partial\dot{\mathbf{q}}}\frac{\partial\dot{\mathbf{q}}}{\partial(c\mathbf{q} + \dot{\mathbf{q}})} \\ &= \frac{1}{2c}\frac{\partial H(\mathbf{q}, \dot{\mathbf{q}})}{\partial\mathbf{q}} + \frac{1}{2}\frac{\partial H(\mathbf{q}, \dot{\mathbf{q}})}{\partial\dot{\mathbf{q}}} = 0.\end{aligned}$$

Therefore, $H(\mathbf{q}, \dot{\mathbf{q}})$ is invariant of $c\mathbf{q} + \dot{\mathbf{q}}$ and only relies on the value of $\dot{\mathbf{q}} - c\mathbf{q}$. Thus, we can further abbreviate $H(\mathbf{q}, \dot{\mathbf{q}})$ as $g(\dot{\mathbf{q}} - c\mathbf{q})$.

However, by Eq. (D2),

$$g(-c\mathbf{q}) = g(0 - c\mathbf{q}) = H(\mathbf{q}, \dot{\mathbf{q}}) = -\mathcal{L}(\mathbf{q}, 0).$$

Therefore,

$$\begin{aligned}\left.\frac{\partial\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial\dot{\mathbf{q}}}\right|_{\dot{\mathbf{q}}=0} &= \lim_{\dot{\mathbf{q}}\to 0}\frac{\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{L}(\mathbf{q}, 0)}{\dot{\mathbf{q}}} \\ &= \lim_{\dot{\mathbf{q}}\to 0}\frac{\dot{\mathbf{q}}\frac{\partial\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial\dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}}) - g(\dot{\mathbf{q}} - c\mathbf{q}) + g(-c\mathbf{q})}{\dot{\mathbf{q}}} \\ &\overset{(*)}{=} \left.\frac{\partial\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial\dot{\mathbf{q}}}\right|_{\dot{\mathbf{q}}=0} - g'(-c\mathbf{q}),\end{aligned}$$

where Eq. $(*)$ is due to that $\frac{\partial\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial\dot{\mathbf{q}}}(\mathbf{q}, \dot{\mathbf{q}})$ is differentiable (thus continuous).

Therefore, we have $g'(\mathbf{q}) = 0$ for any $\mathbf{q}$, which further leads to $H(\mathbf{q}, \dot{\mathbf{q}})$ is a constant function and

$$c\dot{\mathbf{q}} = -\left[\frac{\partial^2\mathcal{L}}{\partial\dot{\mathbf{q}}^2}(\mathbf{q}, \dot{\mathbf{q}})\right]^{-1}\frac{\partial H(\mathbf{q}, \dot{\mathbf{q}})}{\partial\mathbf{q}} = 0.$$

The proof is completed since $c \neq 0$. ■

APPENDIX E: LEARNING PERSPECTIVES OF SECTION II C

We describe the learning task based on the force decomposition in Sec. II C. Given samples $(\mathbf{q}^{(i)}, \dot{\mathbf{q}}^{(i)}, t^{(i)}; \ddot{\mathbf{q}}^{(i)})$, $i = 1, \cdots, N$ that are uniformly drawn from the trajectories of dynamic $\ddot{\mathbf{q}} = f(\mathbf{q}, \dot{\mathbf{q}}, t)$ with $t \in [0, T]$ or a given distribution μ, we aim to learn both f_{c} and f_{n} from data. Because the ground-truth dynamic and its vector space are unknown, we need to select a model space $(\mathcal{G}, \|\cdot\|_{\mathcal{G}})$ which is also a normed vector space to find the best model in it. For this learning problem, we learn the model pair $(f_{\mathrm{c}}^{\mathrm{NN}}, f_{\mathrm{n}}^{\mathrm{NN}})$ simultaneously by solving the following constrained minimization problem

$$\min_{(f_{\mathrm{n}}^{\mathrm{NN}}, f_{\mathrm{c}}^{\mathrm{NN}})} \mathcal{L}_S\big(f_{\mathrm{n}}^{\mathrm{NN}}, f_{\mathrm{c}}^{\mathrm{NN}}\big) = \frac{1}{N}\sum_{i=1}^{N} \big\| f_{\mathrm{c}}^{\mathrm{NN}}(\mathbf{q}^{(i)}, \dot{\mathbf{q}}^{(i)}) + f_{\mathrm{n}}^{\mathrm{NN}}(\mathbf{q}^{(i)}, \dot{\mathbf{q}}^{(i)}, t^{(i)}) - \ddot{\mathbf{q}}^{(i)} \big\|_{\mathcal{G}}$$

$$\text{s.t.} \quad f_{\mathrm{c}}^{\mathrm{NN}} = \arg\min_{g \in \mathcal{G}_c} \frac{1}{N}\sum_{i=1}^{N} \|\ddot{\mathbf{q}}^{(i)} - g(\mathbf{q}^{(i)}, \dot{\mathbf{q}}^{(i)})\|_{\mathcal{G}}.$$

