Journal of Computational Science 90 (2025) 102633

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Physics-informed neural networks for compliant robotic manipulators dynamic modeling

Zhiming Li [a,b], Shuangshuang Wu [a,b], Wenbai Chen [a,*], Fuchun Sun [b,*]

[a] College of Automation, Beijing Information Science and Technology University, Beijing, 100192, China
[b] Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China

ARTICLE INFO

Keywords:
Physics-informed neural networks
Rigid-body dynamics
Compliant robot
Inverse dynamic modeling

ABSTRACT

Deep learning is widely used in robotics, yet often overlooks key physical principles in dynamic modeling, leading to a lack of interpretability and generalization. To address this issue, recent innovations have introduced physics-informed neural networks (PINNs), which integrate fundamental physics into deep learning and offer significant advantages in modeling rigid-body dynamics. This study focuses on the application of PINNs to model compliant robotic manipulators. This requires extending PINNs to handle complex compliant dynamics. We propose an augmented PINN model capable of comprehensively learning manipulator dynamics, including compliant components. The model is tested on dynamic modeling of two physical compliant manipulators and a simulated manipulator. The results highlight its exceptional precision and generalization across a wide range of robotic systems, from purely rigid to compliant structures.

1. Introduction

Accurate inverse dynamics, which links joint motion to applied torque, is essential for model-based control of robotic manipulators [1]. Due to joint, link, and actuator interactions, these systems exhibit strong nonlinearity and coupling, making modeling challenging [2]. Modern collaborative robots often use compliant manipulators with elastic actuators to ensure safe interaction by absorbing contact forces [3]. While this improves safety, the introduced kinematic and elastomechanical coupling further complicates the dynamic modeling process [4,5].

Traditional inverse dynamics modeling for robotic manipulators relies on known physical parameters and manual derivation using Newtonian, Lagrangian, or Hamiltonian mechanics. Unluckily, obtaining precise real numerical values for the physical parameters is not a straightforward task, and numerous robot manufacturers opt to conceal this information for logical safety and copyright reasons [6]. Robot parameters can be obtained by testing individual components, but this requires disassembly, which is both risky and time-consuming and may alter system behavior [7]. Computer-Aided Design (CAD) models provide an alternative, but they often neglect factors such as friction, flexibility, and environmental variations.

Another approach to obtaining robot parameters is physical parameter identification, which uses algorithms such as least squares to estimate the dynamic parameters of robot joints and construct inverse dynamics models [8]. This method enhances accuracy by statistically matching real sampled data from the robot with the mathematical model formulation [9]. However, due to noise and underlying assumptions, this method cannot ensure physically consistent parameters and may produce unrealistic results, such as negative mass values or inertia matrices that are not positive definite [10]. To overcome this, Gautier et al. incorporated these constraints into the identification process using singular value decomposition and Cholesky factorization, ensuring that the resulting mass matrix is symmetric and positive definite [11].

Identification-based modeling methods typically construct model structures under rigid body assumptions using classical physical formulations such as Newtonian or Lagrangian mechanics. While effective for rigid systems, these methods struggle to handle uncertainties and nonlinearities. Due to the linearization requirements, robot dynamics and friction models are often simplified into linear systems [12]. Although flexible joint identification is a viable approach for modeling compliant robots, it faces challenges in accurately capturing parameters and the nonlinear, time-varying dynamics [13]. Simplified analytical models often overlook joint stiffness and elasticity or treat them as linear with uncertain coefficients [14]. Nonlinear effects such as clearance and frictional torque are also frequently ignored [15,16], further reducing accuracy due to their sensitivity to environmental conditions and maintenance. As a result, for low-cost compliant manipulators with elastic actuators, traditional identification methods based on rigid body assumptions struggle to capture their flexible dynamic behavior,

* Corresponding authors.
E-mail addresses: Lizm@bistu.edu.cn (Z. Li), shuangsw@bistu.edu.cn (S. Wu), chenwb@bistu.edu.cn (W. Chen), fcsun@mail.tsinghua.edu.cn (F. Sun).

https://doi.org/10.1016/j.jocs.2025.102633
Received 18 September 2024; Received in revised form 18 May 2025; Accepted 19 May 2025
Available online 5 June 2025

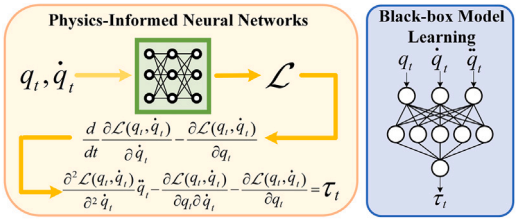

Fig. 1. The diagram illustrates the application of physics-informed and black-box models in inverse dynamics modeling.

limiting their applicability [17]. Furthermore, existing research on robot joint friction rarely models friction concurrently with overall dynamics [18]. Data-driven methods are increasingly applied to enhance the accuracy of inverse dynamics models. Specifically, Deep neural networks excel at learning and capturing complex nonlinear behaviors, improving precision in inverse dynamics modeling [17]. However, the black-box nature of these models disregards prior knowledge from first principles, making them susceptible to overfitting without uncovering the underlying structure [19]. This is particularly problematic for robots, as overfitting to incorrect data can lead to unpredictable behavior, damaging the physical system and complicating model application in controllers [20]. Data-driven methods have been used to model compliant manipulators [21,22], addressing complexities like friction and flexibility with experimental data. However, these methods lack interpretability and theoretical grounding, raising concerns about generalization and requiring large datasets [23].

Recent research has increasingly focused on integrating physical information with deep neural network models [24]. Physics-Informed Neural Networks(PINNs) combines traditional knowledge with deep networks, utilizing deep learning to uncover hidden structures while retaining knowledge-based advantages [20]. Lagrangian and Hamiltonian mechanics [25] provide system descriptions that can be embedded into neural networks. In robotics, physics-informed neural networks like Deep Lagrangian Networks (DeLaN) [26] and Hamiltonian neural networks (HNN) [27] integrate these mechanics with deep learning, improving physical plausibility, interpretability, and generalization [28]. The differences between DeLaN and black-box models are shown in Fig. 1.

Lagrangian-based methods use neural network models to represent physical systems based on the Euler–Lagrange equations of motion utilizing generalized coordinates q, their velocity $\dot{q}$ and a Lagrangian function $\mathcal{L}(q, \dot{q})$, which represents the difference between kinetic and potential energies. These energy terms are modeled by neural networks, either separately [26,29] or together [30]. Hamiltonian-based methods use a Hamiltonian formulation to describe system dynamics, utilizing generalized coordinates q, generalized momenta p and a Hamiltonian function $\mathcal{H}(q, p)$ to represent the total energy of the system. Similarly, the energy terms are modeled by neural networks, either separately [20,31] or together [27]. The potential of DeLaNs and HNNs in learning the dynamics of basic physical systems has been shown in several studies [2,20,23,31–35]. However, their application to modeling complex robotic structures, particularly with real-world data, is still in the early stages. DeLaNs and HNNs, based on rigid body priors, often reduce model accuracy compared to black-box methods, as they struggle to fully capture phenomena such as friction, hysteresis, and contact [2,23,33]. While these models are well-suited for rigid body dynamics, they perform suboptimally when applied to compliant manipulators with non-rigid dynamics, limiting its application scope.

To address the limitations of PINNs in modeling friction due to conservative dynamics, most studies focus on incorporating actuator-induced friction into model learning for robotic manipulators [29,32,34,36], achieving simultaneous modeling of friction during overall dynamics modeling. However, simply including primary friction is insufficient for modeling compliant manipulators. PINNs constrained by rigid-body dynamics priors, often fail to capture uncertainties like nonlinear friction and flexibility, limiting their effectiveness for compliant systems. Our previous work introduced DeLaN-FFNN, an augmented Deep Lagrangian Network that combines DeLaN with a Feed-Forward Neural Network (FF-NN) [2]. This method expands DeLaN's capabilities to handle uncertainties beyond rigid body dynamics, effectively learning the dynamics of both rigid and compliant manipulators while preserving interpretability. Considering all the implications above, we can enumerate the main contributions of this work as follows:

(1) An augmented deep Lagrangian network model with a physically plausible optimization method is proposed to account for the complex nonlinear characteristics, enabling the accurate construction of the inverse dynamics model for compliant manipulators.
(2) A comprehensive experiment on compliant manipulators is conducted, including the Baxter Robot and Barrett WAM, showing that DeLaN underperforms compared to black-box models, while DeLaN-FFNN excels in accuracy and generalization for non-rigid-body dynamics.
(3) A simulation rigid-body dynamics experiment is conducted, further demonstrating that DeLaNs and HNNs are capable of recovering system dynamics from ideal data, showcasing strong extrapolation ability and data efficiency. DeLaN-FFNN also excels in rigid-body dynamics modeling.

The paper is organized as follows. Section 2.1 introduces robot inverse dynamics, discussing both rigid and non-rigid body dynamics and their differences. Section 2.2 reviews previous methods for learning rigid body dynamics using Lagrangian and Hamiltonian mechanics. Section 2.3 covers various DeLaN and HNN architectures, followed by Section 2.4, which describes integrating a friction model into DeLaN. Section 3 explains the parameterization process for the DeLaN-FFNN model and the optimization methods used. Lastly, Section 4 presents experimental results comparing DeLaNs, HNNs, black-box methods, DeLaN-Friction, and DeLaN-FFNN for both non-rigid and rigid body dynamics.

2. Preliminaries

This section aims to present the general mathematical formulation of robot inverse dynamics. Additionally, it offers a concise overview of the theoretical foundations of Lagrangian and Hamiltonian mechanics, as well as DeLaNs and HNNs.

