Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2024-0052

Vol. **39**, No. 1, pp. 59-77
January 2026

Generalized Lagrangian Neural Networks

Shanshan Xiao[1,2], Jiawei Zhang[1,2] and Yifa Tang[1,2,*]

[1] *LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China.*
[2] *School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China.*

Received 8 March 2024; Accepted (in revised version) 11 February 2025

Abstract. Incorporating neural networks for the inverse problem of solution of Ordinary Differential Equations (ODEs) represents a pivotal research direction within computational mathematics. Within neural network architectures, the integration of the intrinsic structure of ODEs offers advantages such as enhanced prediction accuracy and reduced data utilization. Among these structural ODE forms, the Lagrangian representation stands out due to its significant physical underpinnings. Building upon this framework, Bhattoo introduced the concept of Lagrangian Neural Networks (LNNs). Then in this article, we introduce an extension (Generalized Lagrangian Neural Networks) to Lagrangian Neural Networks (LNNs) mainly based on mathematics and physics, innovatively tailoring them for non-conservative systems. By leveraging the foundational importance of the Lagrangian within Lagrange's equations, we formulate the model based on the generalized Lagrange's equation. This modification not only enhances prediction accuracy but also guarantees Lagrangian representation in non-conservative systems. Furthermore, we perform various experiments, encompassing 1-dimensional and 2-dimensional examples, along with an examination of the impact of network parameters, which proved the superiority of Generalized Lagrangian Neural Networks (GLNNs).

AMS subject classifications: 68T05

Key words: Non-conservative system, Lagrangian system, neural networks.

1 Introduction

Machine learning has found significant applications in the field of mathematics, revolutionizing traditional approaches to problem-solving and analysis. With its ability to automatically learn patterns and make predictions from data, machine learning techniques

*Corresponding author. *Email addresses:* xss@lsec.cc.ac.cn (S. Xiao), zhangjiawei19@lsec.cc.ac.cn (J. Zhang), tyf@lsec.cc.ac.cn (Y. Tang)

have been successfully applied to various mathematical tasks. One significant direction is the utilization of machine learning to address mathematical problems associated with differential equations and dynamical systems [1,5–11,14].

In the existing methods for solving dynamical system problems using machine learning, they can be mainly categorized into two types. One type is unstructured methods [7,11,25,31,33–35], which do not consider the physical or mathematical structure of the equations or dynamical systems. When employing this approach, the focus is typically on minimizing the model error or complexity. For example, symbolic regression [33] is a regression method where the control equation under study is treated as an unknown target, and this unknown target is regarded as a function of the state variable data and its time derivative, using certain sparse approximations; Galerkin-closure methods [35] aim to approximate the unresolved scales of turbulence by using a closure model based on a truncated set of resolved scales; furthermore, unstructured neural networks such as LSTM [34] can be used to directly predict the solutions of differential equations or the phase flow of dynamical systems.

In contrast to unstructured methods, considering the inherent properties of the system and utilizing neural networks with specific structures can significantly enhance the predictive performance on specific systems. By incorporating the knowledge of system properties, such as conservation laws, symmetries, or known mathematical structures, into the design of neural networks, it becomes possible to improve the accuracy and effectiveness of predictions for the targeted system [1,6,8–11,15,16,18]. Hamiltonian Neural Network (HNN) [11], is a typical example of a structured neural network that takes into account the geometric structure of Hamiltonian systems. It utilizes neural networks to approximate the Hamiltonian of the system, thereby achieving improved predictive performance. OnsagerNets [14], as a systematic method that can overcome the aforementioned limitations, are base on a highly general extension of the Onsager principle for dissipative dynamics. Lagrangian Neural Networks (LNN) [1] are a class of neural networks specifically designed to parameterize arbitrary Lagrangians through their network architecture. Unlike traditional approaches, LNNs do not impose restrictions on the functional form of the learned energies, allowing them to produce models that conserve energy.

LNNs have demonstrated exceptional performance in many tasks. However, LNNs are limited to Lagrangian systems that adhere to the principle of energy conservation. However, in practical problems, we often encounter non-conservative systems. In the study of Euler-Lagrange equations [2,3], it is known that non-conservative systems can be formulated in the form of generalized Euler-Lagrange equations [5], which can be expressed as follows:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_k}\right)-\frac{\partial \mathcal{L}}{\partial q_k}=F_k.$$

In this article, we extend the scope of LNNs by constructing neural networks specifically designed for non-conservative systems. Our main inspiration stems from the special physical significance of the Lagrangian in Lagrange's equations. Since the Lagrangian

can be understood as the difference between kinetic energy and potential energy, authors replace the baseline model with a fitted Lagrangian in Lagrangian Neural Networks (LNNs). The physical interpretation of the Lagrangian and the structure of Lagrange's equations contribute to higher prediction accuracy and enable the energy of non-conservative systems to fluctuate within a narrow range (as opposed to energy dissipation in the baseline model). Building upon this, we consider the generalized Lagrange's equations, which incorporate non-conservative terms, akin to the conventional Lagrange's equations. In the generalized Lagrange's equations, the Lagrangian $\mathcal{L}$ holds the same physical interpretation. Thus, our objective is to construct a neural network suitable for non-conservative systems using the framework of generalized Lagrange's equations. To achieve this, in Chapter 2, we first present the mathematical and physical properties of Lagrange's equations, along with methods to express most physically motivated non-conservative systems in the form of generalized Lagrange's equations. Consequently, we can regard the majority of physical non-conservative systems as governed by a generalized Lagrange's equation.

Building upon this theoretical foundation, we introduce the Generalized Lagrangian Neural Networks (GLNNs). We conceptualize non-conservative systems as dynamic systems governed by generalized Lagrange's equations, and employ neural networks that preserve the structural integrity of these generalized equations to model such dynamics. Given the physical interpretations embedded within the generalized Lagrange's equations, GLNNs demonstrate enhanced predictive capabilities in specific scenarios.

The structure of our paper unfolds as follows: In Chapter 2, we elucidate the fundamental concepts of Lagrangian equations, emphasizing that the majority of physical systems can be represented in the form of generalized Lagrangian equations. Additionally, we present a methodology for formulating these generalized Lagrangian equations. Subsequently, Chapter 3 delves into the construction and comparison of GLNNs against a baseline model. Our numerical experiments, focusing on predictive analyses of selected one-dimensional and two-dimensional physical models while varying network hyperparameters, are detailed in Chapter 4. Finally, Chapter 5 encapsulates our findings, highlighting both the merits of GLNNs and certain limitations observed during training.

2 Preliminaries

In this chapter, we will introduce Lagrangian systems, generalized Lagrangian systems, and their associated mathematical and physical properties.