We make the following discussions on the learning task: We denote the optimal models of the above optimization problem as $(f_{\mathrm{c}}^{\mathrm{NN}*}, f_{\mathrm{n}}^{\mathrm{NN}*})$. The interpolating prediction ability [which is measured by the gap between $\int_0^T \|f_{\mathrm{c}}^{\mathrm{NN}*} + f_{\mathrm{n}}^{\mathrm{NN}*} - \ddot{\mathbf{q}}\| dt$ and $\mathcal{L}_S(f_{\mathrm{c}}^{\mathrm{NN}*}, f_{\mathrm{n}}^{\mathrm{NN}*})$] is determined by the approximation ability of $\mathcal{G}$ and the number of training data. As the number of training data N increases, the gap will be smaller. As the approximation ability of $\mathcal{G}$ becomes stronger, the gap will be smaller.

APPENDIX F: THEOREM 1 (FORMAL)

In this section, we will provide proof of Theorem 1. We will actually show our results hold for general norms which include the discrete norm we use in experiments. Concretely, the formal version of Theorem 1 with general norms is given as follows:

Theorem 1 (Theorem 1, extended to general norms). Let $f_{\mathrm{c}}^{NN}(\boldsymbol{w}_{\mathrm{c}})$ be the Lagrangian neural network with parameters $\boldsymbol{w}_{\mathrm{c}}$ in NNPhD framework (with input $\mathbf{q}$ and $\dot{\mathbf{q}}$ omitted), and $f_{\mathrm{n}}^{NN}(\boldsymbol{w}_{\mathrm{n}})$ be the unconstrained neural network with parameters $\boldsymbol{w}_{\mathrm{n}}$ in the NNPhD framework. Assume the UNN can represent every continuous function of $(\mathbf{q}, \dot{\mathbf{q}}, t)$ under the norm $\|\cdot\|_\star$, i.e., $\{g : \mathbb{R}^{2n+1} \to \mathbb{R}, \exists \boldsymbol{w}_{\mathrm{n}},\ \|g - f_{\mathrm{n}}^{NN}(\boldsymbol{w}_{\mathrm{n}})\|_\star = 0\} = \mathcal{C}(\mathbf{q}, \dot{\mathbf{q}}, t)$. Then, given any continuous function f, the following claim stands:

(1) For $\lambda > 1$,

$$\arg\min_{\boldsymbol{w}_{\mathrm{c}}, \boldsymbol{w}_{\mathrm{n}}} \big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}}) - f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star + \lambda\big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star = \big\{(\boldsymbol{w}_{\mathrm{c}}, \boldsymbol{w}_{\mathrm{n}}) : \boldsymbol{w}_{\mathrm{c}} \in \arg\min_{\boldsymbol{w}_{\mathrm{c}}} \big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}})\big\|_\star, \quad \big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star = 0\big\}.$$

(2) For $0 < \lambda < 1$,

$$\arg\min_{\boldsymbol{w}_{\mathrm{c}}, \boldsymbol{w}_{\mathrm{n}}} \big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}}) - f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star + \lambda\big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star = \big\{(\boldsymbol{w}_{\mathrm{c}}, \boldsymbol{w}_{\mathrm{n}}) : \boldsymbol{w}_{\mathrm{c}} \in \arg\min_{\boldsymbol{w}_{\mathrm{c}}} \big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}})\big\|_\star, \big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}}) - f + f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}})\big\|_\star = 0\big\}.$$

Remark 1. The informal version of Theorem 1 in the main text is a special case of the formal version, with

$$\|f\|_\star = \left(\frac{1}{Nn}\sum_{i=1}^{N} \|f(x^{(i)}, t^{(i)})\|^p\right)^{\frac{1}{p}},$$

for any function $f : \mathbb{R}^{2n+1} \to \mathbb{R}^n$.