2.1. Robot inverse dynamics

The objective of inverse dynamics modeling is to identify the function f^{-1}, which maps system changes to control inputs, i.e.

$$f^{-1}(q, \dot{q}, \ddot{q}) = \tau \tag{1}$$

where q, $\dot{q}$, and $\ddot{q}$ represent the vectors of joint positions, velocities, and accelerations, respectively. For a robot with n degrees of freedom (dof), these vectors have the dimension R^n. $\tau \in R^n$ represents the unknown joint torques to be learned. Based on these definitions, the general inverse dynamics model for a robot is given by

$$H(q)\ddot{q} + c(q, \dot{q}) + g(q) + \varepsilon(q, \dot{q}, \ddot{q}) = \tau \tag{2}$$

where $H(q) \in R^{n \times n}$ is the symmetric, positive definite mass matrix, $c(q, \dot{q}) \in R^n$ represents the matrix for centrifugal and Coriolis forces, $g(q) \in R^n$ denotes the gravity vector, and $\varepsilon(q, \dot{q}, \ddot{q}) \in R^n$ accounts for the torque/force effects of unmodeled elements, such as viscous friction or the nonlinear effects of serial elastic actuator (SEA) springs [37].

Various methods can be used to derive an inverse dynamics model from the equation of motion, neglecting $\varepsilon(q, \dot{q}, \ddot{q})$. The most common approaches for modeling robot dynamics are based on Lagrangian

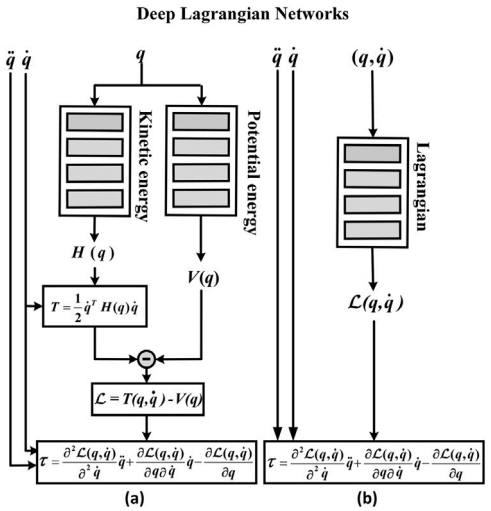

Fig. 2. The inverse model with DeLaNs predicts the system's Lagrangian $\mathcal{L}$ and calculates changes via the Euler–Lagrange equations. In (a), the structured approach uses two networks for kinetic (T) and potential (V) energies to compute $\mathcal{L} = T - V$. In (b), the black-box method directly predicts $\mathcal{L}$.

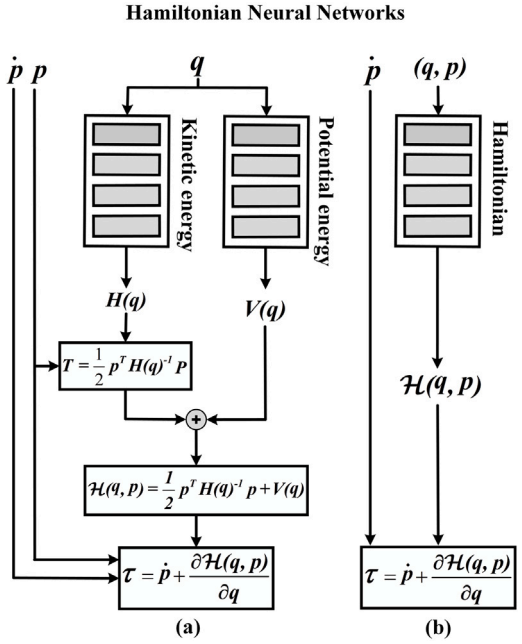

Fig. 3. HNNs predict the Hamiltonian $\mathcal{H}$ and compute changes in position and momentum using Hamilton's equations. In variant (a), the structured approach employs two networks for kinetic and potential energies, while (b) is the black-box approach, directly predicting $\mathcal{H}$ with a single network.

and Hamiltonian mechanics, both derived from rigid body dynamics principles. The robot's rigid body dynamics (τ_{RBD}) can be expressed in a specific mathematical form.

$$H(q)\ddot{q} + c(q,\dot{q}) + g(q) = \tau_{\text{RBD}} \tag{3}$$

As both Lagrangian and Hamiltonian mechanics are based on rigid body dynamics, they often overlook factors like $\varepsilon(q,\dot{q},\ddot{q})$ in robot inverse dynamics, failing to account for complex nonlinear phenomena such as friction, hysteresis, contact, and forces arising from flexibility and uncertainties [38]. Consequently, these methods may produce rough approximations in inverse dynamics models, resulting in inaccurate torque predictions and imprecise robot movements [39].

2.2. Lagrangian and Hamiltonian dynamics

The dynamics of rigid body robots can be described using either Lagrangian or Hamiltonian mechanics. In Lagrangian mechanics, the state is represented by generalized coordinates q and velocities $\dot{q}$. The system's behavior follows the Euler–Lagrange equation: $\frac{d}{dt}\left(\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}(q,\dot{q})}{\partial q} = \tau$, where $\mathcal{L}(q,\dot{q}) = T(q,\dot{q},) - V(q)$, with potential energy $V(q) \in R$ and kinetic energy $T = \frac{1}{2}\dot{q}^T H(q)\dot{q}$.

In Hamiltonian mechanics, momenta $p \in R^n$ replace velocities, where $\dot{q} = H^{-1}(q)p$. The system follows Hamiltonian equations: $\dot{q} = \frac{\partial \mathcal{H}(q,p)}{\partial p}$, $\dot{p} = -\frac{\partial \mathcal{H}(q,p)}{\partial q} + \tau$, with $\mathcal{H}(q,p) = T(q,p) + V(q)$ representing total energy. Here, kinetic energy is defined as $T(q,p) = \frac{1}{2}p^T H^{-1}(q)p$.

Substituting the kinetic and potential energy into the Euler–Lagrange equation for $\mathcal{L}$ and into the Hamiltonian equation for $\mathcal{H}$ both yield a second-order ordinary differential equation (ODE) described by [20]:

$$H(q)\ddot{q} + \underbrace{\dot{H}(q)\dot{q} - \frac{1}{2}\left(\dot{q}^T \frac{\partial H(q)}{\partial q}\dot{q}\right)^T}_{=c(q,\dot{q})} + \underbrace{\frac{\partial V(q)}{\partial q}}_{=g(q)} = \tau \tag{4}$$

This expression can be simplified to yield Eq. (3). The Lagrangian and Hamiltonian methods are commonly used for inverse dynamics analysis, enabling the formulation of simplified inverse dynamics models when uncertain forces $\varepsilon(q,\dot{q},\ddot{q})$ are ignored.

2.3. DeLaNs and HNNs

DeLaNs utilize the principle of least action to learn a Lagrangian function $\mathcal{L}(q,\dot{q})$ from trajectory data, generating dynamics through the Euler–Lagrange framework [26], as illustrated in Fig. 2. On the other hand, HNNs are designed to learn the Hamiltonian function $\mathcal{H}(q,p)$, which then produces dynamics via Hamilton's equations [27], shown in Fig. 3. The loss functions for both DeLaNs and HNNs are calculated as the mean squared error (MSE) between the actual torques τ_R and the predicted torques $\hat{\tau}$.

$$Loss = MSE(\tau_R, \hat{\tau}) \tag{5}$$

Both the Lagrangian $\mathcal{L}$ and Hamiltonian $\mathcal{H}$ can be parameterized using two networks: one for the mass matrix (or its inverse) and the other for potential energy, known as the structured Lagrangian/Hamiltonian models (Figs. 2(a) and 3(a)). Alternatively, a single network can be used, forming the black-box Lagrangian/Hamiltonian models (Figs. 2(b) and 3(b)). It is worth noting that in subsequent models, friction or uncertain forces are introduced, and the model used is the structured DeLaN.

2.4. Introducing friction to model learning

DeLaN is unable to directly model friction due to its conservative dynamics. Most studies on DeLaN focus on integrating major friction components in robotic manipulators [29,32,34,36], with [36] having applied this to the Baxter Robot's compliant manipulator. These approaches assume that motor friction depends only on the velocity of the ith joint $\dot{q}_i$, independent of other joints. Depending on the model's complexity, static, viscous, or Stribeck friction is used as a prior, and the combined effects are expressed as:

$$\hat{\tau}_{f_i} = -\left(\tau_{C_v} + \tau_{C_s}\exp\left(-\frac{\dot{q}_i^2}{v}\right)\right)\text{sign}(\dot{q}_i) - d\dot{q}_i \tag{6}$$

Where τ_{C_v} represents static friction, d is viscous friction, and τ_{C_s} and v are the Stribeck friction coefficients. The friction coefficients are abbreviated as $\varphi = \left\{\tau_{C_v}, \tau_{C_s}, v, d\right\}$. The friction force $\hat{\tau}_{f_i}$ is incorporated into the Lagrangian model [29], with φ learned as network weights. We name the method that integrates friction into DeLaN learning as DeLaN-Friction, which is used as the subsequent experimental baseline.

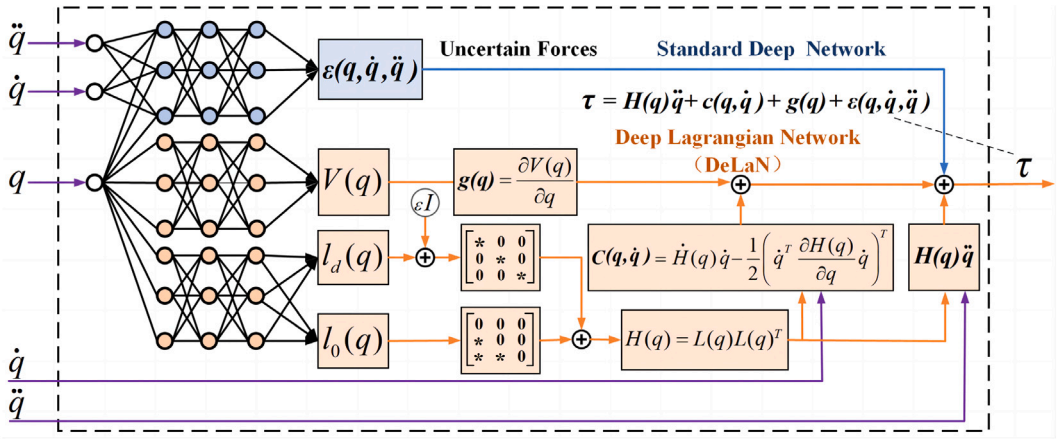

Fig. 4. DeLaN-FFNN predicts dynamics by dividing forces into those following rigid body dynamics and those resulting from flexibility, nonlinear friction, and other uncertainties. In the model, the orange section represents DeLaN, and the blue section denotes the standard deep network. This model incorporates all force components in the robot's inverse dynamics.