Definition 2.1. We call

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_k}\right)-\frac{\partial \mathcal{L}}{\partial q_k}=0 \tag{2.1}$$

the Lagrangian description of mechanical system, where $\mathcal{L}$ is the Lagrangian, $q=(q_1,\cdots,q_N)$ the generalized coordinates of the point.

If we denote the kinetic energy of system by T, and U the potential energy, then we can suppose that the total energy is $T+U$, and the Lagrangian is $\mathcal{L}=T-U$. The Lagrangian systems presented in Definition 2.1 are assumed to be unaffected by external forces, thereby exhibiting inherent energy conservation properties. However, in practical scenarios, systems often experience the influence of external forces, such as frictional forces or artificially applied forces. In response to this, a generalized form of the Lagrange equation, similar in structure to that described in Definition 2.1, can be derived as follows [5]:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_k}\right)-\frac{\partial \mathcal{L}}{\partial q_k}=F_k, \tag{2.2}$$

here F_k represent non-conservative forces.

In certain special cases, we can express the non-conservative external force F_k in a specific form, such as $F_k=-a_k\dot{q_k}^n$, for example:

- If the external force corresponds to frictional force, the terms a_k represent the coefficients of friction, and it is assumed that $n=0$.
- If the external force corresponds to viscous force, then the terms a_k represent the viscosity coefficients, and $n\geq 1$. One of the most common examples is that of a damped harmonic motion, where the external force is given by $F=-a\dot{q}$, and the system can be described as $\ddot{q}+a\dot{q}+k^2q=0$.

After providing the definition of a Lagrangian system, we will reference several theorems to illustrate the types of systems that can be represented in the form of (generalized) Lagrange's equations. Furthermore, we will present some expressions of Lagrange's equations for specific systems.

Theorem 2.1 (Fundamental analytic theorem for configuration space formulations)**.** [3] *A necessary and sufficient condition for a local, holonomic, generally nonconservative Newtonian system in the fundamental form*

$$A_{ki}(t,q,\dot{q})\ddot{q}^i+B_k(t,q,\dot{q})=0,\quad k=1,2,\cdots,n, \tag{2.3}$$

which is well defined, of class $\mathcal{C}^2$, and regular in a star-shaped region $\mathbb{R}^{*2n+1}$ of the variables$(t,q,\dot{q})$, to admit an ordered direct analytic representation in terms of the conventional Lagrange's equation in $\mathbb{R}^{*2n+1}$,

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}^k}-\frac{\partial L}{\partial q^k}\equiv A_{ki}\ddot{q}^i+B_k, \tag{2.4}$$

is that the system of equations of motion is self-adjoint in $\mathbb{R}^{*2n+1}$.

Theorem 2.2 (A method to construct Lagrangian)**.** [3] *A Lagrangian for the ordered direct analytic representation of local, holonomic, generally nonconservative Newtonian systems that are well defined of class $\mathcal{C}^2$, regular and self-adjoint in a star-shaped region $\mathbb{R}^{*2n+1}$ of points $(t,q,\dot{q})$,*

$$A_{ki}(t,q,\dot{q})\ddot{q}^i+B_k(t,q,\dot{q})=0,\quad k=1,2,\cdots,n, \tag{2.5}$$

is given by

$$L=K(t,q,\dot{q})+D_k(t,q)\dot{q}^k+C(t,q), \tag{2.6}$$

where functions K,D_k and C are a solution of partial differential equations

$$\begin{aligned}
\frac{\partial^2 K}{\partial\dot{q}^{k_1}\partial\dot{q}^{k_2}}&=A_{k_1k_2}(t,q,\dot{q}),\\
\frac{\partial D_{k_1}}{\partial q^{k_2}}-\frac{\partial D_{k_2}}{\partial q^{k_1}}&=\frac{1}{2}\left(\frac{\partial B_{k_1}}{\partial q^{k_2}}-\frac{\partial B_{k_2}}{\partial q^{k_1}}\right)+\left(\frac{\partial^2 K}{\partial q^{k_1}\partial\dot{q}^{k_2}}-\frac{\partial^2 K}{\partial\dot{q}^{k_1}\partial q^{k_2}}\right)\\
&\equiv Z_{k_1k_2}(t,q),\\
\frac{\partial C}{\partial q^{k_1}}&=\frac{\partial D_{k_1}}{\partial t}-B_{k_1}-\frac{\partial K}{\partial q^{k_1}}+\frac{\partial^2 K}{\partial\dot{q}^{k_1}\partial t}\\
&\quad+\left[\frac{\partial^2 K}{\partial q^{k_1}\partial\dot{q}^{k_2}}+\frac{1}{2}\left(\frac{\partial B_{k_1}}{\partial q^{k_2}}-\frac{\partial B_{k_2}}{\partial q^{k_1}}\right)\right]\dot{q}^{k_2}\\
&\equiv W_k(t,q),
\end{aligned} \tag{2.7}$$

given by

$$\begin{aligned}
K(t,q,\dot{q})&=\dot{q^{k_1}}\int_0^1 d\tau'\left\{\left[\int_0^1 d\tau A_{k_1k_2}(t,q,\tau\dot{q})\right]\right\}(t,q,\tau'\dot{q}),\\
D_{k_1}&=\left[\int_0^1 d\tau\tau Z_{k_1k_2}(t,\tau q)\right]q^{k_2},\\
C&=\left[\int_0^1 d\tau W_k(t,\tau q)\right]q^k.
\end{aligned} \tag{2.8}$$

Theorem 2.1 and Theorem 2.2 provide conditions under which Newtonian systems, a class of dynamic systems commonly encountered in realistic models, can be expressed in the form of Lagrange's equations. These theorems also provide the transformation formulas that hold when these conditions are satisfied.

Thus, we have established that self-adjoint Newtonian systems can be expressed in the form of Lagrange's equations, as shown in Eq. (2.1), with a general procedure for obtaining the expressions. The next question is, if we consider a non-conservative system, can it be represented by the generalized form of Lagrange's equations as shown in Eq. (2.2)? If so, how can we derive the expression of the generalized Lagrange's equations for a non-conservative system?

In order to achieve better energy prediction performance for the system, we employ the physically meaningful quantities of T (kinetic energy) and U (potential energy) to determine the Lagrangian term $\mathcal{L}=T-U$ in the generalized Lagrangian representation of non-conservative systems. Additionally, the non-conservative force can be obtained using equations:

$$F_k=\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{q}_k}\right)-\frac{\partial\mathcal{L}}{\partial q_k}.$$

Thus, by employing the aforementioned approach, we can obtain the generalized Lagrangian representation for non-conservative systems. In other words, we can view the equations of non-conservative systems as generalized Lagrange's equations.