Proof. We prove the two cases above separately.

(1) If $\lambda > 1$, for any $\boldsymbol{w}_{\mathrm{c}}$ and $\boldsymbol{w}_{\mathrm{n}}$, then

$$\begin{aligned}\big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}}) - f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star + \lambda\big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star &= \big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}}) - f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star + \big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star + (\lambda - 1)\big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star \\ &\overset{(*)}{\geqslant} \big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}})\big\|_\star + (\lambda - 1)\big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star. \end{aligned} \tag{F1}$$

Since $\lambda - 1 > 0$,

$$\arg\min_{\boldsymbol{w}_{\mathrm{c}}, \boldsymbol{w}_{\mathrm{n}}} \Big(\big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}})\big\|_\star + (\lambda - 1)\big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star\Big) = \Big(\arg\min_{\boldsymbol{w}_{\mathrm{c}}} \big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}})\big\|_\star, \arg_{\boldsymbol{w}_{\mathrm{n}}} \big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star = 0\Big).$$

For any $(\boldsymbol{w}_{\mathrm{c}}^0, \boldsymbol{w}_{\mathrm{n}}^0)$ where $\boldsymbol{w}_{\mathrm{c}}^0 \in \arg\min_{\boldsymbol{w}_{\mathrm{c}}} \|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}})\|_\star$ and $\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}}^0)\|_\star = 0$, the equality of inequality $(*)$ of Eq. (F1) is obtained. Therefore,

$$\big(\boldsymbol{w}_{\mathrm{c}}^0, \boldsymbol{w}_{\mathrm{n}}^0\big) \in \arg\min_{\boldsymbol{w}_{\mathrm{c}}, \boldsymbol{w}_{\mathrm{n}}} \big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}}) - f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star + \lambda\big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star,$$

which further leads to

$$\arg\min_{\boldsymbol{w}_{\mathrm{c}}, \boldsymbol{w}_{\mathrm{n}}} \Big[\big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}})\big\|_\star + (\lambda - 1)\big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star\Big] \subset \arg\min_{\boldsymbol{w}_{\mathrm{c}}, \boldsymbol{w}_{\mathrm{n}}} \Big[\big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}}) - f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star + \lambda\big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star\Big],$$

and

$$\min_{\boldsymbol{w}_{\mathrm{c}}, \boldsymbol{w}_{\mathrm{n}}} \Big[\big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}}) - f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star + \lambda\big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star\Big] = \min_{\boldsymbol{w}_{\mathrm{c}}, \boldsymbol{w}_{\mathrm{n}}} \Big[\big\|f - f_{\mathrm{c}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{c}})\big\|_\star + (\lambda - 1)\big\|f_{\mathrm{n}}^{\mathrm{NN}}(\boldsymbol{w}_{\mathrm{n}})\big\|_\star\Big]. \tag{F2}$$

Combining Eqs. (F1) and (F2) further leads to

$$\arg\min_{\boldsymbol{w}_c,\boldsymbol{w}_n}\left[\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\right\|_\star+\lambda\left\|f_n^{NN}(\boldsymbol{w}_n)\right\|_\star\right]\subset\arg\min_{\boldsymbol{w}_c,\boldsymbol{w}_n}\left[\left\|f-f_c^{NN}(\boldsymbol{w}_c)\right\|_\star+(\lambda-1)\left\|f_n^{NN}(\boldsymbol{w}_n)\right\|_\star\right].$$

The proof for $\lambda>1$ is completed.

(2) If $\lambda<1$, for any $\boldsymbol{w}_c$ and $\boldsymbol{w}_n$, then we decompose $\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\|_\star+\lambda\|f_n^{NN}(\boldsymbol{w}_n)\|_\star$ as follows:

$$\begin{aligned}\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\right\|_\star+\lambda\left\|f_n^{NN}(\boldsymbol{w}_n)\right\|_\star &= (\lambda+(1-\lambda))\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\right\|_\star+\lambda\left\|f_n^{NN}(\boldsymbol{w}_n)\right\|_\star\\ &\overset{(**)}{\geqslant}\lambda\left\|f-f_c^{NN}(\boldsymbol{w}_c)\right\|_\star+(1-\lambda)\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\right\|_\star,\end{aligned}\tag{F3}$$

where Eq. $(**)$ is due to the triangle inequality.