3. Proposed algorithms

3.1. Compliant robot dynamic

For compliant robotic manipulators, which involve non-rigid body dynamics as shown in Eq. (2), it is crucial not only to model the dynamics of rigid body components Eq. (3), but also to capture the additional complex dynamics $\varepsilon(q, \dot{q}, \ddot{q})$ inherent in elastic joints, such as friction and elasticity. These dynamics are essential for linking applied torques to motion [5]. As a result, models like DeLaN, which are designed for rigid robots, are often inadequate for elastic manipulators. Thus, elastic robots require more advanced modeling techniques to address their complex dynamics [40].

However, identifying compliant robot parameters related to $\varepsilon(q, \dot{q}, \ddot{q})$ is mathematically challenging [4]. The mathematical intractability motivates the use of data-driven methods for compliant dynamic modeling [17]. As mentioned in Section 2.4, most of the existing improvements based on the DeLaN model focus primarily on incorporating the main joint friction into the model learning, enabling the improved DeLaN-Friction to model friction concurrently with overall dynamics. However, for the complex compliant dynamics $\varepsilon(q, \dot{q}, \ddot{q})$, friction is only a part of it. Considering only frictional dynamics outside of rigid body dynamics still cannot fully capture the complex dynamic phenomena of compliant robotic manipulators. These factors drive us to design a more effective physics-informed neural network structure to improve the modeling performance for compliant robotic manipulators.

3.2. Introducing uncertainty force to model learning

To overcome the limitations of DeLaN, which only accounts for rigid-body dynamics τ_{RBD} and lacks the capacity to model forces beyond this scope, we incorporate the learning of uncertain forces induced by complex nonlinear dynamic phenomena $\varepsilon(q, \dot{q}, \ddot{q})$ into the model.

3.2.1. Network structure

By combining DeLaN with a standard deep network, our model incorporates all force components from Eq. (2), including inertial forces $H(q)\ddot{q}$, Coriolis and Centrifugal forces $c(q, \dot{q})$, gravitational forces $g(q)$, and uncertain forces $\varepsilon(q, \dot{q}, \ddot{q})$. This augmented deep Lagrangian network, called DeLaN-FFNN, integrates physics-informed and standard deep networks, providing a structure that better aligns with the dynamics of compliant robotic manipulators.

The structure of DeLaN-FFNN is shown in Fig. 4, with three fully connected networks used to parameterize the parameters $\hat{H}(q;\theta)$, $\hat{V}(q;\phi)$, and $\hat{\varepsilon}(q, \dot{q}, \ddot{q};\psi)$, where θ, ϕ, and ψ represent the respective network parameters. The orange section depicts the DeLaN structure, while the blue section represents the FF-NN. FF-NN is a standard neural network, used to model the uncertainty term $\varepsilon(q, \dot{q}, \ddot{q})$ as $\hat{\varepsilon}(q, \dot{q}, \ddot{q};\psi)$. Substituting

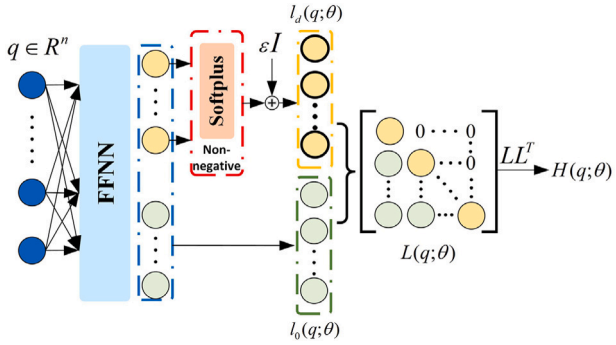

Fig. 5. Diagram of the mass matrix including a feed-forward neural network, a non-negative shift for diagonal entries, and the Cholesky decomposition.

$$\hat{\tau} = \hat{f}^{-1}(q, \dot{q}, \ddot{q}; \theta, \phi, \psi) \tag{7}$$

the approximated robotic manipulator' inverse model can be described by

$$\hat{f}^{-1}(q, \dot{q}, \ddot{q}; \theta, \phi, \psi) = \hat{H}(q;\theta)\ddot{q} + \underbrace{\frac{\partial \hat{V}(q, \phi)}{\partial q}}_{\hat{g}(q;\phi)} + \hat{\varepsilon}(q, \dot{q}, \ddot{q}; \psi) + \underbrace{\dot{\hat{H}}(q;\theta)\dot{q} - \frac{1}{2}\left(\dot{q}^T \frac{\partial \hat{H}(q;\theta)}{\partial q}\dot{q}\right)}_{\hat{c}(q,\dot{q};\theta)} \tag{8}$$

3.2.2. Positive-definite mass matrix

To ensure physically plausible kinetic energy $T = \frac{1}{2}\dot{q}^T H(q)\dot{q}$, the mass matrix $H(q)$ must be positive definite, guaranteeing positive kinetic energy for all non-zero velocities. This is achieved by applying the Cholesky decomposition [11], which decomposes a symmetric positive definite matrix into the product of a lower triangular matrix and its transpose. Therefore, the predicted mass matrix $\hat{H}(q;\theta)$ of the physical system can be expressed in the form of a Cholesky decomposition plus a positive regularization offset ε, which guarantees the positive definiteness of $\hat{H}(q;\theta)$, i.e.

$$\hat{H}(q;\theta) = \hat{L}(q;\theta)\hat{L}(q;\theta)^T + \varepsilon I \tag{9}$$

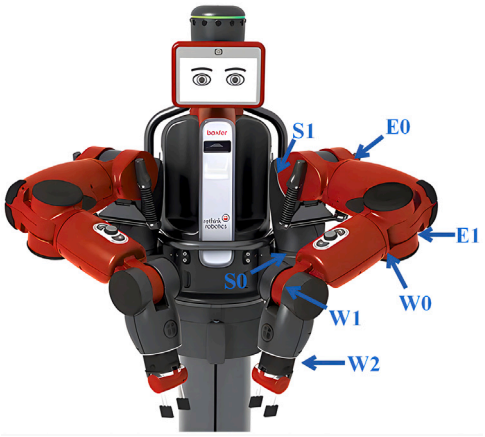

Fig. 6. Baxter manipulator. Joints S0 and S1 form the shoulder, joints E0 and E1 form the elbow, and joints W0, W1, and W2 constitute the wrist.

where $\hat{L}(q;\theta)$ is a lower triangular matrix with positive diagonal elements, and $\hat{L}(q;\theta)$ consists of $\hat{l}_d(q;\theta)$ for predicting diagonal elements and $\hat{l}_0(q;\theta)$ for predicting other elements. The diagonal elements $\hat{l}_d(q;\theta)$ are predicted using activation functions such as Softplus or ReLU to ensure non-negativity, with a small scalar εI added to guarantee positive definiteness, I is identity matrix and ε is a hyperparameter that should be selected to be small-enough but strictly positive. For further details, see Fig. 5.

3.2.3. Optimization method

For Eq. (2), most rigid or near-rigid robots, typically equipped with high-ratio gearboxes, often assume $\varepsilon(q,\dot{q},\ddot{q}) = 0$ in dynamic modeling, as the torque contributions from τ_{RBD} are much larger than those from $\varepsilon(q,\dot{q},\ddot{q})$ [2,5]. If DeLaN-FFNN uses a simple mean square error loss function as shown in Eq. (5), it may, in some cases, cause the potential and kinetic energy to be neglected, and result in the uncertain force network dominating the predicted dynamics. This is clearly inconsistent with real-world scenarios, which drives us to design a more physically plausible optimization method that better aligns with actual conditions.

To prevent potential and kinetic energy from being neglected, and to avoid the uncertain force network dominating the predicted dynamics, additional penalty terms are introduced. This ensures that in rigid or near-rigid body dynamics, DeLaN's output torque is maximized while minimizing $\varepsilon(q,\dot{q},\ddot{q})$. However, for compliant manipulators, $\varepsilon(q,\dot{q},\ddot{q})$ plays a crucial role [5]. Therefore, to make DeLaN-FFNN applicable to most robotic manipulators, whether a purely rigid manipulator or a non-rigid manipulator, such as one with a compliant structure, our optimization objective is defined as follows:

$$(\theta^*,\phi^*,\psi^*) = \arg\min_{\theta,\phi,\psi}\left(\left\|\hat{f}^{-1}(q,\dot{q},\ddot{q};\theta,\phi,\psi) - \tau_R\right\|^2_{W_{\tau_R}} + \lambda\hat{\varepsilon}(q,\dot{q},\ddot{q};\psi)^2\right) \tag{10}$$

In this model, τ_R represents the actual torque measured from the physical manipulator, which is compared with the network's output torque $\hat{\tau}$ to compute the loss. To avoid inaccuracies in the torque modeling of the upper joints, caused by significant differences in the magnitude of the torques among different joints. The Mahalanobis norm $\|\cdot\|_W$ is used, with W_{τ_R} representing the diagonal covariance matrix of generalized forces. Normalizing the loss with the covariance matrix helps account for varying residual magnitudes across different joints. The value of λ must be positive to ensure that the rigid-body dynamics component of DeLaN-FFNN's output dominates, while the torque output from the uncertain force network remains secondary. However, the value of λ needs to be adjusted based on the compliance of the robotic manipulator to ensure that DeLaN-FFNN is applicable to robotic manipulators with varying degrees of non-rigidity.