Lemma 2.1. *For most physically motivated non-conservative system, its generalized Lagrangian representation can be obtained using the following approach:*

$$\begin{aligned} &\mathcal{L}=T-U, \\ &F_k=\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_k}\right)-\frac{\partial \mathcal{L}}{\partial q_k}. \end{aligned} \tag{2.9}$$

Here T and U can be calculated by physical formula.

It is worth noting that, for a given system, the expression form of the generalized Lagrange's equation is not unique. Moreover, it has been proven that different generalized Lagrange's equations representing the same system exhibit certain mathematical connections.

Building upon the theoretical foundations described above, we are able to consider most physically meaningful non-conservative system models as determined by generalized Lagrange's equations. Consequently, we can establish generalized Lagrangian neural networks based on this framework.

3 Theory of generalized Lagrangian neural networks

Considering a non-conservative system S established from a physical model, given an observed dataset $T=\{(x_i,x_{i+1})\}$ obtained from system S, our objective is to construct a neural network capable of learning system S by training on the available dataset T. The goal is to enable the neural network to predict the phase flow of system S from arbitrary initial points.

As per Lemma 2.1, we understand that system S can be viewed as governed by the following generalized Lagrange's equations:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_k}\right)-\frac{\partial \mathcal{L}}{\partial q_k}=F_k.$$

In the context of generalized LNNs, we treat both $\mathcal{L}$ and the non-conservative term F as black boxes, i.e., we employ neural networks to learn these two functions. In order to proceed with obtaining predictive results, we apply the chain rule to expand the generalized Lagrange's equations as follows:

$$(\nabla_{\dot{q}}\nabla_{\dot{q}}^{\top}\mathcal{L})\ddot{q}+(\nabla_q\nabla_{\dot{q}}^{\top}\mathcal{L})\dot{q}=\nabla_q\mathcal{L}+F. \tag{3.1}$$

Here, the notation ∇ is consistent with the convention used in LNNs, i.e., $(\nabla_{\dot{q}})_i\equiv\frac{\partial}{\partial \dot{q}_i}$. By performing matrix transformations on the above equation, we obtain:

$$\ddot{q}=(\nabla_{\dot{q}}\nabla_{\dot{q}}^{\top}\mathcal{L})^{-1}[\nabla_q\mathcal{L}-(\nabla_q\nabla_{\dot{q}}^{\top}\mathcal{L})\dot{q}+F]. \tag{3.2}$$

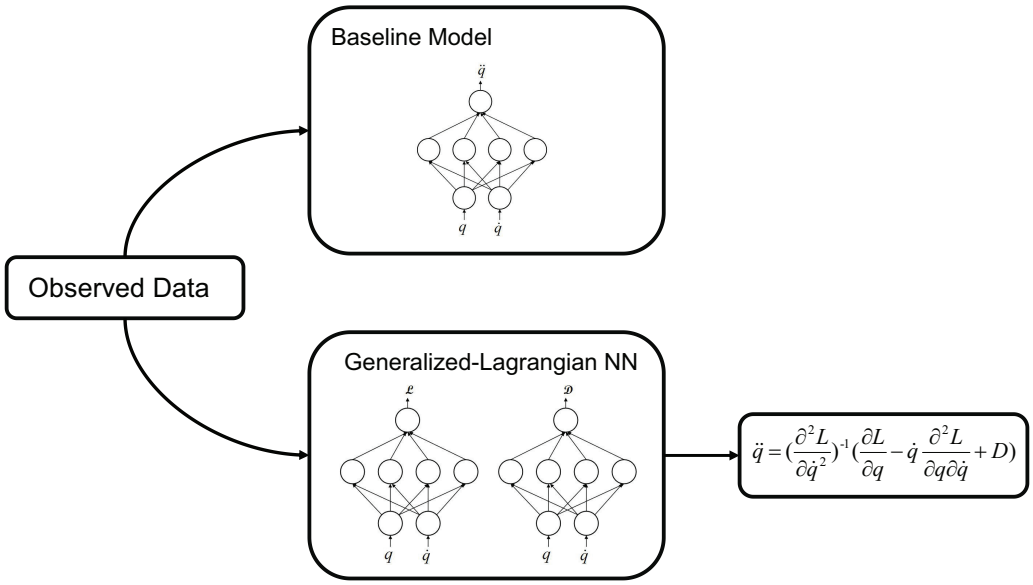

Figure 1: Architecture of models.

Consequently, for the coordinate $x_t = (q_t, \dot{q}_t)$ at time t, we obtain a method for computing $\ddot{q}_t$ from generalized LNNs, allowing us to calculate the phase flow of the system using any numerical scheme.

Loss function

Here, we provide two possible loss functions, allowing for the selection of the most appropriate one for training based on different raw data. When it is possible to obtain or compute the $\ddot{q}$ value at a certain time from raw data, the loss function is chosen as:

$$\mathcal{L} = \frac{1}{|S|} \sum_{x_t \in S} \|\tilde{\ddot{q}} - \ddot{q}\|^2,$$

where x_t is the coordinate at time t, and $\tilde{\ddot{q}}$ is computed using Eq. (3.2). Otherwise, the loss function

$$\mathcal{L} = \frac{1}{|S|} \sum_{x_t \in S} \|\tilde{x_t} - x_t\|^2$$

is used, where $\tilde{x_t}$ represents the prediction of coordinates by our network, S is the sample data.

4 Learning a non-conservative system from data

4.1 Selection of baseline model

The examples selected in our experiments can be regarded as second-order equations, specifically $\ddot{q}=F(\dot{q},q,t)$. Therefore, we consider choosing from the following two baseline models:

- The first choice is Neural ODE, where we treat the system as determined by the equation $\dot{q}=f(q,t)$, and we use a neural network to approximate the function f. The loss function is
$$\mathcal{L}=\frac{1}{|S|}\sum_{x_k\in S}\|\tilde{x_k}-x_k\|^2,$$
here $\widetilde{x_t}$ represents the prediction of coordinates.
- Another choice for the baseline model is based on the form of our example equations, which can all be seen as $\ddot{q}=F(\dot{q},q,t)$. In this case, we consider Baseline 2: using a neural network to approximate the function F, with the same loss function as Neural ODE.

Although both of these baseline models can achieve decent prediction performance, due to the shared characteristics of the selected systems, the second baseline model may exhibit superior performance. Consequently, we choose it as the baseline model for comparison.

4.2 Damped harmonic motion

Perhaps the simplest example of a non-conservative system that can be described by the generalized Lagrange's equation is Damped Harmonic Motion, which is expressed as

$$\ddot{q}+a\dot{q}+k^2q=0,$$

where a represents the coefficient of friction, and k denotes the coefficient of elasticity. In Fig. 2, we illustrate the model of this system and the variation of its coordinate q over time.