However, for any fixed $\boldsymbol{w}_c$, the minimum of Eq. (F3) is obtained if and only if $\|f_n^{NN}(\boldsymbol{w}_n)-f+f_c^{NN}(\boldsymbol{w}_c)\|_\star=0$, in which case equality of inequality $(**)$ of Eq. (F3) is also obtained. Therefore, for a given $\boldsymbol{w}_c$,

$$\min_{\boldsymbol{w}_n}\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\right\|_\star+\lambda\left\|f_n^{NN}(\boldsymbol{w}_n)\right\|_\star=\lambda\left\|f-f_c^{NN}(\boldsymbol{w}_c)\right\|_\star,$$

and

$$\arg\min_{\boldsymbol{w}_n}\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\right\|_\star+\lambda\left\|f_n^{NN}(\boldsymbol{w}_n)\right\|_\star=\left\{\boldsymbol{w}_n:\left\|f_n^{NN}(\boldsymbol{w}_n)-f+f_c^{NN}(\boldsymbol{w}_c)\right\|_\star=0\right\}.\tag{F4}$$

Since

$$\arg\min_{\boldsymbol{w}_c,\boldsymbol{w}_n}\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\right\|_\star+\lambda\left\|f_n^{NN}(\boldsymbol{w}_n)\right\|_\star=\arg\min_{\boldsymbol{w}_c}\min_{\boldsymbol{w}_n}\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\right\|_\star+\lambda\left\|f_n^{NN}(\boldsymbol{w}_n)\right\|_\star,$$

by applying Eq. (F4), we finally have

$$\begin{aligned}&\arg\min_{\boldsymbol{w}_c,\boldsymbol{w}_n}\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\right\|_\star+\lambda\left\|f_n^{NN}(\boldsymbol{w}_n)\right\|_\star\\ &=\left\{(\boldsymbol{w}_c,\boldsymbol{w}_n):\boldsymbol{w}_c\in\arg\min_{\boldsymbol{w}_c}\lambda\left\|f-f_c^{NN}(\boldsymbol{w}_c)\right\|_\star,\left\|f_n^{NN}(\boldsymbol{w}_n)-f+f_c^{NN}(\boldsymbol{w}_c)\right\|_\star=0\right\}\\ &=\left\{(\boldsymbol{w}_c,\boldsymbol{w}_n):\boldsymbol{w}_c\in\arg\min_{\boldsymbol{w}_c}\left\|f-f_c^{NN}(\boldsymbol{w}_c)\right\|_\star,\left\|f_n^{NN}(\boldsymbol{w}_n)-f+f_c^{NN}(\boldsymbol{w}_c)\right\|_\star=0\right\}.\end{aligned}$$

The proof is completed. ■

The above theorem describes the case that $f_n^{NN}(\boldsymbol{w}_n)$ can represent every continuous function. However, in practice, the UNN neural network can only access functions close to the original solution. In this general case, we instead have

Corollary 1.1. Let $f_c^{NN}(\boldsymbol{w}_c)$ be the Lagrangian neural network with parameters $\boldsymbol{w}_c$ in NNPhD framework, and $f_n^{NN}(\boldsymbol{w}_n)$ be the UNN network with parameters $\boldsymbol{w}_n$ in the NNPhD framework. Assume the UNN neural network can approximate every continuous function of $t,\mathbf{q},\dot{\mathbf{q}}$ by error $\varepsilon>0$ under some norm $\|\cdot\|_\star$ on function space $\mathcal{C}(\mathbf{q},\dot{\mathbf{q}},t)$, i.e., $\forall f\in\mathcal{C}(\mathbf{q},\dot{\mathbf{q}},t)$, there exists a $\boldsymbol{w}_n^f$, such that, $\|f_n^{NN}(\boldsymbol{w}_n^f)-f\|_\star\leqslant\varepsilon$. Furthermore, assume there exists $\boldsymbol{w}_0$, such that, $f_n^{NN}(\boldsymbol{w}_0)\equiv0$. Then, given any continuous function f, the following claim stands:

(1) For $\lambda>1$,

$$\arg\min_{\boldsymbol{w}_c,\boldsymbol{w}_n}\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\right\|_\star+\lambda\left\|f_n^{NN}(\boldsymbol{w}_n)\right\|_\star=\left[\arg\min_{\boldsymbol{w}_c}\left\|f-f_c^{NN}(\boldsymbol{w}_c)\right\|_\star,0\right].$$