4. Experiment

In these experiments, DeLaN-FFNN is used to learn the non-linear dynamics of both real and simulated systems, compared with baselines like DeLaNs, HNNs, DeLaN-Friction, and black-box methods. For real systems, the models are applied to compliant manipulators such as the Baxter Robot and the Barrett WAM, testing PINNs where rigid-body priors do not apply. DeLaN-FFNN's performance is evaluated against these baselines to address DeLaN's limitations in non-rigid dynamics. In simulations, a 3-dof manipulator is used to test the advantages of PINNs in modeling rigid body dynamics.

4.1. Experiment setup

In the experiments, DeLaNs and HNNs that use a single network for the Lagrangian or Hamiltonian are termed black-box DeLaN/HNN. When two networks represent the mass matrix and potential energy, this is called structured DeLaN/HNN. Structured/black-box DeLaN, structured/black-box HNN, black-box feedforward network (FF-NN), DeLaN-Friction, and DeLaN-FFNN are evaluated for inverse dynamics modeling in different systems. Structured/black-box HNN is assessed only in simulations, as real systems lack momentum data p.

In the experiments, mean square error (MSE), normalized mean square error (nMSE), and the coefficient of determination (R^2) are used to evaluate the accuracy and predictive performance of the models. These metrics offer a comprehensive evaluation of model performance. The R^2 is defined as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\tau_i - \hat{\tau}_i)^2}{\sum_{i=1}^{N}(\tau_i - \bar{\tau})^2} \tag{11}$$

where, τ_i and $\hat{\tau}_i$ represent the actual and predicted torque values, respectively, while $\bar{\tau}$ is the dataset's mean torque value. The R^2 value, ranging from 0 to 1, measures how well a model explains the variance in the data, with higher values indicating better predictions and 1 representing perfect accuracydof [23].

4.1.1. Plants

In the experiments, the model learning techniques are applied to the Baxter Robot, the Barrett WAM, and a simulated three-link manipulator.

Baxter Robot. A collaborative and compliant robot, features two arms, each with 7-dof, and is equipped with SEAs, is shown in Fig. 6. These SEAs, unlike rigid actuators, include a spring between the motor gear and the actuator end, enabling them to absorb contact forces to a degree and provide passive compliance. Additionally, Baxter has a passive spring in its second joint, further augmenting the dynamic complexity of the robot manipulator [5].

Barrett WAM. A robot with direct cable drives, generates high torques leading to fast and agile movements but results in complex dynamics. The variability in stiffness and lengths of the cables makes it challenging for rigid-body dynamics models to accurately capture these dynamics [26,41].

Three-link Manipulator. The three-link manipulator, with continuous revolute joints, operates in the vertical x–z plane under gravity, enabling it to reach any position within its workspace. Notably, complex friction effects are absent in its motion. The simulation is conducted using the Pybullet physics engine.

4.1.2. Neural network training details

In each experiment, the dynamics models are trained on a fixed dataset until convergence and evaluated on a separate test dataset. To reduce variability from random seed training, results are averaged over 5 seeds. The neural networks are built using JAX, leveraging

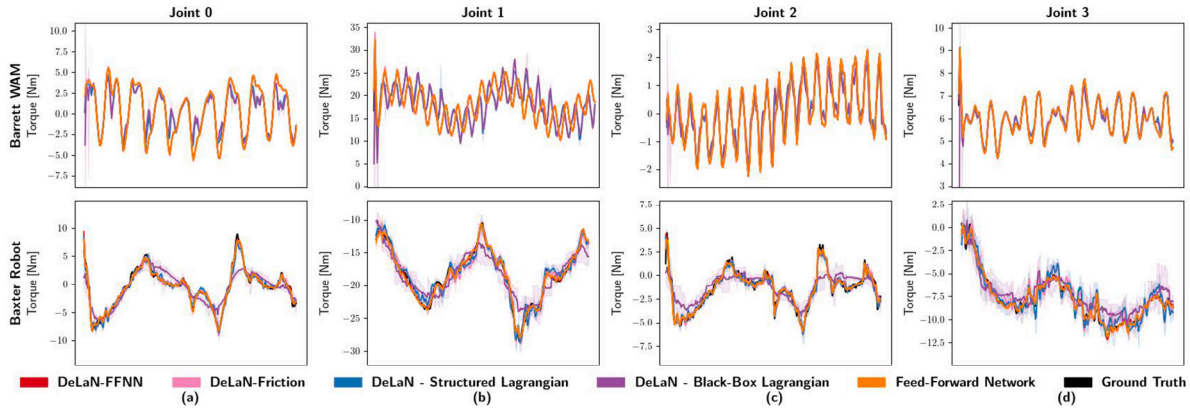

Fig. 7. (a–d) show the inverse dynamics modeling performance for the first four joints of each system on the selected sub-trajectories, with the learned inverse model averaged over 5 seeds.

Table 1
The hyperparameters used for the dynamics models vary across the different dynamical systems.

	Baxter Robot	Barrett WAM	3-Dof manipulator
Total Dof	7	7	3
Activation	ReLu	Tanh	Tanh
Network Dimension	[2×128]	[4×300]	[2×256]

its automatic differentiation for computing partial derivatives in the dynamics model.

For all systems, the ADAM optimizer is used with a batch size of 512, a learning rate of 10^{-4}, and a weight decay of 10^{-5}. Hyperparameters for each system's dynamics models are detailed in Table 1. Here, in the Network Dimension, for example, $[2 \times 128]$ denotes two hidden layers, with 128 neurons in each layer.

4.2. Non-rigid body dynamics modeling

In non-rigid body dynamics modeling experiments, dynamics models are applied to the Baxter Robot and Barrett WAM to assess the effectiveness of PINNs when rigid-body priors are inapplicable. The goal is to evaluate whether DeLaN-FFNN maintains interpretability while offering better modeling performance and generalization compared to other baselines.

4.2.1. Dataset construction

The datasets for the Baxter Robot and Barrett WAM are sourced from publicly available real system datasets. Detailed descriptions of these datasets are provided below.

Barrett WAM Dataset. This dataset is specifically designed for tackling an inverse dynamics problem associated with a 7-dof Barrett WAM manipulator. The dataset contains 12,000 training samples of seven degrees of freedom q, $\dot{q}$, $\ddot{q}$, and corresponding τ, as well as 3000 test samples [42]. The dataset can be accessed online at thislink.

Baxter Robot Dataset. This dataset contains the q, $\dot{q}$, and corresponding τ of a 7-dof Baxter Robot compliant manipulator. However, this dataset does not provide acceleration data $\ddot{q}$. In real-world systems, $\ddot{q}$ is usually unobserved and estimated via finite differences, which can amplify high-frequency noise [20,43]. To address this, low-pass filters are applied for more accurate estimates. Following [20,43], we compute accelerations using finite differences combined with low-pass filtering. Based on the different trajectories executed by the end effector, this dataset is categorized. The training set includes random trajectories, helical trajectories, and spiral trajectories, while the test set comprises a square trajectory, a helical trajectory, and circular trajectories on the XY and XZ plane. For more data details, please refer to Table 2 and the paper [5]. The dataset can be accessed online at thislink.

Table 2
Classification of trajectories in Baxter Robot training and testing data, and the number of data samples for each type.

	Trajectory	Number of samples
Training trajectories	Random trajectories	5661
	Spiral trajectories	28 395
	Helical trajectories	62 220
Test trajectories	Circular trajectory XY	5651
	Circular trajectory XZ	5659
	Square trajectory	5663
	Helical trajectory	5661

4.2.2. Prediction performance evaluation

The Barrett WAM dataset, along with the helical trajectories from the Baxter Robot training and test sets, is used to assess the accuracy of non-rigid dynamics modeling for each model. For the Barrett WAM, there are 12,000 training samples and 3000 test samples, while for the Baxter Robot, the training set consists of 62,220 samples, and the test set includes 5661 samples. The modeling results for the Barrett WAM and Baxter Robot, including detailed results for each of the 7-dof and the overall modeling accuracy, are presented in Table 3.

For joint modeling accuracy (Table 3), FF-NN and DeLaN-FFNN achieve the lowest MSEs in both the Barrett WAM and Baxter Robot. In the Barrett WAM, FF-NN, DeLaN-Friction, and DeLaN-FFNN outperform structured/black-box DeLaN, especially in the first four joints, which are key to its dynamics as they primarily drive the system's dynamic behavior. For the coefficient of determination (R^2), DeLaN-FFNN, DeLaN-Friction, and FF-NN perform similarly with values near 1, surpassing structured/black-box DeLaN.

In the Baxter Robot, DeLaN-FFNN and FF-NN also show high R^2, while DeLaN-Friction performs better than structured/black-box DeLaN but remains lower. The system's non-rigid dynamics, with its more pronounced elasticity, make friction alone insufficient. To illustrate the modeling effectiveness, visualizations of the first four joints, which dominate the dynamics, are provided. In the test set, the first 200 data samples from the Barrett WAM and 500 from the Baxter Robot are selected as sub-trajectories for clearer visualization, with the results displayed in Fig. 7.

The above results indicate that DeLaNs, which incorporate rigid body dynamics priors, are less effective for non-rigid systems, while DeLaN-Friction improves accuracy by compensating for torque output. However, it still underperforms compared to black-box models,

Table 3
The MSE of each joint in the Barrett WAM and Baxter Robot as well as the corresponding confidence interval averaged over 5 seeds. The coefficient of determination R^2 is calculated on the entire set for each system.