Despite several classical expressions of generalized Lagrange's equations being proposed for Damped Harmonic Motion, we will provide an expression for the generalized Lagrange's equation based on Lemma 2.1 introduced earlier, ensuring that $\mathcal{L}$ exactly represent the difference between the system's kinetic and potential energies.

In this example, the kinetic energy is $\frac{1}{2}m\dot{q}^2$, and the potential energy is represented by elastic potential energy $\frac{1}{2}k^2q^2$. Hence, we have

$$\mathcal{L}=\frac{1}{2}m\dot{q}^2-\frac{1}{2}k^2q^2,$$

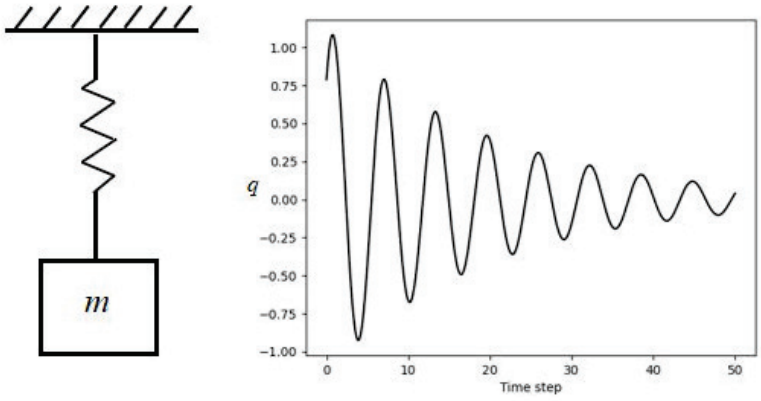

Figure 2: Damped harmonic motion.

Table 1: MSE of two methods (Damped harmonic).

	q	dq	Energy
GLNN	1.94e-4	1.94e-4	2.37e-7
Baseline	1.45e-4	1.15e-4	2.32e-6

leading to the equation

$$F=\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{q}}\right)-\frac{\partial\mathcal{L}}{\partial q}=m\ddot{q}-kq.$$

In the numerical experiment, we choose the initial parameter values as $a=0.02$ and $k=1$. To generate training data, we select 40 trajectories, each starting from a randomly chosen point within the range of $[-1,1]^2$. For each trajectory, we sample 200 points with a step size of $h=0.05$, resulting in a time interval of $T=10$. These 8000 point pairs (x_t,x_{t+1}) are used as training data. Subsequently, we randomly split the training data into a training set and a testing set in a 1:1 ratio.

We use two networks to learn this system: GLNNs and the Baseline model (Baseline 2 model mentioned in Section 4.1). For the Baseline model, we utilize a three-layer fully connected network with a hidden dimension of 200. As for the GLNNs model, we employ two three-layer fully connected networks with a hidden dimension of 200. Both networks are trained using an Adam optimizer with a batch size of 1000 and a learning rate of 0.001. And we presented more training details on Table 2.

In Fig. 3, we present the prediction results for damped harmonic motion. The first row contains the first two plots showing the position and acceleration predictions for q. In the short-term prediction at $T=50$, neither network demonstrates a significant superiority. However, the third plot in the first row displays the prediction for the total energy of the system. In contrast to the position in the phase flow, GLNNs starts to exhibit a noticeable

Table 2: Training details of damped harmonic.

	Activation function	Optimizer	Batch size	Epochs	Learning rate	Initialization
Baseline	tanh	Adam	1000	300	10^{-3}	Xavier
GLNN	tanh	Adam	1000	300	10^{-3}	Xavier

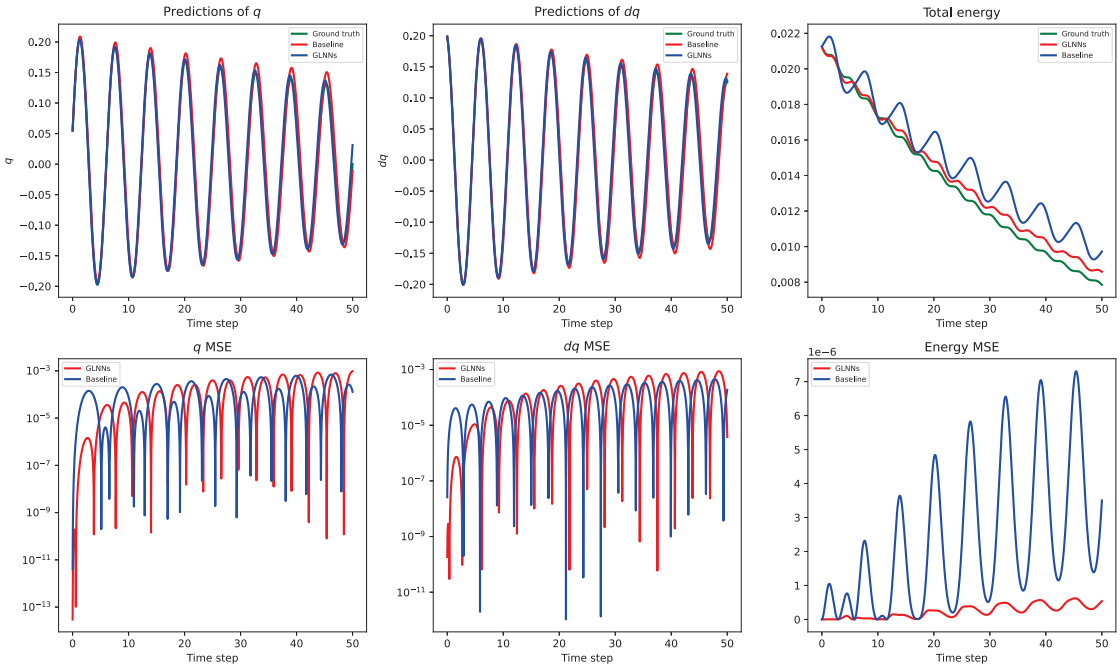

Figure 3: Prediction of damped harmonic motion. The figure displays the phase flow and predicted energy by both baseline model and GLNNs. And GLNNs demonstrates superior prediction accuracy for energy.

advantage in energy prediction. In the second row, we plot the mean squared error (MSE) between the predicted q, dq and energy of both networks and the ground truth. Through the MSE plot, we can clearly observe the higher accuracy of GLNNs in energy prediction.

4.3 Compound double pendulum with friction

Another non-conservative system example we consider is the double pendulum model. Unlike the conventional energy conservative double pendulum, we adopt the compound double pendulum with friction model proposed by Williams [4]. In this model, the two mass points are replaced with irregular-shaped rigid bodies, and frictional forces are considered within the system.