(2) For $\lambda<1$, for any $(\boldsymbol{w}_c^0,\boldsymbol{w}_n^0)\in\arg\min_{\boldsymbol{w}_c,\boldsymbol{w}_n}\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\|_\star+\lambda\|f_n^{NN}(\boldsymbol{w}_n)\|_\star$,

$$\left\|f_n^{NN}\left(\boldsymbol{w}_n^0\right)-f+f_c^{NN}\left(\boldsymbol{w}_c^0\right)\right\|_\star\leqslant\frac{1+\lambda}{1-\lambda}\varepsilon,\tag{F5}$$

$$\left\|f-f_c^{NN}\left(\boldsymbol{w}_c^0\right)\right\|_\star\leqslant\min_{\boldsymbol{w}_c}\left\|f-f_c^{NN}(\boldsymbol{w}_c)\right\|_\star+(1+\lambda)\varepsilon.\tag{F6}$$

Proof. When $\lambda>1$, the claim can be proved following exactly the same routine for Theorem 1. When $\lambda<1$, we follow the same routine for Theorem 1 to decompose the optimization problem into a two step minimization problem: fixed $\boldsymbol{w}_c$, let $g=f-f_c^{NN}(\boldsymbol{w}_c)$. Then, there exists a $\boldsymbol{w}_n^g$, such that

$$\left\|f_n^{NN}\left(\boldsymbol{w}_n^g\right)-g\right\|_\star\leqslant\varepsilon.$$

Therefore,

$$\begin{aligned}\min_{\boldsymbol{w}_n}\left[\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}(\boldsymbol{w}_n)\right\|_\star+\lambda\left\|f_n^{NN}(\boldsymbol{w}_n)\right\|_\star\right]&\leqslant\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}\left(\boldsymbol{w}_n^g\right)\right\|_\star+\lambda\left\|f_n^{NN}\left(\boldsymbol{w}_n^g\right)\right\|_\star\\ &\leqslant(1+\lambda)\left\|f-f_c^{NN}(\boldsymbol{w}_c)-f_n^{NN}\left(\boldsymbol{w}_n^g\right)\right\|_\star+\lambda\left\|f-f_c^{NN}(\boldsymbol{w}_c)\right\|_\star\\ &\leqslant\lambda\left\|f-f_c^{NN}(\boldsymbol{w}_c)\right\|_\star+(1+\lambda)\varepsilon.\end{aligned}$$

Let $\boldsymbol{w}_\mathrm{n} = \arg\min_{\boldsymbol{w}_\mathrm{n}}(\|f - f_\mathrm{c}^\mathrm{NN}(\boldsymbol{w}_\mathrm{c}) - f_\mathrm{n}^\mathrm{NN}(\boldsymbol{w}_\mathrm{n})\|_\star + \lambda\|f_\mathrm{n}^\mathrm{NN}(\boldsymbol{w}_\mathrm{n})\|_\star)$. Then,

$$(1-\lambda)\left\|f - f_\mathrm{c}^\mathrm{NN}(\boldsymbol{w}_\mathrm{c}) - f_\mathrm{n}^\mathrm{NN}(\boldsymbol{w}_\mathrm{n})\right\|_\star \leqslant \left\|f - f_\mathrm{c}^\mathrm{NN}(\boldsymbol{w}_\mathrm{c}) - f_\mathrm{n}^\mathrm{NN}(\boldsymbol{w}_\mathrm{n})\right\|_\star + \lambda\left\|f_\mathrm{n}^\mathrm{NN}(\boldsymbol{w}_\mathrm{n})\right\|_\star - \lambda\left\|f - f_\mathrm{c}^\mathrm{NN}(\boldsymbol{w}_\mathrm{c}))\right\|_\star$$
$$\leqslant (1+\lambda)\varepsilon, \tag{F7}$$

which finishes the proof of Eq. (F5).