		Inverse model (MSE)							
Barrett WAM # Test samples = 3000		Joint 0	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	R^2
DeLaN	Structured Lagrangian	1.2e+0 ± 2.7e−2	1.6e+1 ± 1.5e+0	1.3e−1 ± 4.0e−2	2.4e−2 ± 2.2e−3	2.1e−3 ± 2.2e−4	6.7e−4 ± 3.4e−4	2.8e−4 ± 6.4e−4	0.943 ± 5.09e−3
DeLaN	Black-Box Lagrangian	1.8e+0 ± 8.2e−2	1.7e+1 ± 7.7e−1	1.4e−1 ± 2.2e−2	1.0e−1 ± 5.3e−2	2.6e−3 ± 3.3e−4	5.4e−4 ± 2.1e−4	7.7e−5 ± 1.9e−5	0.939 ± 2.83e−3
FF-NN	Feed-forward Network	**5.3e−3 ± 3.2e−3**	9.7e−3 ± 6.3e−3	**3.6e−4 ± 7.5e−4**	**2.9e−4 ± 7.9e−5**	3.0e−5 ± 1.4e−5	**2.5e−5 ± 6.6e−6**	**1.2e−6 ± 1.5e−5**	**0.999 ± 2.15e−5**
DeLaN-Friction	Introducing Friction	1.3e−2 ± 1.1e−3	7.6e−2 ± 9.8e−3	2.8e−3 ± 4.4e−4	1.1e−3 ± 3.7e−4	1.6e−4 ± 6.7e−5	1.0e−4 ± 3.1e−5	4.1e−5 ± 2.9e−5	**0.999 ± 3.69e−5**
DeLaN-FFNN	An Augmented DeLaN	6.0e−3 ± 2.8e−3	**8.3e−2 ± 4.9e−3**	4.9e−4 ± 3.1e−4	3.2e−4 ± 3.0e−4	**2.9e−5 ± 2.1e−5**	2.9e−5 ± 2.5e−5	1.6e−5 ± 1.5e−5	**0.999 ± 2.19e−5**
Baxter Robot # Test samples = 5661		Joint 0	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	R^2
DeLaN	Structured Lagrangian	4.5e−1 ± 3.6e−2	9.6e−1 ± 8.8e−2	2.0e−1 ± 3.8e−2	1.0e+0 ± 3.4e−1	1.6e−2 ± 1.4e−3	2.1e−2 ± 3.4e−3	1.2e−2 ± 4.0e−3	0.990 ± 1.65e−3
DeLaN	Black-Box Lagrangian	3.1e+0 ± 6.7e−1	3.2e+0 ± 2.0e+0	1.3e+0 ± 3.8e−1	2.0e+0 ± 1.5e+0	2.3e−2 ± 3.8e−3	2.6e−2 ± 4.0e−3	1.4e−3 ± 5.7e−4	0.964 ± 1.35e−2
FF-NN	Feed-forward Network	**1.9e−1 ± 2.3e−2**	2.5e−1 ± 4.5e−2	1.0e−1 ± 1.9e−2	**6.1e−2 ± 4.5e−3**	6.3e−3 ± 8.0e−4	8.8e−3 ± 1.3e−3	**4.3e−4 ± 2.1e−5**	**0.998 ± 3.12e−4**
DeLaN-Friction	Introducing Friction	4.2e−1 ± 5.9e−2	8.6e−1 ± 2.2e−1	1.8e−1 ± 2.9e−2	8.7e−1 ± 5.3e−1	1.6e−2 ± 1.8e−3	2.1e−2 ± 1.4e−3	1.2e−2 ± 1.6e−3	0.991 ± 1.95e−3
DeLaN-FFNN	An Augmented DeLaN	**1.9e−1 ± 1.2e−2**	**2.2e−1 ± 2.6e−2**	**9.8e−2 ± 7.7e−3**	6.9e−2 ± 9.8e−3	**6.1e−3 ± 5.3e−4**	**8.0e−3 ± 1.2e−3**	5.0e−4 ± 3.3e−5	**0.998 ± 1.59e−4**

Note: The bolded part refers to best results.

Table 4
The MSE of each joint in the Baxter Robot across three test trajectories, as well as the corresponding confidence interval averaged over 5 seeds. The coefficient of determination R^2 is calculated on the entire set for each system.

		Inverse Model (MSE)							
Circular Trajectory XY #Samples = 5651		Joint 0	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	R^2
DeLaN	Structured Lagrangian	4.4e−1 ± 9.7e−2	1.1e+0 ± 5.0e−1	2.9e−1 ± 4.9e−2	1.3e+0 ± 6.0e−1	1.7e−2 ± 7.8e−4	1.5e−2 ± 3.9e−3	8.4e−3 ± 2.5e−3	0.953 ± 2.37e−2
DeLaN	Black-Box Lagrangian	4.2e+0 ± 3.1e+0	2.0e+0 ± 2.3e+0	2.3e+0 ± 8.6e−1	1.9e+0 ± 1.5e+0	1.0e−2 ± 1.4e−3	1.6e−2 ± 4.6e−3	3.0e−3 ± 2.0e−3	0.986 ± 4.97e−3
FF-NN	Feed-forward Network	**2.0e−1 ± 2.9e−2**	3.9e−1 ± 1.6e−1	**1.4e−1 ± 5.7e−2**	**3.9e−2 ± 5.8e−3**	**5.2e−3 ± 1.1e−3**	5.3e−3 ± 3.8e−4	**6.0e−4 ± 1.2e−4**	0.996 ± 7.05e−4
DeLaN-Friction	Introducing Friction	3.6e−1 ± 2.9e−2	8.0e−1 ± 1.6e−1	2.0e−1 ± 1.5e−2	1.4e+0 ± 1.7e+0	1.6e−2 ± 7.2e−4	1.2e−2 ± 2.3e−3	8.3e−3 ± 6.1e−3	0.987 ± 7.52e−3
DeLaN-FFNN	An Augmented DeLaN	**2.0e−1 ± 3.9e−2**	**3.3e−1 ± 3.2e−2**	1.5e−1 ± 4.2e−2	4.0e−2 ± 8.9e−3	5.7e−3 ± 9.0e−4	**5.2e−3 ± 1.2e−3**	7.4e−4 ± 1.8e−4	**0.997 ± 4.19e−4**
Circular Trajectory XZ #Samples = 5659		Joint 0	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	R^2
DeLaN	Structured Lagrangian	3.7e−1 ± 3.9e−2	1.3e+0 ± 1.8e−1	2.1e−1 ± 7.0e−2	1.0e+0 ± 5.3e−1	**1.2e−2 ± 2.2e−3**	1.7e−2 ± 8.6e−3	1.4e−2 ± 5.8e−3	0.988 ± 2.38e−3
DeLaN	Black-Box Lagrangian	4.7e+0 ± 5.7e−1	3.8e+0 ± 6.4e−1	1.0e+0 ± 4.0e−1	1.6e+0 ± 7.8e−1	1.4e−2 ± 2.0e−3	1.1e−2 ± 2.6e−3	1.0e−2 ± 7.8e−3	0.955 ± 5.63e−3
FF-NN	Feed-forward Network	8.8e−1 ± 1.6e−1	2.8e−1 ± 6.9e−2	4.8e−1 ± 1.3e−1	**6.9e−2 ± 4.9e−2**	1.5e−2 ± 3.3e−3	9.0e−3 ± 2.9e−3	**2.1e−3 ± 5.5e−4**	0.993 ± 1.48e−3
DeLaN-Friction	Introducing Friction	**2.7e−1 ± 3.6e−2**	9.5e−1 ± 6.8e−1	**1.5e−1 ± 2.9e−2**	9.9e−1 ± 7.9e−1	1.3e−2 ± 2.7e−3	1.8e−2 ± 6.1e−3	1.2e−2 ± 4.4e−3	0.990 ± 6.00e−3
DeLaN-FFNN	An Augmented DeLaN	5.8e−1 ± 1.5e−1	**2.7e−1 ± 1.3e−1**	2.9e−1 ± 7.5e−2	7.6e−2 ± 3.5e−2	**1.2e−2 ± 7.3e−3**	**8.6e−3 ± 3.4e−3**	2.2e−3 ± 9.9e−4	**0.995 ± 4.68e−4**
Square Trajectory #Samples = 5663		Joint 0	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	R^2
DeLaN	Structured Lagrangian	7.2e−1 ± 1.3e−1	2.1e+0 ± 6.1e−1	**3.7e−1 ± 2.4e−1**	1.6e+0 ± 6.9e−1	5.5e−2 ± 2.4e−2	7.2e−2 ± 1.5e−2	3.6e−2 ± 1.4e−2	0.972 ± 7.38e−3
DeLaN	Black-Box Lagrangian	1.3e+1 ± 3.3e+0	4.3e+0 ± 3.7e+0	4.3e+0 ± 9.8e−1	2.0e+0 ± 1.6e+0	**2.3e−2 ± 6.1e−3**	7.7e−2 ± 1.1e−1	9.7e−3 ± 7.0e−3	0.864 ± 3.24e−2
FF-NN	Feed-forward Network	8.8e−1 ± 5.5e−1	1.2e+0 ± 4.9e−1	6.6e−1 ± 4.0e−1	**1.6e−1 ± 6.0e−2**	3.3e−2 ± 1.2e−2	**4.2e−2 ± 1.1e−2**	2.5e−3 ± 9.5e−4	0.983 ± 4.76e−3
DeLaN-Friction	Introducing Friction	**5.4e−1 ± 7.5e−2**	1.0e+0 ± 3.9e−1	3.8e−1 ± 1.2e−1	1.6e+0 ± 6.5e−1	4.5e−2 ± 2.6e−2	7.3e−2 ± 1.4e−2	2.2e−2 ± 4.0e−3	0.979 ± 4.93e−3
DeLaN-FFNN	An Augmented DeLaN	6.7e−1 ± 5.1e−1	**8.3e−1 ± 1.8e−1**	6.2e−1 ± 2.4e−1	1.7e−1 ± 8.7e−2	**2.3e−2 ± 5.0e−3**	4.7e−2 ± 1.7e−2	**2.2e−3 ± 1.5e−3**	**0.986 ± 3.77e−3**

Note: The bolded part refers to best results.

highlighting that adding friction alone is insufficient. DeLaN-FFNN, by learning non-linear dynamics, including non-rigid elements, maintains interpretability while achieving superior accuracy.