Next, we'll introduce the system. Consider a double pendulum with two identical

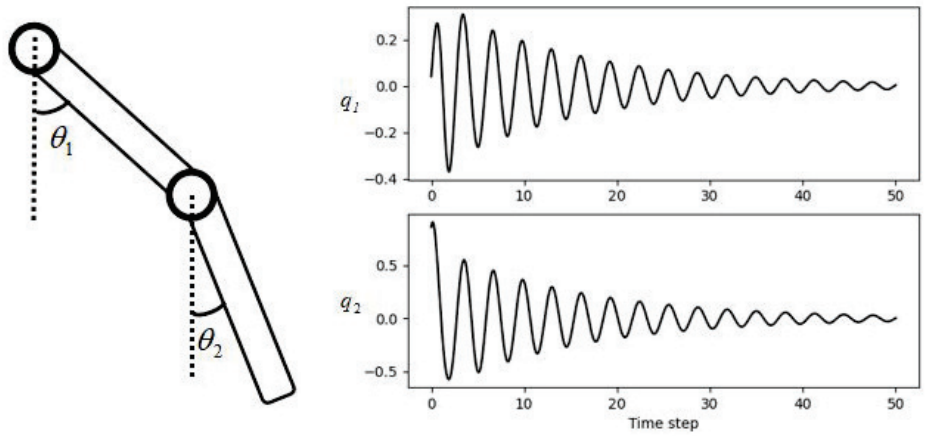

Figure 4: Double pendulum with friction.

Table 3: Parameters of the system.

Quantity	Dimension	Description
m	kg	Mass of first and second pendulum
θ_1	Dimensionless	Angle which first pendulum makes with y axis
θ_2	Dimensionless	Angle which second pendulum makes with y axis
d	m	Distance between the two axles
I	kg m^2	Moment of inertia
c	m	Distance from an axle to the centre of mass of pendulum
γ_1	Dimensionless	Friction coefficient for first axle
γ_2	Dimensionless	Friction coefficient for second axle
g	m s^{-2}	Acceleration due to gravity

rigid bodies of mass m and irregular shape (see Fig. 4). Note that the bodies need not be rods or squares. The upper body rotates around a fixed axle at P in the x-direction, while the lower body rotates about an axle at Q passing through the upper body. The massless axles are positioned at the same points within their respective plate's geometries. The bodies have uniform mass density distributions and equal moments of inertia about their axles. In our generalized setting, we account for friction and other nonconservative forces at the axles.

Here, we omit the specific derivation process of the system's equations of motion. Referring to the results presented in the article by Williams [4], we can obtain the equations

of motion for the system as follows:

$$\begin{aligned}
&-\gamma_1\dot{\theta}_1-\gamma_2(\dot{\theta}_1-\dot{\theta}_2)=2md^2\ddot{\theta}_1+I\ddot{\theta}_1+md^2\ddot{\theta}_2\cos(\theta_1-\theta_2)+md^2\dot{\theta}_2^2\sin(\theta_1-\theta_2)+2mgd\sin\theta_1,\\
&\gamma_2(\dot{\theta}_1-\dot{\theta}_2)=md^2\ddot{\theta}_2+I\ddot{\theta}_2+md^2\ddot{\theta}_1\cos(\theta_1-\theta_2)-md^2\dot{\theta}_1^2\sin(\theta_1-\theta_2)+mgd\sin\theta_2.
\end{aligned}\tag{4.1}$$

By solving the aforementioned equations involving $\ddot{\theta}_1$ and $\ddot{\theta}_2$, we can obtain an expression in the form of

$$\ddot{\theta}_1=F_1(\theta_1,\theta_2,\dot{\theta}_1,\dot{\theta}_2),\quad \ddot{\theta}_2=F_2(\theta_1,\theta_2,\dot{\theta}_1,\dot{\theta}_2).$$

Next, using the approach outlined in Lemma 2.1, we provide an expression for a generalized Lagrange's equation for this system. In this system, the potential energies of the double pendulum are denoted as

$$\begin{aligned}
T_1&=\frac{1}{2}m(\dot{x}_1+\dot{y}_1)^2+\frac{1}{2}J_1\dot{\theta}_1^2,\\
T_2&=\frac{1}{2}m(\dot{x}_2+\dot{y}_2)^2+\frac{1}{2}J_1\dot{\theta}_2^2,
\end{aligned}\tag{4.2}$$

where $J_1=\frac{1}{3}m\left(\frac{c}{2}\right)^2$ and $J_2=\frac{1}{3}m\left(\frac{c}{2}\right)^2$ represent the moments of inertia, and $(x_1,y_1),(x_2,y_2)$ represent the traditional coordinates of the double pendulum. The potential energies of the double pendulum are represented as $U_1=mgy_1$ and $U_2=mgy_2$. Thus, we have

$$\begin{aligned}
\mathcal{L}&=T-U=\frac{1}{2}m(\dot{x}_1+\dot{y}_1)^2+\frac{1}{2}m(\dot{x}_2+\dot{y}_2)^2+\frac{1}{6}m\left(\frac{c}{2}\right)^2(\dot{\theta}_1^2+\dot{\theta}_2^2)-mg(y_1+y_2),\\
F_1&=\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}_1}\right)-\frac{\partial\mathcal{L}}{\partial\theta_1}\\
&=\frac{d}{dt}\left(\frac{m^2c^2}{4}\cos^2\theta_1\dot{\theta}_1+\frac{m^2c^2}{4}\cos\theta_1\sin\theta_1\dot{\theta}_1+\frac{mc^2}{12}\dot{\theta}_1\right)\\
&\quad-\frac{mc}{2}g\sin\theta_1+\frac{mc^2}{4}(\cos\theta_1\sin\theta_1\dot{\theta}_1-\cos\theta_1\dot{\theta}_1^2)\\
&=\frac{m^2c^2}{4}\dot{\theta}_1^2(\cos^2\theta_1-\sin^2\theta_1-2\sin\theta_1\cos\theta_1)\\
&\quad+\frac{mc^2}{4}(\cos\theta_1\sin\theta_1\dot{\theta}_1-\cos\theta_1\dot{\theta}_1^2)-\frac{mc}{2}g\sin\theta_1,\\
F_2&=\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}_2}\right)-\frac{\partial\mathcal{L}}{\partial\theta_2}\\
&=\frac{d}{dt}\left(\frac{m^2c^2}{4}\cos^2\theta_2\dot{\theta}_2+\frac{m^2c^2}{4}\cos\theta_2\sin\theta_2\dot{\theta}_2+\frac{mc^2}{12}\dot{\theta}_2\right)\\
&\quad-\frac{mc}{2}g\sin\theta_2+\frac{mc^2}{4}(\cos\theta_2\sin\theta_2\dot{\theta}_2-\cos\theta_2\dot{\theta}_2^2)\\
&=\frac{m^2c^2}{4}\dot{\theta}_2^2(\cos^2\theta_2-\sin^2\theta_2-2\sin\theta_2\cos\theta_2)\\
&\quad+\frac{mc^2}{4}(\cos\theta_2\sin\theta_2\dot{\theta}_2-\cos\theta_2\dot{\theta}_2^2)-\frac{mc}{2}g\sin\theta_2.
\end{aligned}\tag{4.3}$$