Let $h(\boldsymbol{w}_\mathrm{c}) = \min_{\boldsymbol{w}_\mathrm{n}} \|f - f_\mathrm{c}^\mathrm{NN}(\boldsymbol{w}_\mathrm{c}) - f_\mathrm{n}^\mathrm{NN}(\boldsymbol{w}_\mathrm{n})\|_\star + \lambda\|f_\mathrm{n}^\mathrm{NN}(\boldsymbol{w}_\mathrm{n})\|_\star$. By Eq. (F7),

$$\|h(\boldsymbol{w}_\mathrm{c})\|_\star - \lambda\left\|f - f_\mathrm{c}^\mathrm{NN}(\boldsymbol{w}_\mathrm{c})\right\|_\star \leqslant (1+\lambda)\varepsilon,$$

which further leads to

$$\min_{\boldsymbol{w}_\mathrm{c}} \|h(\boldsymbol{w}_\mathrm{c})\|_\star \leqslant (1+\lambda)\varepsilon + \lambda \min_{\boldsymbol{w}_\mathrm{c}} \left\|f - f_\mathrm{c}^\mathrm{NN}(\boldsymbol{w}_\mathrm{c})\right\|_\star.$$

This completes the proof. ■

[1] C. Burgess and G. Moore, *The Standard Model: A Primer* (Cambridge University Press, Cambridge, MA, 2007).

[2] I. M. Hutchings, Leonardo da Vinci studies of friction, Wear **360-361**, 51 (2016).

[3] Wikipedia contributors, Discovery of neptune—Wikipedia, the free encyclopedia, retrieved from https://en.wikipedia.org/w/index.php?title=Discovery_of_Neptune&oldid=1000734782 (2021).

[4] Wikipedia contributors, Cowan-Reines neutrino experiment—Wikipedia, the free encyclopedia, retrieved from https://en.wikipedia.org/w/index.php?title=Cowan%E2%80%93Reines_neutrino_experiment&oldid=1000625707 (2021).

[5] M. S. Turner, The dark side of the universe: From Zwicky to accelerated expansion, Phys. Rep. **333**, 619 (2000).

[6] V. C. Rubin, Dark matter in spiral galaxies, Sci. Am. **248**, 96 (1983).

[7] A. Wolszczan and D. A. Frail, A planetary system around the millisecond pulsar psr1257+ 12, Nature (London) **355**, 145 (1992).

[8] L. Esposito and E. Harrison, Properties of the Hulse-Taylor binary pulsar system, Astrophys. J. **196**, L1 (1975).

[9] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, Lagrangian neural networks, ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations (2020).

[10] S. Greydanus, M. Dzamba, and J. Yosinski, Hamiltonian neural networks, in *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, MA, 2019), pp. 15379–15389.

[11] M. Finzi, K. A. Wang, and A. G. Wilson, Simplifying Hamiltonian and Lagrangian neural networks via explicit constraints, *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Curran Associates, Inc., 2020), Vol. 33, pp. 13880–13889.

[12] A. Choudhary, J. F. Lindner, E. G. Holliday, S. T. Miller, S. Sinha, and W. L. Ditto, Forecasting Hamiltonian dynamics without canonical coordinates, Nonlinear Dynamics **103**, 1553 (2021).

[13] P. Jin, Z. Zhang, A. Zhu, Y. Tang, and G. E. Karniadakis, Sympnets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems, Neural Netw. **132**, 166 (2020).

[14] P. Toth, D. J. Rezende, A. Jaegle, S. Racanière, A. Botev, and I. Higgins, Hamiltonian Generative Networks, International Conference on Learning Representations (2020).

[15] Z. Long, Y. Lu, X. Ma, and B. Dong, PDE-Net: Learning PDEs from data, in *Proceedings of the International Conference on Machine Learning* (PMLR, Vienna, Austria, 2018), pp. 3208–3216.

[16] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, Neural ordinary differential equations, Adv. Neural Info. Process. Syst. **31**, 6571 (2018).

[17] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, and K. Kavukcuoglu, Interaction networks for learning about objects, relations and physics, *Advances in Neural Information Processing Systems*, edited by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Inc., 2016), Vol. 29.

[18] F. Alet, E. Weng, T. Lozano-Pérez, and L. P. Kaelbling, Neural relational inference with fast modular meta-learning, in *Advances in Neural Information Processing Systems*, Vol. 32, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Red Hook, NY, 2019), pp. 11827–11838.

[19] P. Y. Lu, S. Kim, and M. Soljačić, Extracting Interpretable Physical Parameters from Spatiotemporal Systems using Unsupervised Learning, Phys. Rev. X **10**, 031056 (2020).

[20] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, Data-driven discovery of coordinates and governing equations, Proc. Natl. Acad. Sci. USA **116**, 22445 (2019).