4.2.3. Generalization experiment

To assess the generalization capabilities of each dynamics model, all training trajectories of the Baxter Robot, including random, spiral, and helical trajectories, totaling 96,276 data samples, are used for model training. The models are then tested on circular trajectory XY, circular trajectory XZ, and square trajectory test sets, none of which appeared in the training data. A generalizable inverse dynamics model should accurately predict torque values for any trajectory, regardless of the training trajectory used [5].

The modeling results, including joint-specific and overall accuracy, are shown in Table 4. For the R^2 metric across the three test trajectories, the black-box DeLaN performs the worst with the lowest R^2 values. Although the structured DeLaN shows some improvement, it still underperforms compared to FF-NN, DeLaN-Friction, and DeLaN-FFNN. This indicates that DeLaNs have lower accuracy and generalization for compliant manipulators than black-box models. DeLaN-Friction also lags behind FF-NN, suggesting that friction alone is not enough to improve accuracy. DeLaN-FFNN, with R^2 scores near 1 across all trajectories, outperforms all other models, showing superior accuracy and generalization for compliant manipulators without rigid-body dynamics constraints.

Table 4 analyzes the modeling accuracy for each joint. For the circular XY trajectory test data, DeLaN-FFNN and FF-NN consistently show the highest accuracy, emphasizing their advantage of not being restricted by rigid-body dynamics priors. For the circular XZ and square trajectories, DeLaN-FFNN performs best for most joints, with a few exceptions in other models. This demonstrates DeLaN-FFNN's superior accuracy for compliant manipulators and better generalization for unknown trajectories. To illustrate this, Fig. 8 visualizes the first 500 data samples from the first four joints across the three test trajectories.

4.2.4. Summary and discussion

Synthesizing the experimental results, we conclude that PINNs constrained by rigid-body dynamics are insufficient for compliant manipulators, showing limited accuracy and poor generalization. Adding friction to DeLaN improves accuracy but still falls short of ideal inverse dynamics modeling and generalization, especially compared to black-box methods that are not constrained by any assumptions. In contrast, DeLaN-FFNN maintains physical interpretability while achieving superior modeling accuracy for the Barrett WAM and Baxter Robot, with better generalization for unknown trajectories. These findings show that DeLaN-FFNN overcomes the limitations of rigid-body dynamics, making it suitable for a wider range of physical systems.

Flexible multibody dynamics involves strong nonlinear coupling, especially under finite rotations [44]. This complexity intensifies in compliant mechanisms with large deformations or complex serial-parallel structures, where the main challenges lie in modeling coupled nonlinear behaviors, managing topological redundancy, and balancing accuracy with generality [45].

The Baxter Robot used in this study is a compliant manipulator equipped with SEAs and passive springs, offering limited passive compliance. However, it is not a flexure-based mechanism and exhibits only small joint-level deformations, forming a simple serial structure without the topological complexity of parallel or hybrid configurations [5]. Thus, experiments on this platform are insufficient to validate the effectiveness of DeLaN-FFNN for highly deformable compliant systems, which require further investigation.

DeLaN-FFNN introduces an uncertainty force term to extend the original DeLaN model beyond rigid-body assumptions, enabling it to

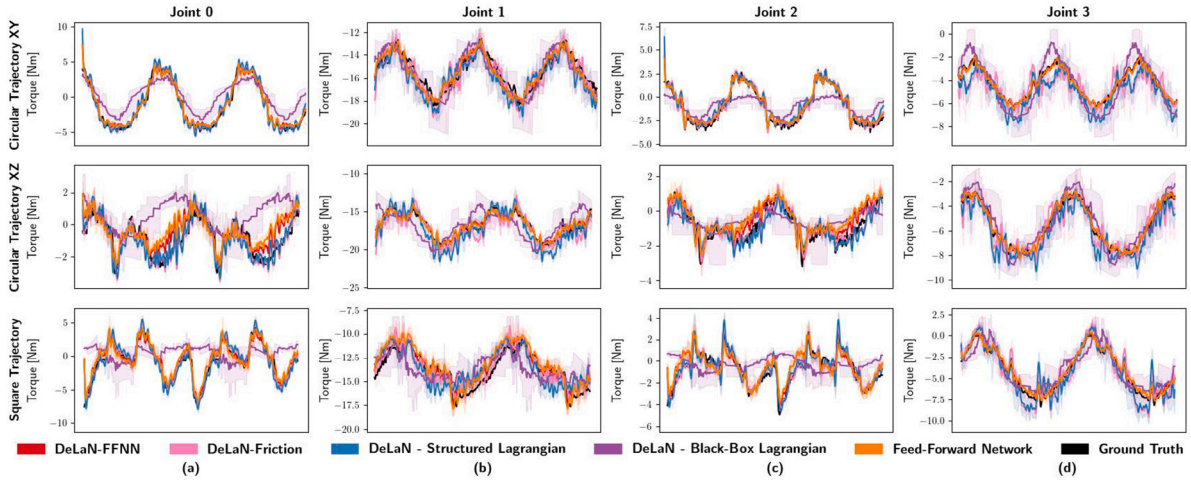

Fig. 8. (a–d) show the inverse dynamics modeling performance for the first four joints of each test trajectory (circular trajectory XY, circular trajectory XZ and square trajectory) on the selected sub-trajectories, with the learned inverse model averaged over 5 seeds.

capture certain compliant behaviors. However, since DeLaN was originally designed for rigid serial manipulators, the current framework remains limited in handling compliant mechanisms with complex serial-parallel topologies. Future work should focus on adapting its architecture to better model such systems.

4.3. Rigid body dynamics modeling

For the rigid body dynamics experiments, our goal is to assess whether the PINNs can learn and recover the underlying system dynamics with ideal observations.

4.3.1. Dataset construction

To ensure smooth velocity and direction changes without abrupt reactions in the robotic manipulator, we base trajectory planning on sinusoidal functions. This allows for a gradual transition from the initial angle (q_0) to the final angle (q_f), minimizing sudden accelerations or decelerations at the start and end. The trajectory for the ith joint is defined as follows:

$$q_t^i = q_0^i + \frac{q_f^i - q_0^i}{T}\left(t - \frac{T}{2\pi}\sin\left(\frac{2\pi}{T}t\right)\right) \tag{12}$$

In this setup, T is the total movement time, and t represents the current time. Within the robot's joint angle limits, 200 random sets of initial and final angles for three joints are generated. With an 8-second sampling period and a frequency of 50 Hz, 200 trajectories are collected, each containing 400 data points, resulting in 80,000 data sets for training. These include joint positions q, velocities $\dot{q}$, accelerations $\ddot{q}$, momenta p, their derivatives $\dot{p}$, and joint torques τ, uniformly covering the state domain. Inverse dynamics is calculated using the Newton-Euler method in Pybullet. Three additional trajectories are collected for testing, as shown in Fig. 9(a).

4.3.2. Prediction performance evaluation

The results of the rigid body inverse model experiments are summarized in Table 5 and shown in Fig. 10. All models effectively learned an inverse model that fits the test set. On average, PINNs achieved lower MSE than FF-NN. Structured Lagrangian/Hamiltonian methods showed better accuracy than their corresponding black-box models. DeLaN-FFNN and DeLaN-Friction performed similarly to structured DeLaN, outperforming other models. Notably, DeLaN-Friction and DeLaN-FFNN performed well even with data containing only conservative forces. This is because DeLaN-Friction trains friction coefficients as network weights, while DeLaN-FFNN reduces the output of the uncertainty force network when there are no uncertainties to learn.

4.3.3. Torque decomposition experiment

In the decomposition of torque into inertial, centrifugal, Coriolis, and gravitational forces, all models performed well. As seen in Table 5, PINNs consistently outperformed FF-NN in terms of nMSE across all torque components. For inertial and Coriolis torques, only structured

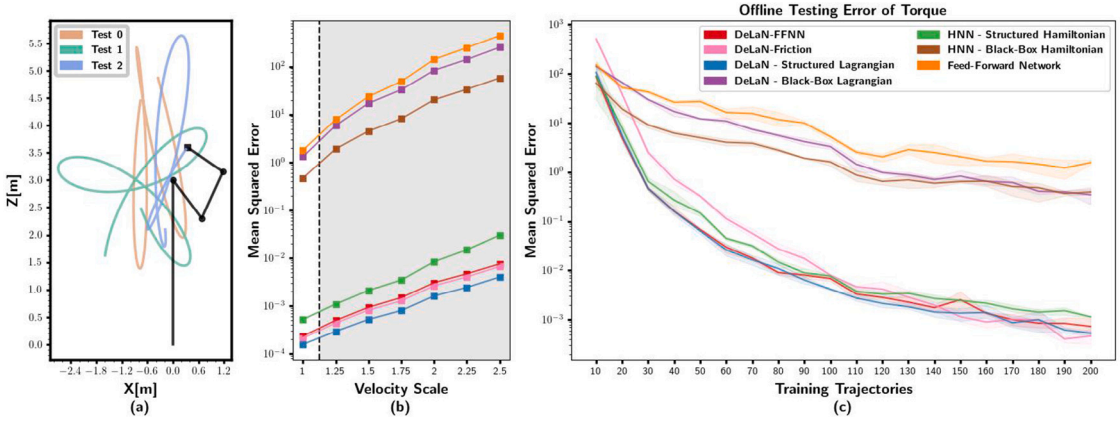

Fig. 9. (a) Depicts the three test trajectories of the simulated 3-dof robotic manipulator operating in the x–z plane. (b) Shows each dynamics model perform when applied to increased velocities, assessing their capability to extrapolate to new velocity conditions(the gray areas represent the test data where velocities are increased) and (c) presents the assessment of inverse dynamics modeling accuracy for each dynamics model as the number of training trajectories increases from 10 to 200, with each trajectory containing 400 data samples.