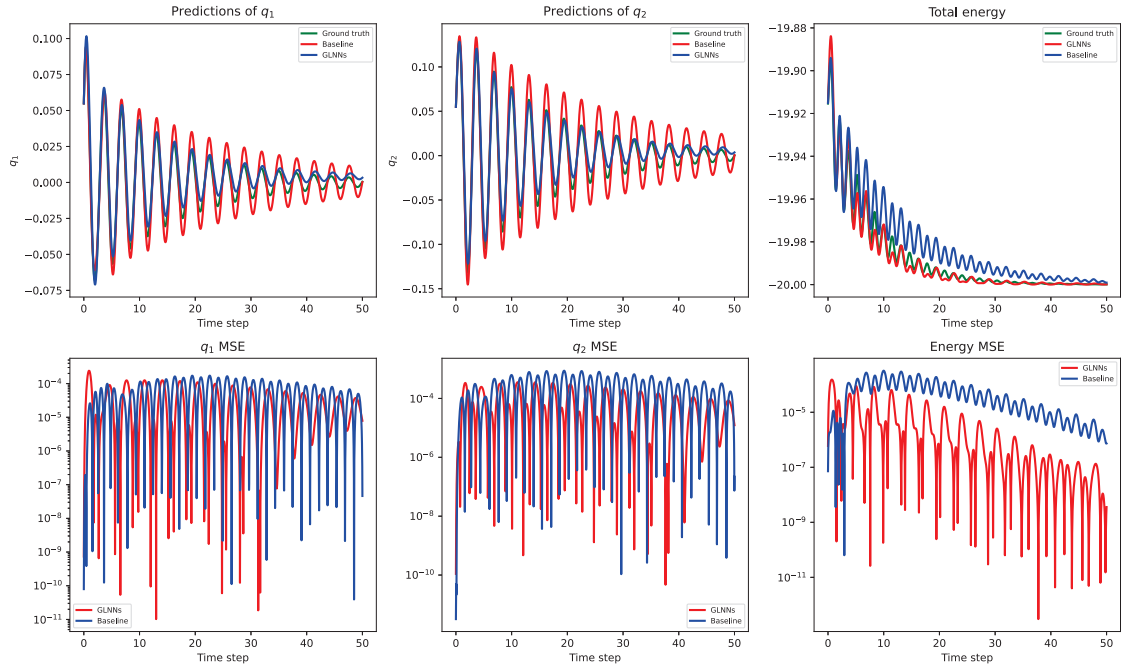

Figure 5: Prediction of double pendulum with friction. This figure illustrates the phase flow, energy, and mean squared error (MSE) of predictions generated by both models. It is evident that GLNNs outperform in both phase flow and energy prediction.

Table 4: MSE of two methods (Compound double pendulum).

	q_1	q_2	Energy
GLNN	3.27e-5	7.96e-5	8.63e-6
Baseline	5.48e-5	2.50e-4	6.11e-5

Next, we proceed with numerical experiments, selecting initial parameters as $m=1$, $c=1$, $d=1$, $g=10$, $\gamma_1=0.5$, and $\gamma_2=0.5$. To generate training data, we choose 20 trajectories, each starting from a randomly selected point $(\theta_1^0,\theta_2^0,\dot{\theta}_1^0,\dot{\theta}_2^0)$ within the range of $[1,-1]^4$. For each trajectory, we sample 500 points with a step size of $h=0.02$, resulting in a time interval of $T=10$. These 10,000 point pairs (x_t,x_{t+1}) are used as training data. Subsequently, we randomly split the training data into a training set and a testing set in a 1:1 ratio.

Also, we test two network models. In the case of the Baseline model, a three-layer fully connected network with a hidden dimension of 200 is utilized. In the GLNNs model, two four-layer fully connected networks with a hidden dimension of 200 are employed. Both networks are trained using an Adam optimizer with a batch size of 1000 and a learning rate of 0.001. Also we presented more training details on Table 5.

In Fig. 5, we present the prediction results for the compound double pendulum with

Table 5: Training details of compound double pendulum.

	Activation function	Optimizer	Batch size	Epochs	Learning rate	Initialization
Baseline	tanh	Adam	1000	300	10^{-3}	Xavier
GLNN	tanh	Adam	1000	300	10^{-3}	Xavier

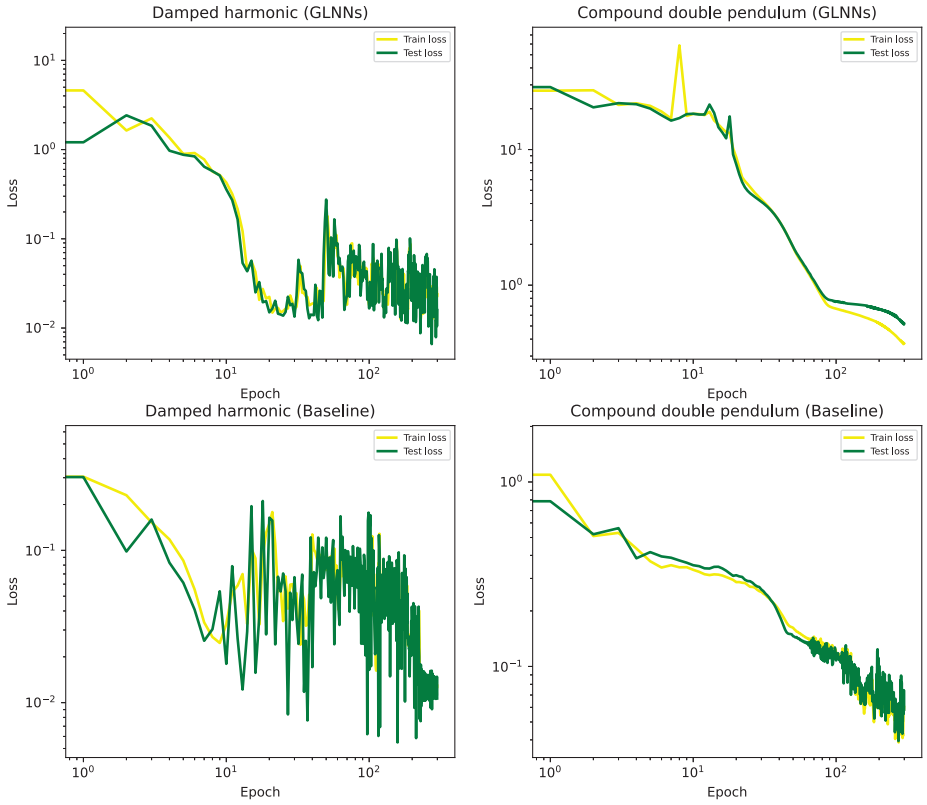

Figure 6: Evolution of Loss.

friction. The first two plots in one row depict the angle predictions for q_1 and q_2 of the pendulum. The third plot shows the prediction for the system's energy. It is evident that in the case of the compound double pendulum, both two models can predict well for q_1 and q_2, but GLNNs performs better in energy prediction.