[21] S.-M. Udrescu and M. Tegmark, Symbolic pregression: Discovering physical laws from raw distorted video, Phys. Rev. E **103**, 043307 (2021).

[22] S. Kim, P. Y. Lu, S. Mukherjee, M. Gilbert, L. Jing, V. Čeperić, and M. Soljačić, Integration of neural network-based symbolic regression in deep learning for scientific discovery, *IEEE Transactions on Neural Networks and Learning Systems* (IEEE, 2020).

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

[24] T. Matsubara, A. Ishikawa, and T. Yaguchi, Deep energy-based modeling of discrete-time physics, *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Curran Associates, Inc., 2020), Vol. 33, pp. 13100–13111.

[25] M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho, Discovering symbolic models from deep learning with inductive biases, *Advances in Neural Information Processing Systems*, edited by H. Larochelle,

M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Curran Associates, Inc., 2020), Vol. 33, pp. 17429–17442.

[26] M. Raissi, A. Yazdani, and G. E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, Science **367**, 1026 (2020).

[27] T. Wu and M. Tegmark, Toward an artificial intelligence physicist for unsupervised learning, Phys. Rev. E **100**, 033311 (2019).

[28] S.-H. Li, C.-X. Dong, L. Zhang, and L. Wang, Neural Canonical Transformation with Symplectic Flows, Phys. Rev. X **10**, 021020 (2020).

[29] M. Lutter, C. Ritter, and J. Peters, Deep Lagrangian Networks Using Physics as Model Prior for Deep Learning, International Conference on Learning Representations (2019).

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

[31] G. Welch, G. Bishop *et al.*, *An Introduction to the Kalman Filter* (Chapel Hill, NC, USA, 1995).

[32] Y. Yin, V. L. Guen, J. Dona, E. de Bézenac, I. Ayed, N. Thome, and P. Gallinari, Augmenting physical models with deep networks for complex dynamics forecasting, International Conference on Learning Representations (2020).

[33] A. Ajay, J. Wu, N. Fazeli, M. Bauza, L. P. Kaelbling, J. B. Tenenbaum, and A. Rodriguez, Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, Piscataway, NJ, 2018), pp. 3066–3073.

[34] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, Galileo: Perceiving physical object properties by integrating a physics engine with deep learning, Adv. Neural Info. Process. Syst. **28**, 127 (2015).

[35] Y. D. Zhong, B. Dey, and A. Chakraborty, Dissipative symoden: Encoding Hamiltonian dynamics with dissipation and control into deep learning, ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations (2020).

[36] A. R. Barron, Universal approximation bounds for superpositions of a sigmoidal function, IEEE Trans. Inf. Theory **39**, 930 (1993).

[37] H. W. Lin, M. Tegmark, and D. Rolnick, Why does deep and cheap learning work so well? J. Stat. Phys. **168**, 1223 (2017).

[38] D. Rolnick and M. Tegmark, The power of deeper networks for expressing natural functions, International Conference on Learning Representations (2018).

[39] Wikipedia contributors, Discovery of neptune—Wikipedia, the free encyclopedia, retrieved from https://en.wikipedia.org/w/index.php?title=Discovery_of_Neptune&oldid=1029263211 (2021).

[40] Wikipedia contributors, Hulse-taylor binary—Wikipedia, the free encyclopedia, retrieved from https://en.wikipedia.org/w/index.php?title=Hulse%E2%80%93Taylor_binary&oldid=1040338055 (2021).

[41] Caltech Press, 2017 Nobel prize in physics awarded to Ligo founders, retrieved from https://www.ligo.caltech.edu/page/press-release-2017-nobel-prize (2017).

[42] L. Scientific, V. collaborations, B. Abbott, R. Abbott, T. Abbott, M. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso *et al.*, The basic physics of the binary black hole merger GW150914, Ann. Phys. **529**, 1600209 (2017).

[43] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach, Phys. Rev. Lett. **120**, 024102 (2018).

[44] S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural Comput. **9**, 1735 (1997), retrieved from https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf.

[45] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, Neural ordinary differential equations, *Advances in Neural Information Processing Systems*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett (Curran Associates, Inc., 2018), Vol. 31.

[46] J. Hanc, S. Tuleja, and M. Hancova, Symmetries and conservation laws: Consequences of Noether's theorem, Am. J. Phys. **72**, 428 (2004).

[47] H. Goldstein, C. Poole, and J. Safko, Classical Mechanics (Pearson Education India, 2011).