Table 5
The normalized mean squared error for dynamics modeling, as well as the corresponding confidence interval averaged over 5 seeds.

3-Dof Manipulator # Test Samples = 1200		Inverse Model (nMSE)			
		Torque τ	Inertial Torque τ_I	Coriolis Torque τ_c	Gravitational Torque τ_g
DeLaN	Structured Lagrangian	**4.35e−7 ± 1.44e−7**	**1.97e−6 ± 1.75e−6**	**1.04e−5 ± 2.89e−6**	**2.06e−7 ± 9.29e−8**
DeLaN	Black-Box Lagrangian	3.91e−3 ± 6.04e−4	2.55e−3 ± 1.83e−3	1.38e−1 ± 2.22e−2	**9.99e−6 ± 7.30e−6**
HNN	Structured Hamiltonian	**1.59e−6 ± 1.94e−7**	1.92e−4 ± 1.07e−4	3.58e−4 ± 9.91e−5	**6.34e−7 ± 2.28e−7**
HNN	Black-Box Hamiltonian	1.99e−3 ± 1.27e−3	6.80e−1 ± 2.20e−1	8.96e−1 ± 1.25e−1	1.01e−4 ± 3.01e−5
FF-NN	Feed-Forward Network	5.46e−3 ± 2.21e−3	1.30e−1 ± 3.15e−2	1.82e−1 ± 7.95e−2	1.75e−4 ± 1.25e−4
DeLaN-Friction	Introducing Friction	**9.42e−7 ± 1.29e−6**	**5.16e−6 ± 8.66e−6**	**2.40e−5 ± 2.88e−5**	**2.90e−7 ± 3.62e−7**
DeLaN-FFNN	An Augmented DeLaN	**7.42e−7 ± 3.63e−7**	**6.14e−6 ± 1.65e−6**	**2.10e−5 ± 1.14e−5**	**2.29e−7 ± 6.00e−8**

Note: The bolded part refers to better results.

DeLaN, DeLaN-Friction, and DeLaN-FFNN showed superior results because they learned the true and reasonable mass matrix of the system during training. In modeling gravitational force, all models except FF-NN and black-box HNN demonstrated higher accuracy. Structured Lagrangian/Hamiltonian methods were notably more effective than their black-box counterparts, especially for inertial and Coriolis torques.

4.3.4. Velocity extrapolation and data efficiency experiments

In addition to inverse dynamics modeling and force decomposition, we evaluate the models' ability to extrapolate to new velocity conditions (Fig. 9(b)) and their performance as training data volume increased (Fig. 9(c)). Initially trained at a velocity scale of 1×, the models were tested at higher velocities. At 2.5×, FF-NN and black-box DeLaN/HNN were no longer viable, while structured DeLaN/HNN, DeLaN-Friction, and DeLaN-FFNN still maintained accuracy, even outperforming the black-box models at 1×. Structured DeLaN had the best velocity extrapolation, as its input domain includes only q, making it unaffected by velocity changes. Models like FF-NN, black-box DeLaN, and HNNs, which depend on $(q, \dot{q}, \ddot{q})$, $(q, \dot{q})$, or (q, p), were more affected. DeLaN-FFNN and DeLaN-Friction showed similar extrapolation to structured DeLaN. Fig. 9(c) highlights structured DeLaN/HNN, DeLaN-Friction, and DeLaN-FFNN achieving lower test MSEs than FF-NN and black-box DeLaN/HNN, with the gap widening as the training set grows, demonstrating their better data efficiency.

4.3.5. Summary

In summary, the rigid body dynamics experiments demonstrate that PINNs effectively capture the underlying structure of the dynamical system. In inverse dynamics modeling, torque decomposition, velocity extrapolation, and data efficiency, structured DeLaN/HNN outperform their black-box counterparts and FF-NN. Across all rigid body dynamics modeling tests, structured DeLaN achieves the best results, followed closely by DeLaN-FFNN and DeLaN-Friction. DeLaN-FFNN's results are very similar to those of structured DeLaN, partly due to the optimization objective in Eq. (10), which emphasizes the role of structured DeLaN when the dynamics being learned consist solely of rigid-body dynamics.

5. Conclusion

This paper introduces PINNs like DeLaNs and HNNs, which incorporate rigid-body dynamics priors to effectively learn the structure of simulated rigid-body systems. However, they struggle with non-rigid systems like compliant manipulators. To address this, an augmented deep Lagrangian network (DeLaN-FFNN) is proposed, combining physics-informed and standard deep networks to model both rigid and non-rigid dynamics. Experiments with compliant manipulators, including the Baxter Robot, Barrett WAM, and a rigid-body simulated robot, show that DeLaN-FFNN matches the performance of PINNs for rigid systems while excelling in compliant systems. It provides precise modeling, improved generalization, and retained interpretability across a range of manipulators. However, its effectiveness in highly deformable compliant systems still requires further validation, and it remains limited in handling compliant mechanisms with complex serial-parallel structures. Future work should focus on adapting and extending the network architecture to make it more suitable for modeling such systems.

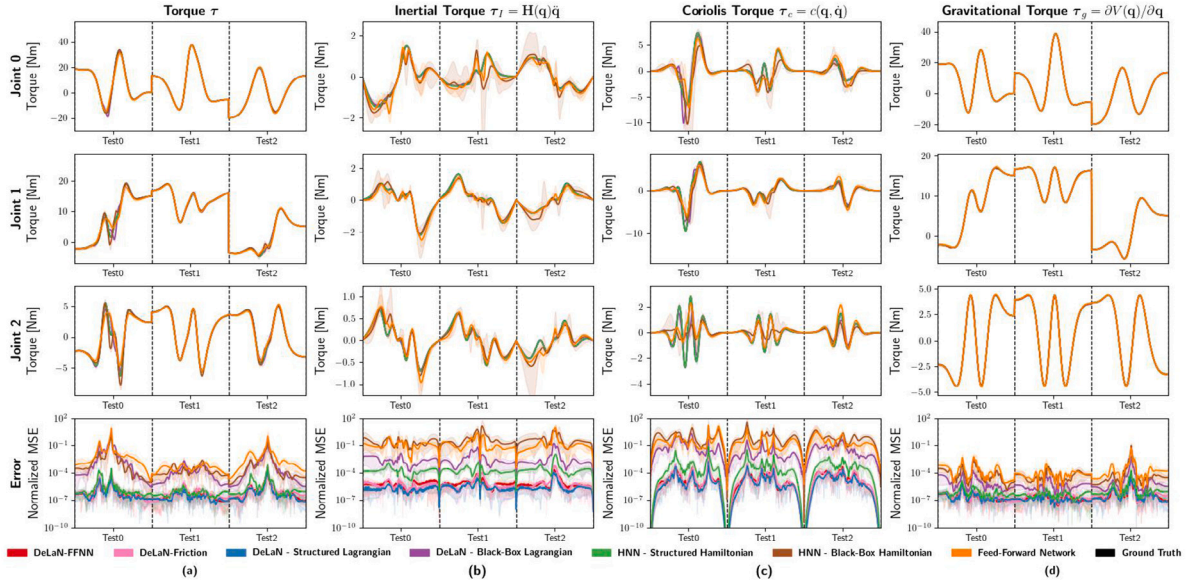

Fig. 10. (a) Displays the learned inverse model using the training dataset, with three test trajectories not included in the training set and results averaged over 5 seeds. The subsequent columns provide the predicted force decomposition. (b) Illustrates the inertial force $H(q)\ddot{q}$, (c) shows the Coriolis and Centrifugal forces $c(q, \dot{q})$, and (d) depicts the gravitational force $g(q)$.

CRediT authorship contribution statement

Zhiming Li: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Shuangshuang Wu:** Writing – review & editing, Validation, Supervision, Software, Methodology, Conceptualization. **Wenbai Chen:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition. **Fuchun Sun:** Writing – review & editing, Supervision, Project administration, Investigation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This research was funded by the National Key Research and Development Program of China (No. 2021ZD0114505), the National Natural Science Foundation of China (62276028), the Major Research Plan of the National Natural Science Foundation of China (92267110), the Beijing Municipal Natural Science Foundation–Xiaomi Innovation Joint Fund (L233006), the Qin Xin Talents Cultivation Program at Beijing Information Science & Technology University (QXTCP A202102), and the Beijing Information Science and Technology University School Research Fund (No. 2023XJJ12).

Data availability

Data will be made available on request.