Remark 4.1. In Fig. 6, we have presented the train loss and test loss for both methods. The first row demonstrates the changes in loss for GLNNs over epochs, while the second row displays the loss for the Baseline method. Upon examining the figure, it is evident that our models were well-trained in both experiments for both methods, with GLNNs converging faster than the Baseline method.

4.4 Investigation into the hyper-parameters

In this subsection, we conduct experiments to investigate influence the hyper-parameters of our networks of predictions.

We main consider change the number of layers and the size of hidden layers, we test layers of 2,3,4,5 and test the size of hidden layers of 50,100,200,400.

In Fig. 7 and Tables 6,7 we present the experimental results. It is evident that for the Damped Harmonic system, a network depth of two layers fails to achieve satisfactory training outcomes. However, increasing the depth to 4 or 5 layers with our available dataset does not lead to optimal training. Consequently, for the Damped Harmonic system, we opt for a three-layered GLNNs architecture, supplemented by a hidden size of 200. For the Compound Double Pendulum with friction system, given the increased dimensionality of the data, optimal training performance is observed at a network depth

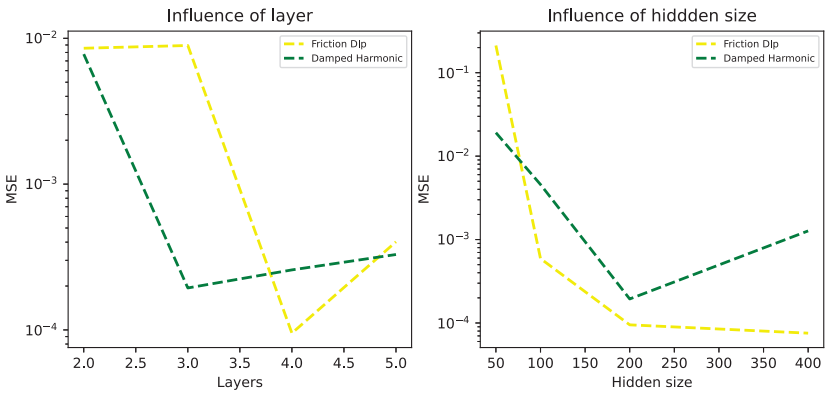

Figure 7: Influence of hyper-parameters. This figure illustrates the impact of network layers and hidden size on our models. It is evident that the optimal hidden size is 200, while the optimal number of layers varies for different systems.

Table 6: MSE of two methods (Compound double pendulum).

	q_1	q_2	Energy
GLNN	3.27e-5	7.96e-5	8.63e-6
Baseline	5.48e-5	2.50e-4	6.11e-5

Table 7: Training details of compound double pendulum.

	Activation function	Optimizer	Batch size	Epochs	Learning rate	Initialization
Baseline	tanh	Adam	1000	300	10^{-3}	Xavier
GLNN	tanh	Adam	1000	300	10^{-3}	Xavier

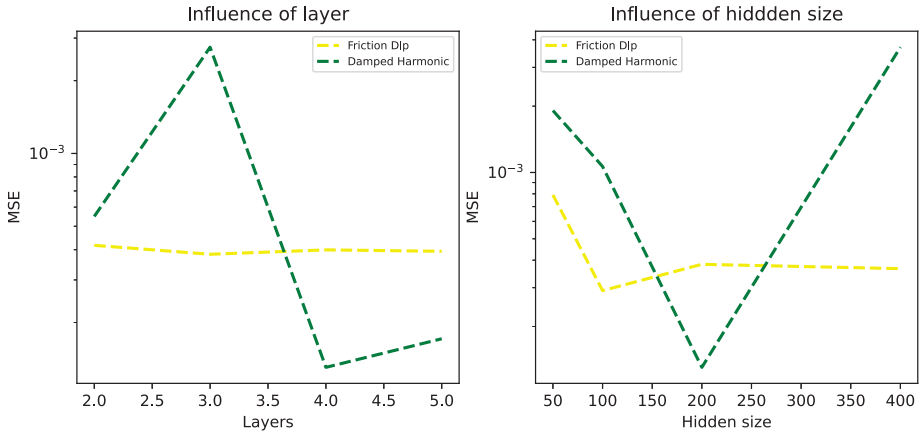

Figure 8: Influence of hyper-parameters of Baseline method.

of four layers. While a hidden size of 400 offers marginal improvements over 200, the associated increase in computational cost leads us to maintain a hidden size of 200.

Furthermore, we have conducted tests to assess the impact of hyper-parameters on the Baseline method, as illustrated in Fig. 8. Our findings reveal that, in the context of the damped harmonic experiment, the optimal hyper-parameters for the Baseline method are 4 layers and a hidden size of 200. Conversely, in the friction dlp example, the best configuration is 3 layers and a hidden size of 100. Based on these results, we have opted to utilize the same hyper-parameters for the Baseline method in both experiments, namely 3 layers and a hidden size of 200.

5 Conclusions

We outline the primary contributions of our research as follows:

- We introduce a methodology to transform physically motivated non-conservative systems into a generalized Lagrangian framework.
- We develop Generalized Neural Networks (GLNNs) tailored for physically motivated non-conservative systems.
- We derive the generalized Lagrangian representation for a compound double pendulum subject to friction.
- Through various experiments, we demonstrate the superior performance of GLNNs in both one-dimensional and two-dimensional scenarios.

Based on our theoretical findings and experimental results, we ascertain the following advantages associated with GLNNs:

- In contrast to LNNs, GLNNs are versatile, applicable to both conservative and non-conservative systems.
- When addressing real-world physical systems with frictional components, Generalized Lagrangian Neural Networks (GLNNs) offer improved accuracy in predicting system energy and phase flow dynamics.
- The inherent preservation of the system's Lagrangian structure in GLNNs enhances their predictive efficacy in specific contexts.

While GLNNs offer certain advantages, it is crucial to acknowledge their inherent limitations. One notable limitation is that, when applied to systems characterized by energy dissipation, GLNNs cannot accurately ensure that the predicted energy diminishes entirely. The forecasting process may exhibit instances of energy rebound, deviating from expected physical characteristics. Another constraint lies in the elevated training complexity associated with GLNNs compared to baseline models. Given the increased network complexity, there is an inevitable escalation in both training time requirements and computational resources.

Acknowledgments

This research is supported by National Natural Science Foundation of China (Grant Nos. 12171466, 12271025 and 92470119).

References

[1] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, S. Ho, Lagrangian neural networks, arXiv preprint arXiv:2003.04630, 2020.
[2] R. M. Santilli, Foundations of Theoretical Mechanics I: The Inverse Problem in Newtonian Mechanics, Springer Science & Business Media, 2013.
[3] R. M. Santilli, Foundations of theoretical mechanics II: Birkhoffian generalization of Hamiltonian mechanics, 1982.
[4] H. Williams, A compound double pendulum with friction, Forces in Mechanics, 10:100164, 2023.
[5] A. M. Bersani, P. Caressa, Lagrangian descriptions of dissipative systems: A review, Mathematics and Mechanics of Solids, 26(6):785–803, 2021.
[6] Y. D. Zhong, N. Leonard, Unsupervised learning of Lagrangian dynamics from images for prediction and control, Advances in Neural Information Processing Systems, 33:10741–10752, 2020.
[7] R. T. Chen, Y. Rubanova, J. Bettencourt, D. K. Duvenaud, Neural ordinary differential equations, Advances in Neural Information Processing Systems, 31, 2018.
[8] Z. Chen, J. Zhang, M. Arjovsky, L. Bottou, Symplectic recurrent neural networks, arXiv preprint arXiv:1909.13334, 2019.

[9] P. Jin, Z. Zhang, A. Zhu, Y. Tang, G. E. Karniadakis, SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems, Neural Networks, 132:166–179, 2020.
[10] Y. Tong, S. Xiong, X. He, G. Pan, B. Zhu, Symplectic neural networks in Taylor series form for Hamiltonian systems, Journal of Computational Physics, 437:110325, 2021.
[11] S. Greydanus, M. Dzamba, J. Yosinski, Hamiltonian neural networks, Advances in Neural Information Processing Systems, 32, 2019.
[12] M. Cranmer, A. S. Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, S. Ho, Discovering symbolic models from deep learning with inductive biases, Advances in Neural Information Processing Systems, 33:17429–17442, 2020.
[13] J. K. Gupta, K. Menda, Z. Manchester, M. J. Kochenderfer, A general framework for structured learning of mechanical systems, arXiv preprint arXiv:1902.08705, 2019.
[14] H. Yu, X. Tian, W. E, Q. Li, OnsagerNet: Learning stable and interpretable dynamics using a generalized Onsager principle, Physical Review Fluids, 6(11):114402, 2021.
[15] M. Lutter, C. Ritter, J. Peters, Deep Lagrangian networks: Using physics as model prior for deep learning, arXiv preprint arXiv:1907.04490, 2019.
[16] A. Sanchez-Gonzalez, V. Bapst, K. Cranmer, P. Battaglia, Hamiltonian graph networks with ODE integrators, arXiv preprint arXiv:1909.12790, 2019.
[17] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Transactions on Neural Networks, 20(1):61–80, 2008.
[18] P. Toth, D. J. Rezende, A. Jaegle, S. Racanière, A. Botev, I. Higgins, Hamiltonian generative networks, arXiv preprint arXiv:1909.13789, 2019.
[19] H. Lyu, N. Sha, S. Qin, M. Yan, Y. Xie, R. Wang, Advances in neural information processing systems, Advances in Neural Information Processing Systems, 32, 2019.
[20] O. Bondesan, A. Lamacraft, Learning symmetries of classical integrable systems, arXiv preprint arXiv:1906.04645, 2019.
[21] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, K. Müller, Machine learning of accurate energy-conserving molecular force fields, Science advances, 3(5):e1603015, 2017.
[22] F. A. Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, J. Z. Kolter, End-to-end differentiable physics for learning and control, Advances in neural information processing systems, 31, 2018.
[23] M. Gastegger, P. Marquetand, High-dimensional neural network potentials for organic reactions and an improved training algorithm, Journal of Chemical Theory and Computation, 11(5):2187–2198, 2015.
[24] S. M. Girvin, K. Yang, Modern Condensed Matter Physics, Cambridge University Press, 2019.
[25] A. N. Gomez, M. Ren, R. Urtasun, R. B. Grosse, The reversible residual network: Backpropagation without storing activations, Advances in neural information processing systems, 30, 2017.
[26] R. Grzeszczuk, D. Terzopoulos, G. Hinton, Neuroanimator: Fast neural network emulation and control of physics-based models, In Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pages 9–20, 1998.
[27] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, J. Davidson, Learning latent dynamics for planning from pixels, in: International Conference On Machine Learning, pages 2555–2565. PMLR, 2019.
[28] R. Iten, T. Metger, H. Wilming, L. D. Rio, R. Renner, Discovering physical concepts with

neural networks, Physical Review Letters, 124(1):010508, 2020.
[29] J. Jacobsen, A. Smeulders, E. Oyallon, i-RevNet: Deep invertible networks, arXiv preprint arXiv:1802.07088, 2018.
[30] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems, 25, 2012.
[31] M. MacKay, P. Vicol, J. Ba, R. B. Grosse, Reversible recurrent neural networks, Advances in Neural Information Processing Systems, 31, 2018.
[32] A. Pukrittayakamee, M. Malshe, M. Hagan, L. Raff, R. Narulkar, S. Bukkapatnum, R. Komanduri, Simultaneous fitting of a potential-energy surface and its corresponding force fields using feedforward neural networks, The Journal of Chemical Physics, 130(13), 2009.
[33] M. Quade, M. Abel, K. Shafi, R. K. Niven, B. R. Noack, Prediction of dynamical systems by symbolic regression, Physical Review E, 94(1):012214, 2016.
[34] A. Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network, Physica D: Nonlinear Phenomena, 404:132306, 2020.
[35] T. F. Chan, M. K. Ngmichael, Galerkin projection methods for solving multiple linear systems, SIAM Journal on Scientific Computing, 21(3):836–850, 1999.
[36] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, M. W. Mahoney, Characterizing possible failure modes in physics-informed neural networks, Advances in Neural Information Processing Systems, 34:26548–26560, 2021.
[37] B. Liu, Z. Wan, X. Lu, L. Liu, A physics-informed deep learning closure for Lagrangian velocity gradient evolution, Physics of Fluids, 35(11), 2023.
[38] D. Lavado, C. Soares, A. Micheletti, Achieving constraints in neural networks: A stochastic augmented Lagrangian approach, arXiv preprint arXiv:2310.16647, 2023.
[39] M. Maggini, M. Tiezzi, M. Gori, A Lagrangian framework for learning in graph neural networks, in: Artificial Intelligence in the Age of Neural Networks and Brain Computing, pages 343–365. Elsevier, 2024.