References

[1] A. Dalla Libera, R. Carli, A data-efficient geometrically inspired polynomial kernel for robot inverse dynamic, IEEE Robot. Autom. Lett. 5 (1) (2019) 24–31.
[2] S. Wu, Z. Li, W. Chen, F. Sun, Dynamic modeling of robotic manipulator via an augmented deep Lagrangian network, Tsinghua Sci. Technol. 29 (5) (2024) 1604–1614.
[3] M. Giuliani, C. Lenz, T. Müller, M. Rickert, A. Knoll, Design principles for safety in human-robot interaction, Int. J. Soc. Robot. 2 (2010) 253–274.
[4] C. Lee, S. Kwak, J. Kwak, S. Oh, Generalization of series elastic actuator configurations and dynamic behavior comparison, in: Actuators, vol. 6, (3) MDPI, 2017, p. 26.
[5] B. Valencia-Vidal, E. Ros, I. Abadía, N.R. Luque, Bidirectional recurrent learning of inverse dynamic models for robots with elastic joints: a real-time real-world implementation, Front. Neurorobotics 17 (2023) 1166911.
[6] W. Deng, F. Ardiani, K.T. Nguyen, M. Benoussaad, K. Medjaher, Physics informed machine learning model for inverse dynamics in robotic manipulators, Appl. Soft Comput. 163 (2024) 111877.
[7] B. Armstrong, O. Khatib, J. Burdick, The explicit dynamic model and inertial parameters of the PUMA 560 arm, in: Proceedings. 1986 IEEE International Conference on Robotics and Automation, vol. 3, IEEE, 1986, pp. 510–518.
[8] J. Zhang, F. Zhang, M. Cheng, R. Ding, B. Xu, H. Zong, Parameter identification of hydraulic manipulators considering physical feasibility and control stability, IEEE Trans. Ind. Electron. 71 (1) (2023) 718–728.
[9] A.K. Tangirala, Principles of System Identification: Theory and Practice, CRC Press, 2018.
[10] J.-A. Ting, M.N. Mistry, J. Peters, S. Schaal, J. Nakanishi, A Bayesian approach to nonlinear parameter identification for rigid body dynamics, in: Robotics: Science and Systems, Citeseer, 2006, pp. 32–39.
[11] M. Gautier, G. Venture, Identification of standard dynamic parameters of robots with positive definite inertia matrix, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 5815–5820.
[12] J. Swevers, W. Verdonck, J. De Schutter, Dynamic model identification for industrial robots, IEEE Control Syst. Mag. 27 (5) (2007) 58–71.
[13] G.A. Pratt, M.M. Williamson, Series elastic actuators, in: Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, vol. 1, IEEE, 1995, pp. 399–406.
[14] C. Lee, S. Kwak, J. Kwak, S. Oh, Generalization of series elastic actuator configurations and dynamic behavior comparison, in: Actuators, vol. 6, (3) MDPI, 2017, p. 26.
[15] E. Madsen, O.S. Rosenlund, D. Brandt, X. Zhang, Comprehensive modeling and identification of nonlinear joint dynamics for collaborative industrial robot manipulators, Control Eng. Pract. 101 (2020) 104462.
[16] A. Calanca, L.M. Capisani, A. Ferrara, L. Magnani, MIMO closed loop identification of an industrial robot, IEEE Trans. Control Syst. Technol. 19 (5) (2010) 1214–1224.
[17] A.S. Polydoros, L. Nalpantidis, V. Krüger, Real-time deep learning of robotic manipulator inverse dynamics, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2015, pp. 3442–3448.
[18] M. Ruderman, M. Iwasaki, Sensorless torsion control of elastic-joint robots with hysteresis and friction, IEEE Trans. Ind. Electron. 63 (3) (2015) 1889–1899.
[19] K. Laddach, R. Łangowski, T.A. Rutkowski, B. Puchalski, An automatic selection of optimal recurrent neural network architecture for processes dynamics modelling purposes, Appl. Soft Comput. 116 (2022) 108375.
[20] M. Lutter, J. Peters, Combining physics and deep learning to learn continuous-time dynamics models, Int. J. Robot. Res. 42 (3) (2023) 83–107.
[21] A.S. Polydoros, E. Boukas, L. Nalpantidis, Online multi-target learning of inverse dynamics models for computed-torque control of compliant manipulators, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2017, pp. 4716–4722.
[22] A. Carron, E. Arcari, M. Wermelinger, L. Hewing, M. Hutter, M.N. Zeilinger, Data-driven model predictive control for trajectory tracking with a robotic arm, IEEE Robot. Autom. Lett. 4 (4) (2019) 3758–3765.
[23] Z. Li, S. Wu, W. Chen, F. Sun, Extrapolation of physics-inspired deep networks in learning robot inverse dynamics, Mathematics 12 (16) (2024) 2527.
[24] B.Z. Cunha, C. Droz, A.-M. Zine, S. Foulard, M. Ichchou, A review of machine learning methods applied to structural dynamics and vibroacoustic, Mech. Syst. Signal Process. 200 (2023) 110535.
[25] S.J. Malham, An Introduction to Lagrangian and Hamiltonian Mechanics, Maxwell Institute for Mathematical Sciences & School of Mathematical and Computer Sciences Heriot-Watt University, Edinburgh EH14 4AS, UK, 2014.
[26] M. Lutter, C. Ritter, J. Peters, Deep lagrangian networks: Using physics as model prior for deep learning, in: International Conference on Learning Representations, 2019.
[27] S. Greydanus, M. Dzamba, J. Yosinski, Hamiltonian neural networks, Adv. Neural Inf. Process. Syst. 32 (2019).
[28] W. Zhai, D. Tao, Y. Bao, Parameter estimation and modeling of nonlinear dynamical systems based on Runge–Kutta physics-informed neural network, Nonlinear Dynam. 111 (22) (2023) 21117–21130.
[29] M. Lutter, K. Listmann, J. Peters, Deep lagrangian networks for end-to-end learning of energy-based control for under-actuated systems, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2019, pp. 7718–7725.
[30] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, S. Ho, Lagrangian neural networks, 2020, arXiv preprint arXiv:2003.04630.
[31] T. Duong, A. Altawaitan, J. Stanley, N. Atanasov, Port-Hamiltonian neural ODE networks on Lie groups for robot dynamics learning and control, IEEE Trans. Robot. (2024).
[32] H. Hu, Z. Shen, C. Zhuang, A PINN-based friction-inclusive dynamics modeling method for industrial robots, IEEE Trans. Ind. Electron. (2024).
[33] J. Liu, P. Borja, C. Della Santina, Physics-informed neural networks to model and control robots: A theoretical and experimental investigation, Adv. Intell. Syst. 6 (5) (2024) 2300385.
[34] M. Lahoud, G. Marchello, M. D'Imperio, A. Müller, F. Cannella, A deep learning framework for non-symmetrical Coulomb friction identification of robotic manipulators, in: 2024 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2024, pp. 10510–10516.
[35] C. Feng, J. Wang, Y. Shen, Q. Wang, Y. Xiong, X. Zhang, J. Fan, Physics-informed neutral network with physically consistent and residual learning for excavator precision operation control, Appl. Soft Comput. 167 (2024) 112402.
[36] X. Li, W. Shang, S. Cong, Offline reinforcement learning of robotic control using deep kinematics and dynamics, IEEE/ASME Trans. Mechatronics (2023).
[37] D. Nguyen-Tuong, J. Peters, Using model knowledge for learning inverse dynamics, in: 2010 IEEE International Conference on Robotics and Automation, IEEE, 2010, pp. 2677–2682.
[38] M. Lutter, J. Silberbauer, J. Watson, J. Peters, A differentiable newton euler algorithm for multi-body model learning, 2020, arXiv preprint arXiv:2010.09802.
[39] K. Hitzler, F. Meier, S. Schaal, T. Asfour, Learning and adaptation of inverse dynamics models: A comparison, in: 2019 IEEE-RAS 19th International Conference on Humanoid Robots, Humanoids, IEEE, 2019, pp. 491–498.
[40] E. Madsen, O.S. Rosenlund, D. Brandt, X. Zhang, Comprehensive modeling and identification of nonlinear joint dynamics for collaborative industrial robot manipulators, Control Eng. Pract. 101 (2020) 104462.
[41] V. Shaj, P. Becker, D. Büchler, H. Pandya, N. van Duijkeren, C.J. Taylor, M. Hanheide, G. Neumann, Action-conditional recurrent kalman networks for forward and inverse dynamics learning, in: Conference on Robot Learning, PMLR, 2021, pp. 765–781.
[42] D. Nguyen-Tuong, M. Seeger, J. Peters, Model learning with local gaussian process regression, Adv. Robot. 23 (15) (2009) 2015–2034.
[43] M. Reuss, N. van Duijkeren, R. Krug, P. Becker, V. Shaj, G. Neumann, End-to-end learning of hybrid inverse dynamics models for precise and compliant impedance control, 2022, arXiv preprint arXiv:2205.13804.

[44] A.A. Shabana, Flexible multibody dynamics: review of past and recent developments, Multibody Syst. Dyn. 1 (1997) 189–222.
[45] M. Ling, L.L. Howell, J. Cao, G. Chen, Kinetostatic and dynamic modeling of flexure-based compliant mechanisms: a survey, Appl. Mech. Rev. 72 (3) (2020) 030802.

Zhiming Li received the B.Eng. degree in Automation from Lanzhou Jiaotong University, China, in 2022. He is currently pursuing a Master's degree in Control Science and Engineering at Beijing Information Science and Technology University, China. His main research interests include physics-informed neural networks, operator learning, AI for Science, and robotic modeling and control.

Shuangshuang Wu received the BEng degree in Automation and Ph.D. degree in control science and from Yanshan University, China, in 2014 and 2020, respectively. She was a Post-Doctoral Researcher with the Department of Computer Science and Technology in Tsinghua University, from 2020 to 2022. She is currently a lecturer with school of Automation, Beijing Information Science and Technology University, China. She current research interests include time-delay systems, and intelligent Modeling and Control.

Wenbai Chen received the B.S. degree from Northeastern University at Qinhuangdao in 2007, the M.S. degree from Yanshan University in 2004, and the Ph.D. degree from the Beijing University of Posts and Telecommunications in 2011. He is currently a Professor with the school of automation, Beijing Information Science and Technology University. His current research interests include intelligent robot, artificial intelligence, sensor fusion, machine learning and wireless sensor network. He is the director of Chinese Association for Artificial Intelligence.

Fuchun Sun received the Ph.D. degree from Tsinghua University, Beijing, China, in 1998. From 1998 to 2000, he was a Post-Doctoral Fellow with the Department of Automation, Tsinghua University, where he is currently a Professor with the Department of Computer Science and Technology. His current research interest includes robotic perception and cognition. Mr. Sun was a recipient of the National Science Fund for Distinguished Young Scholars. He serves as the Editor-in-Chief for Cognitive Computation and Systems and an Associate Editor for a series of international journals, including the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, the IEEE TRANSACTIONS ON FUZZY SYSTEMS, Mechatronics, and Robotics and Autonomous Systems.